Rario

Smart Contract Security Audit

Prepared by: **Halborn**

Date of Engagement: **October 11th, 2021 - November 1st, 2021**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	10/11/2021	Roberto Reigada
0.2	Document Edits	11/01/2021	Roberto Reigada
0.3	Document Review	11/02/2021	Gabi Urrutia
1.0	Remediation Plan	11/08/2021	Roberto Reigada
1.1	Remediation Plan Review	11/09/2021	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Roberto Reigada	Halborn	Roberto.Reigada@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Rario engaged Halborn to conduct a security audit on their smart contracts beginning on October 11th, 2021 and ending on November 1st, 2021. The security assessment was scoped to the smart contracts provided in the Github repository rariohq/mjolnir - v1 branch

1.2 AUDIT SUMMARY

The team at Halborn was provided two weeks for the engagement and assigned a full time security engineer to audit the security of the smart contract. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that smart contract functions operate as intended
- Identify potential security issues with the smart contracts

In summary, Halborn identified some security risks that were addressed by the Rario team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of this audit. While manual testing is recommended to uncover flaws in logic, process,and implementation; automated testing techniques help enhance coverage of the bridge code and can quickly identify items that do not follow security best practices. The following phases and associated tools were used throughout the term of the audit:

- Research into architecture and purpose
- Smart contract manual code review and walkthrough
- Graphing out functionality and contract logic/connectivity/functions (solgraph)
- Manual assessment of use and safety for the critical Solidity variables and functions in scope to identify any arithmetic related vulnerability classes
- Manual testing by custom scripts
- Static Analysis of security for scoped contract, and imported functions. (Slither)
- Testnet deployment (Brownie, Remix IDE)

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident, and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. It's quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that was used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.

1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

IN-SCOPE:
The security assessment was scoped to the following smart contracts:

- RarioCurrencyManager.sol
- RarioData.sol
- RarioEthereumToken.sol
- RarioMarketplace.sol
- RarioMarketplaceConfig.sol
- RarioPolygonToken.sol
- RarioUsers.sol
- traits/AccessControl.sol
- traits/Context.sol
- traits/CurrencyManager.sol
- traits/ERC165Interface.sol
- traits/ERC1822UUPS.sol
- traits/ERC1967Upgrade.sol
- traits/ERC721NFT.sol
- traits/Initializable.sol
- traits/Marketplace.sol
- traits/MarketplaceConfig.sol
- traits/MetadataDB.sol
- traits/Pausable.sol
- traits/RarioEthereumNFT.sol
- traits/RarioNFT.sol
- traits/RarioPolygonNFT.sol
- traits/ReentrancyGuard.sol
- traits/Upgradeable.sol
- traits/UserDB.sol
- All contracts inherited by these contracts

Commit ID: 2a16dcf44b02e3ecf90dcc6b05c8dffd17e4f3aa
Fixed Commit ID: 01f0ba72f9067f6737a1d677e587959223f557e5

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	2	1	2	4

LIKELIHOOD

IMPACT

		(HAL-01) (HAL-02)		
	(HAL-03)			
(HAL-05)				
	(HAL-04)			
(HAL-06) (HAL-07) (HAL-08) (HAL-09)				

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - INCORRECT LOGIC LEADS TO DOS IN AUCTION SALES	High	SOLVED - 11/08/2021
HAL02 - OUTBID USER DOES NOT RECEIVE BACK HIS FUNDS	High	SOLVED - 11/08/2021
HAL03 - UNCHECKED TRANSFER	Medium	SOLVED - 11/08/2021
HAL04 - OF USE TRANSFER WHEN TRANSFERRING NFTS	Low	SOLVED - 11/08/2021
HAL05 - UPDATEUSER FUNCTION MISSING REQUIRE STATEMENT	Low	SOLVED - 11/08/2021
HAL06 - MINT FUNCTION IN RARIOPOLYGONNFT DOES NOT VERIFY IF TOKEN WAS WITHDRAWN	Informational	SOLVED - 11/08/2021
HAL07 - POSSIBLE MISUSE OF PUBLIC FUNCTIONS	Informational	SOLVED - 11/08/2021
HAL08 - UNUSED CONSTANTS	Informational	SOLVED - 11/08/2021
HAL09 - UNUSED EVENTS	Informational	SOLVED - 11/08/2021

FINDINGS & TECH DETAILS

3.1 (HAL-01) INCORRECT LOGIC LEADS TO DOS IN AUCTION SALES - HIGH

Description:

In the contract Marketplace.sol, the function placeBid is used to buy an NFT token listing. If the listingType is a RESERVE_PRICE_AUCTION, the function follows this code:

Listing 1: Marketplace.sol - placeBid (Lines 331,332,370,374)

```
330 else if (listing.listingType == ListingType.RESERVE_PRICE_AUCTION)
        {
331     if (listing.endTime == 0) {
332         // If this is the first bid, ensure it's >= the reserve
                price
333         require(
334             listing.price <= bidValueInListingCurrency,
335             "NFTMarketReserveAuction: Bid must be at least the
                    reserve price"
336         );
337         listing.price = bidValueInListingCurrency;
338         listing.buyerPaymentWallet = payable(msgSender());
339         listing.buyerTokenWallet = payable(buyerTokenWallet);
340         listing.bidCurrency = currency;
341         listing.bidAmount = bidValue;
342         // On the first bid, the endTime is now + duration
343         listing.endTime = block.timestamp + listing.duration;
344         emit AuctionBidPlaced(listing.listingId, msgSender(),
                bidValue, listing.endTime);
345     } else {
346         // If this bid outbids another, confirm that the bid is at
                least x% greater than the last
347         require(listing.endTime >= block.timestamp, "
                NFTMarketReserveAuction: Auction is over");
348         require(
349             listing.buyerPaymentWallet != msgSender(),
350             "NFTMarketReserveAuction: You already have an
                    outstanding bid"
351         );
352         uint256 minAmount = (listing.price * (BASIS_POINTS +
                _auctionPercentIncrementInBasisPoints)) /
```

```
353             BASIS_POINTS;
354         require(bidValueInListingCurrency >= minAmount, "
                NFTMarketReserveAuction: Bid amount too low");
355         // Cache and update bidder state before a possible
                reentrancy (via the value transfer)
356         uint256 originalAmount = listing.bidAmount;
357         address payable originalBidder = listing.
                buyerPaymentWallet;
358         listing.price = bidValueInListingCurrency;
359         listing.bidCurrency = currency;
360         listing.bidAmount = bidValue;
361         listing.buyerPaymentWallet = payable(msgSender());
362         listing.buyerTokenWallet = payable(buyerTokenWallet);
363
364         // When a bid outbids another, check to see if a time
                extension should apply.
365         if (listing.endTime - block.timestamp < listing.
                extensionDuration) {
366             listing.endTime = block.timestamp + listing.
                    extensionDuration;
367         }
368
369         if (currencyType == 2) {
370             IERC20(currencyContractAddress).transferFrom(msgSender
                    (), address(this), bidValue);
371         }
372
373         emit AuctionBidPlaced(listing.listingId, msgSender(),
                bidValue, listing.endTime);
374         _sendValue(originalBidder, originalAmount, _gasLimitMedium
                );
375     }
376 }
```

For the first bid, `if (listing.endTime == 0)` code block, the contract does not call `sendValue` nor `transferFrom(msgSender(), address(this), bidValue );`. This means that a user can place a bid as high as he wants and the contract would not check if the user actually owns those tokens.

If the amount bid is high enough, none else would be able to place a higher bid. This happens because after the first bid, the contract does require the user to send the tokens to the contract by calling `sendValue`

and `transferFrom` function as seen above.

Denial of Service example

First bid vs. Second bid

Risk Level:

Likelihood - 3

Impact - 5

Recommendation:

It is recommended to force the bidder of the first bid to transfer the amount into the contract following the same logic that the smart contract currently has now for all the bids that are not the first one. This can be achieved by simply adding these lines into the `if (listing.endTime == 0)`

code block:

Listing 2: Marketplace.sol - placeBid

```
if (currencyType == 2) {
    IERC20(currencyContractAddress).transferFrom(msgSender(),
        address(this), bidValue);
}

_sendValue(originalBidder, originalAmount, _gasLimitMedium);
```

Remediation Plan:

SOLVED: Rario team successfully fixed the placeBid function logic, correcting this vulnerability.

3.2 (HAL-02) OUTBID USER DOES NOT RECEIVE BACK HIS FUNDS - HIGH

Description:

In the contract Marketplace.sol, the function placeBid is used to buy an NFT token listing. In case that the listingType is a RESERVE_PRICE_AUCTION , when a user places a bid, the amount of tokens bid are transferred from his account to the contract address. Then, when a user outbids the previous user, the new user token amount bid is also transferred to the contract but the previous outbid user is not sent his bid tokens back:

```
USDC_AMOUNT2 = rarioCurrencyManager.convert("USD", "USDC", 6000) -> 59988002
Transaction sent: 0xfe70f1a46d2ed1acfb052f5e0699fa49961e0703489ec24bdcb56c09d61910912
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 32
  ERC20Token.transfer confirmed   Block: 13499849   Gas used: 51061 (0.76%)

Transaction sent: 0xff3180a760aec5de5adf0d635b5b777e87a71d00e4f5ec6b3740c0151a16201d
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  ERC20Token.approve confirmed   Block: 13499850   Gas used: 44127 (0.66%)

usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user3.address) -> 59988003
usdc.balanceOf(rarioMarketplace.address) -> 0
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user3.address, "USDC", USDC_AMOUNT2 + 1, {'from': user3})
Transaction sent: 0x0480921d845af703b2821ccc64b110c293c82df1963429785de88ba3dd290b2
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed   Block: 13499851   Gas used: 175308 (2.61%)

rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x54aF1a5AE93743D421b79368936c008919c435
47', '0x54aF1a5AE93743D421b79368936c008919c43547', '0xb91170570dE17326ff66a385773A4eD423d6a479', '0xb91170570dE17326ff66a385773A4eD423d6a479', 6000, 2592000, 900, 1637935150)
usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user3.address) -> 0
usdc.balanceOf(rarioMarketplace.address) -> 59988003
token.ownerOf(1) -> 0xb897bFDeC756b5AEA01f42B1Ad691Bb962383f8
user3.address -> 0xb91170570dE17326ff66a385773A4eD423d6a479
USDC_AMOUNT3 = rarioCurrencyManager.convert("USD", "USDC", 7000) -> 69986002
Transaction sent: 0xf590af46d545f545a108ae8825c52f6290317be3606dc0cafb53bf46ef4f4f686
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 33
  ERC20Token.transfer confirmed   Block: 13499852   Gas used: 51061 (0.76%)

Transaction sent: 0xd546de84ab54e5c76e9bd387e83b3c09e61abf98cdd6fcbc64e3a0ff2469c10c
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  ERC20Token.approve confirmed   Block: 13499853   Gas used: 44127 (0.66%)

usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user3.address) -> 0
usdc.balanceOf(user4.address) -> 69986003
usdc.balanceOf(rarioMarketplace.address) -> 59988003
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user4.address, "USDC", USDC_AMOUNT3 + 1, {'from': user4})
Transaction sent: 0x8aa4091b0fefcc41d20c8169f5e856c0bad8ac49f69a17b40bf1dafac7fb4d51
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed   Block: 13499854   Gas used: 149508 (2.22%)

rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x54aF1a5AE93743D421b79368936c008919c435
47', '0x54aF1a5AE93743D421b79368936c008919c43547', '0xf3d2FD3999C4Da19C7b08C5867b4985B5B41DF0D', '0xf3d2FD3999C4Da19C7b08C5867b4985B5B41DF0D', 7000, 2592000, 900, 1637935150)
usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user3.address) -> 0
usdc.balanceOf(user4.address) -> 0
usdc.balanceOf(rarioMarketplace.address) -> 129974006
token.ownerOf(1) -> 0xb897bFDeC756b5AEA01f42B1Ad691Bb962383f8
user3.address -> 0xf3d2FD3999C4Da19C7b08C5867b4985B5B41DF0D
```

It is worth mentioning that this issue does not occur when the payment is done with NATIVE currency. In this case, the funds are sent back to the outbid user. The issue only happens when the payment is done with TOKENS.

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

It is recommended that the smart contract sends back the bid tokens to the outbid user once a higher bid is placed.

Remediation Plan:

SOLVED: Rario team rightly modified the placeBid function and now refunds are given to the outbid users.

```
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user2.address, "USDC", USDC_AMOUNT + 1, {'from': user2})
Transaction sent: 0x0fdbc853ca2ca387d393e9dc6d2e03c6d9ccdf696b1bddc7619d50561975b51b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed   Block: 13575147   Gas used: 183068 (2.72%)

rarioMarketplace.getListingDetail(78541660797044910968829902406342334108369226379826116161446442989268089806461, True) -> (3, 1, 5000, 'USD', 49990002, 'USDC', '0x496a6Ae6Fca6A66fa23BC719aF9FD87ABFE758c9', '0x496a6Ae6Fca6A66fa23BC719aF9F
D87ABFE758c9', 2592000)
usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user2.address) -> 0
usdc.balanceOf(rarioMarketplace.address) -> 49990002
token.ownerOf(1) -> 0xd73476316Da797f08B81Dbb36b934201d104E78A
user2.address -> 0xe8206D9AB773cda109320A74f5239650C0b5bf2E
USDC_AMOUNT2 = rarioCurrencyManager.convert("USD", "USDC", 6000) -> 59988002
Transaction sent: 0x087dd8450f29b82dbe9ab571cb6e691dac2f0cddb4703b37166ec6f3034a2a9a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 30
  ERC20Token.transfer confirmed   Block: 13575148   Gas used: 51061 (0.76%)

Transaction sent: 0xa3cdf2012150668c3b4508bb00254e06d9b3e06ceca0e858de48a339147791b0
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  ERC20Token.approve confirmed   Block: 13575149   Gas used: 44127 (0.66%)

usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user2.address) -> 0
usdc.balanceOf(user3.address) -> 59988003
usdc.balanceOf(rarioMarketplace.address) -> 49990002
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user3.address, "USDC", USDC_AMOUNT2 + 1, {'from': user3})
Transaction sent: 0x00523dd18a2f41c820529aab877d2d2523a1977c032769c60d53c6051c6a15fa
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed   Block: 13575150   Gas used: 169841 (2.53%)

rarioMarketplace.getListingDetail(78541660797044910968829902406342334108369226379826116161446442989268089806461, True) -> (3, 1, 6000, 'USD', 59988003, 'USDC', '0x496a6Ae6Fca6A66fa23BC719aF9FD87ABFE758c9', '0x496a6Ae6Fca6A66fa23BC719aF9F
D87ABFE758c9', 2592000)
usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user2.address) -> 49990002
usdc.balanceOf(user3.address) -> 0
usdc.balanceOf(rarioMarketplace.address) -> 59988003
token.ownerOf(1) -> 0xd73476316Da797f08B81Dbb36b934201d104E78A
user3.address -> 0xBf67e3e6E47c464d5169C41954c49B4bC9787C94
USDC_AMOUNT3 = rarioCurrencyManager.convert("USD", "USDC", 7000) -> 69986002
```

FINDINGS & TECH DETAILS

3.3 (HAL-03) UNCHECKED TRANSFER - MEDIUM

Description:

In the contract Marketplace.sol the return value of some external transfer/transferFrom calls are not checked. Several tokens do not revert in case of failure and return false. If one of these tokens is used, a deposit would not revert if the transfer fails, and an attacker could deposit tokens for free.

Code Location:

- Line 370: IERC20(currencyContractAddress).transferFrom(msgSender(), address(this), bidValue);
- Line 611: IERC20(contractAddress).transferFrom(msgSender(), _treasury, platformPayableAmount);
- Line 612: IERC20(contractAddress).transferFrom(msgSender(), seller, sellerPayableAmount);

Risk Level:

Likelihood - 2
Impact - 4

Recommendation:

It is recommended to use SafeERC20.

Remediation Plan:

SOLVED: Rario team uses now SafeERC20 in all the transfers.

3.4 (HAL-04) USE OF TRANSFER WHEN TRANSFERRING NFTS - LOW

Description:

In the contract Marketplace.sol the function IERC721(_trustedToken).transferFrom is called when transferring the NFT:

- Line 351: IERC721(_trustedToken).transferFrom(address(this), sellerTokenWallet, tokenId);
- Line 539: IERC721(_trustedToken).transferFrom(_msgSender(), address(this), tokenId);
- Line 622: IERC721(_trustedToken).transferFrom(address(this), listing.buyerTokenWallet, listing.tokenId);

However, this function does not check whether the recipient is aware of the ERC721 protocol and calls _transfer directly. If the recipient is a contract not aware of incoming NFTs, then the transferred NFT would be locked in the recipient forever.

Risk Level:

Likelihood - 2
Impact - 2

Recommendation:

It is recommended to use SafeTransferFrom which internally calls _safeTransfer checking if the recipient contract implements the onERC721Received interface to avoid loss of NFTs.

Remediation Plan:

SOLVED: Rario team uses IERC721.safeTransferFrom when sending the token from the smart contract to any user/wallet:

```
IERC721(_trustedToken).safeTransferFrom(address(this), listing.
buyerTokenWallet, listing.tokenId);
```

At the same time, to avoid implementing the `onERC721Received` function, Rario team has decided to keep using `IERC721.transfer` when transferring tokens from any user to their smart contract address:

```
IERC721(_trustedToken).transferFrom(_msgSender(), address(this),
tokenId);
```

3.5 (HAL-05) UPDATEUSER FUNCTION MISSING REQUIRE STATEMENT - LOW

Description:

In the contract UserDB.sol the function addUser checks that the wallet address has not been added previously:

Listing 3: UserDB.sol - addUser (Lines 45)

```
41 function addUser(uint256 rarioId, address walletAddress) external
       override {
42     _checkRole(OPERATOR, msgSender());
43     require(walletAddress != address(0), ERR_ZERO_ADDRESS_WALLET);
44     require(!rarioIds.contains(rarioId), ERR_USER_ALREADY_EXISTS);
45     require(walletAddressToRarioId[walletAddress] == 0,
           ERR_WALLET_ALREADY_EXISTS);
46
47     rarioIds.add(rarioId);
48     rarioIdToWalletAddress[rarioId] = walletAddress;
49     walletAddressToRarioId[walletAddress] = rarioId;
50     emit UserAdded(rarioId, walletAddress);
51 }
```

The function updateUser allows to update that address but is missing that validation:

Listing 4: UserDB.sol - updateUser

```
140 function updateUser(uint256 rarioId, address walletAddress)
       external override {
141     _checkRole(OPERATOR, msgSender());
142     require(walletAddress != address(0), ERR_ZERO_ADDRESS_WALLET);
143     require(rarioIds.contains(rarioId), ERR_USER_DOES_NOT_EXIST);
144
145     address oldAddress = rarioIdToWalletAddress[rarioId];
146
147     rarioIdToWalletAddress[rarioId] = walletAddress;
148     walletAddressToRarioId[walletAddress] = rarioId;
149     delete walletAddressToRarioId[oldAddress];
150
```

```
151     emit UserUpdated(rarioId, oldAddress, walletAddress);
152 }
```

This can cause some inconsistencies, for example, with the function getUserByAddress:

```
>>> rariousers.getUserById(1)
(, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserById(2)
(, "0x3BDE98b2F0f540b3Bda46ca679bb288E01D0D61B")
>>> rariousers.updateUser(2, user1.address)
Transaction sent: 0xb1e17a4a9c3e28486698a4a5fb0b1abf3ca1d1bb4c45bb1b5d13df1216b5053f
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 29
  RarioUsers.updateUser confirmed   Block: 13511179   Gas used: 29009 (0.43%)

<Transaction '0xb1e17a4a9c3e28486698a4a5fb0b1abf3ca1d1bb4c45bb1b5d13df1216b5053f'>
>>> rariousers.getUserById(1)
(, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserById(2)
(, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserByAddress(user1.address)
(, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

It is recommended to add the missing require statement to the updateUser function.

Remediation Plan:

SOLVED: Rario team added the missing require statement:

Listing 5: UserDB.sol - updateUser (Lines 144)

```
140 function updateUser(uint256 rarioId, address walletAddress)
        external override {
141     _checkRole(OPERATOR, _msgSender());
142     require(walletAddress != address(0), ERR_ZERO_ADDRESS_WALLET);
143     require(rarioIds.contains(rarioId), ERR_USER_DOES_NOT_EXIST);
144     require(walletAddressToRarioId[walletAddress] == 0,
            ERR_WALLET_ALREADY_EXISTS);
145
146     address oldAddress = rarioIdToWalletAddress[rarioId];
```

```
147
148     rarioIdToWalletAddress[rarioId] = walletAddress;
149     walletAddressToRarioId[walletAddress] = rarioId;
150     delete walletAddressToRarioId[oldAddress];
151
152     emit UserUpdated(rarioId, oldAddress, walletAddress);
153 }
```

3.6 (HAL-06) MINT FUNCTION IN RARIOPOLYGONNFT DOES NOT VERIFY IF TOKEN WAS WITHDRAWN - INFORMATIONAL

Description:

The contract RarioPolygonNFT, as per the documentation, contains the functions deposit, withdraw and mint. The mint function is inherited from RarioNFT and contains the following code:

Listing 6: RarioNFT.sol - mint

```
39 function mint(address to, uint256 tokenId) external virtual {
40     _checkRole(OPERATOR, msgSender());
41     _mint(to, tokenId);
42 }
```

The function does not verify if the token was withdrawn previously as it is done in ChildMintableERC721.sol:

Listing 7: ChildMintableERC721.sol - mint (Lines 2268,2273)

```
2264 /**
2265  * @notice Example function to handle minting tokens on matic chain
2266  * @dev Minting can be done as per requirement,
2267  * This implementation allows only admin to mint tokens but it can be changed as per requirement
2268  * Should verify if token is withdrawn by checking `withdrawnTokens` mapping
2269  * @param user user for whom tokens are being minted
2270  * @param tokenId tokenId to mint
2271  */
2272 function mint(address user, uint256 tokenId) public only(DEFAULT_ADMIN_ROLE) {
2273     require(!withdrawnTokens[tokenId], "ChildMintableERC721: TOKEN_EXISTS_ON_ROOT_CHAIN");
```

```
2274     _mint(user, tokenId);
2275 }
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to check in RarioPolygonNFT if the token was withdrawn before minting.

Remediation Plan:

SOLVED: Rario team modified the mint function in RarioPolygonNFT. Now the function checks if the token was withdrawn before minting:

Listing 8: RarioPolygonNFT.sol - mint (Lines 34)

```
32 function mint(address user, uint256 tokenId) external override {
33  _checkRole(OPERATOR, _msgSender());
34  require(!withdrawnTokens[tokenId], "ChildMintableERC721:
        TOKEN_EXISTS_ON_ROOT_CHAIN");
35  _mint(user, tokenId);
36 }
```

3.7 (HAL-07) POSSIBLE MISUSE OF PUBLIC FUNCTIONS - INFORMATIONAL

Description:

In the contract Marketplace.sol there are functions marked as public but they are never directly called within the same contract or in any of its descendants:

Marketplace.sol
- getPendingWithdrawal(address) (Marketplace.sol#387-389)
- withdraw() (Marketplace.sol#394-396)

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

If the function is not intended to be called internally or by descendants, it is better to mark all these functions as external to save gas.

Remediation Plan:

SOLVED: Rario team declared those functions as external.

3.8 (HAL-08) UNUSED CONSTANTS - INFORMATIONAL

Description:

The following constants are not used anywhere in the code.

Code Location:

Listing 9: ERC721NFT.sol

```
30 string internal constant E_NFT_TRANSFER_TO_NON_RECEIVER = "E03001"
       ;
```

Listing 10: Marketplace.sol

```
57 string private constant ERR_CURRENCY_MANAGER = "NFTMarketplace:
       Error in CurrencyManager";
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Check whether these constants should be used and if not remove them in order to save gas.

Remediation Plan:

SOLVED: Rario team removed these unused constants.

3.9 (HAL-09) UNUSED EVENTS - INFORMATIONAL

Description:

The following events are declared but they are not emitted by any function:

Listing 11: MetadataDB.sol

```
105 event TokenStringValueUpdated(uint256 indexed tokenId, string
      indexed attribute, string value);
106 event TokenNumberValueUpdated(uint256 indexed tokenId, string
      indexed attribute, int256 value);
107 event TokenBooleanValueUpdated(uint256 indexed tokenId, string
      indexed attribute,bool value);
```

Listing 12: Marketplace.sol

```
140 event WithdrawPending(address indexed user, uint256 amount);
```

Listing 13: CurrencyManager.sol

```
 97 event BaseCurrencyUpdated(string indexed symbol);
```

Listing 14: MarketplaceConfig.sol

```
 49 event TokenPriceFeedUpdated(address indexed newAddress, address
      indexed oldAddress, string symbol);
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Check whether these events should be used and if not remove them.

Remediation Plan:

SOLVED: Rario team removed all the unused events from their smart contracts.

MANUAL TESTING

4.1 INTRODUCTION

Halborn performed different manual tests in all the contracts trying to find logic flaws and vulnerabilities that were not detected by the automatic tools.

During the manual testing multiple questions were considered while evaluating each of the defined functions:

- Can it be re-called changing admin/roles and permissions?
- Can somehow an external controlled contract call again the function during the execution of it? (Re-entrancy)
- Can a function be called twice in the same block causing issues?
- Do we control sensitive or vulnerable parameters?
- Does the function check for boundaries on the parameters and internal values? Bigger than zero or equal? Argument count, array sizes, integer truncation. . .
- Are the function parameters and variables controlled by external contracts?
- Can extended contracts cause issues on the extender contract?

4.2 UPGRADEABLE CONTRACTS' ARCHITECTURE

As our first test, we validated that the upgradeable contracts' architecture was correct. This is the hierarchy of the contract Upgradeable:

1. ERC1967Upgrade: This abstract contract provides getters and event emitting update functions for EIP1967 slots.
2. ERC1822UUPS: Inherits from ERC1967Upgrade. This contract is used as an upgradeability mechanism designed for UUPS proxies. Correctly initializes ERC1967Upgrade:

Listing 15: ERC1822UUPS.sol - initializeERC1822UUPS (Lines 22)

```
21 function initializeERC1822UUPS() internal initializer {
22  initializeERC1967Upgrade();
23 }
```

3. Upgradeable: Inherits from ERC1822UUPS, Constants, Initializable, AccessRoles and Pausable. Correctly initializes all of its parent contracts:

Listing 16: Upgradeable.sol - initializeUpgradeable

```
16 function initializeUpgradeable(address trustedForwarder) internal
       initializer {
17     setTrustedForwarder(trustedForwarder);
18     initializeAccessControl();
19     initializePausable();
20     initializeERC1822UUPS();
21
22     _setupRole(DEFAULT_ADMIN_ROLE, msgSender());
23     _setupRole(SUPER_RARIO, msgSender());
24     _setupRole(RARIO_ADMIN, msgSender());
25     _setupRole(OPERATOR, msgSender());
26 }
```

Moreover, it overrides ERC1822UUPS._authorizeUpgrade function, adding the modifier whenPaused and checking that msgSender() has SUPER_RARIO role. This way the contract will only be able to be upgraded by members of the SUPER_RARIO group and only if the contract is paused.

Listing 17: Upgradeable.sol - authorizeUpgrade

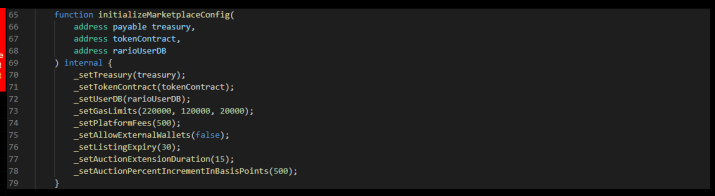
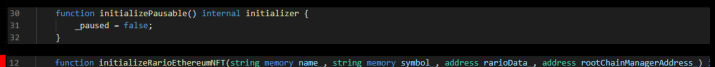
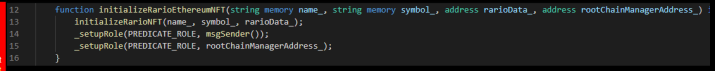
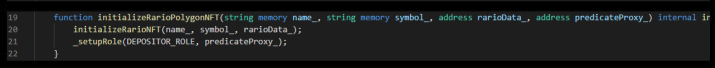
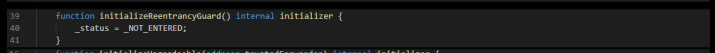
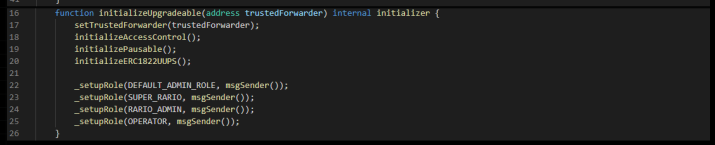
```
43 function _authorizeUpgrade(address) internal view override
       whenPaused {
44     _checkRole(SUPER_RARIO, msgSender());
45 }
```

4.3 CONTRACTS INITIALIZATION

No issues found on the initialization. Although:

1. `CurrencyManager`, `ERC721NFT`, `MarketplaceConfig` and `RarioEthereumNFT` init functions should be kept as internal as they are missing the `initializer` modifier.
2. We recommend calling `initializeReentrancyGuard()` in the contract `Marketplace initializeMarketplace` function.

Contract	Inherits from	Has initialize function?	Has initializer modifier?	Code
RarioCurrencyManager	CurrencyManager	Yes	Yes	`function initialize(address trustedForwarder) external initializer { initializeUpgradeable(trustedForwarder); }`
RarioData	MetadataDB	Yes, MetadataDB has no initalize function, hence it is not initialized in RarioData	Yes	`function initialize(address trustedForwarder) external initializer { initializeUpgradeable(trustedForwarder); }`
RarioEthereumToken	RarioEthereumNFT	Yes	Yes	`function initialize( address trustedForwarder_, address rarioData_, address rootChainManagerAddress_, string calldata name_, string calldata symbol_ ) external initializer { initializeUpgradeable(trustedForwarder_); initializeRarioEthereumNFT(name_, symbol_, rarioData_, rootChainManagerAddress_); }`
RarioMarketplace	Marketplace	Yes	Yes	`function initialize(address trustedForwarder, address configManager, address currencyManager) external initializer { initializeUpgradeable(trustedForwarder); initializeMarketplace(configManager, currencyManager); name = "RarioMarketplace"; version = "3"; }`
RarioMarketplaceConfig	MarketplaceConfig	Yes	Yes	`function initialize( address trustedForwarder, address payable treasury, address tokenContract, address rarioUserDB ) external initializer { initializeUpgradeable(trustedForwarder); initializeMarketplaceConfig(treasury, tokenContract, rarioUserDB); }`
RarioPolygonToken	RarioPolygonNFT	Yes	Yes	`function initialize( address trustedForwarder_, address rarioData_, address predicateProxy_, string calldata name_, string calldata symbol_ ) external initializer { initializeUpgradeable(trustedForwarder_); initializeRarioPolygonNFT(name_, symbol_, rarioData_, predicateProxy_); }`
RarioUsers	UserDB	Yes, UserDB has no initalize function, hence it is not initialized in RarioUsers	Yes	`function initialize(address trustedForwarder) external initializer { initializeUpgradeable(trustedForwarder); }`
AccessControl	Initializable, Context, IAccessControl, IAccessControlEnumerable, ERC165Interface	Yes, internal function. It is called in the Upgradeable contract	Yes	`function initializeAccessControl() internal initializer { registerInterface(type(IAccessControl).interfaceId); registerInterface(type(IAccessControlEnumerable).interfaceId); }`
Context	-	No	-	-
CurrencyManager	Upgradeable, ICurrencyManager	Yes	No, although it is internal and is called by RarioCurrencyManager init function which is correctly protected with the initializer modifier, hence it is safe as long as it is kept as an internal function	`function initializeCurrencyManager(string calldata nativeCurrencySymbol) internal { _setFiatDecimals(2); _addCurrency(CurrencyType.FIAT, "USD", address(0), _fiatDecimals); _nativeCurrency = nativeCurrencySymbol; }`
ERC165Interface	Initializable, IERC165	Yes	Yes	`function initializeERC165Interface() internal initializer { registerInterface(type(IERC165).interfaceId); }`
ERC721NFT	Upgradeable, IERC721, IERC721Metadata, IERC721Enumerable	Yes	No, although it is internal and is called by RarioNFT init function which is correctly protected with the initializer modifier, hence it is safe as long as it is kept as an internal function	`function initializeERC721NFT(string memory name_, string memory symbol_) internal { require(bytes(name_).length > 0, E_NFT_EMPTY_TOKEN_NAME); require(bytes(symbol_).length > 0, E_NFT_EMPTY_TOKEN_SYMBOL); _name = name_; _symbol = symbol_; initializeERC165Interface(); registerInterface(type(IERC721).interfaceId); registerInterface(type(IERC721Metadata).interfaceId); registerInterface(type(IERC721Enumerable).interfaceId); }`
ERC1822UUPS	Initializable, ERC1967Upgrade	Yes	Yes	`function initializeERC1822UUPS() internal initializer { initializeERC1967Upgrade(); }`
ERC1967Upgrade	Initializable	Yes	Yes	`function initializeERC1967Upgrade() internal initializer {}`
Initializable	-	No	-	-
Marketplace	Upgradeable, ReentrancyGuard	Yes, although ReentrancyGuard is not initialized by calling initializeReentrancyGuard(). It is recommended to initialize the ReentrancyGuard contract as well even if the current code has no security impact	Yes	`function initializeMarketplace(address config, address currencyManager) internal initializer { _config = IMarketplaceConfig(config); _currencyManager = ICurrencyManager(currencyManager); _hasConfiguration = false; }`

MarketplaceConfig	Upgradeable, IMarketplaceConfig	Yes	No, although it is internal and is called by RarioMarketplaceConfig init function which is correctly protected with the initializor modifier, hence it is safe as long as it is kept as an internal function	
MetadataDB	Upgradeable, IMetadataDB	No, although it is worth mentioning that initializeUpgradeable is called as expected by all the contracts that inherits from MetadataDB, for example, RarioData	-	-
Pausable	Initializable, Context	Yes, internal function. It is called in the Upgradeable contract	Yes	
RarioEthereumNFT	IMintableERC721, RarioNFT	Yes, internal function. It is called in the RarioEthereumToken contract	No, although it is internal and is called by RarioEthereumToken init function which is correctly protected with the initializer modifier, hence it is safe as long as it is kept as an internal function	
RarioNFT	ERC721NFT	Yes	Yes	
RarioPolygonNFT	IPolygonChildToken, RarioNFT	Yes, internal function. It is called in the RarioPolygonToken contract	Yes	
ReentrancyGuard	Initializable	Yes	Yes	
Upgradeable	Context, AccessControl, Pausable, ERC1822UUPS	Yes, internal function. It is called all the contracts that inherits from this one	Yes	
UserDB	Upgradeable, IUserDB	No, although it is worth mentioning that initializeUpgradeable is called as expected by all the contracts that inherits from UserDB, for example, RarioUsers	-	-

4.4 RARIOCURRENCYMANAGER CONTRACT

The contract RarioCurrencyManager inherits from CurrencyManager. This contract is used to store all the different attributes related to the tokens.

The contract contains the following functions:

- From CurrencyManager
  - _addCurrency(CurrencyManager.CurrencyType,string,address,uint8) (private)
  - _convert(string,string,uint256) (private)
  - _getCurrency(string) (private)
  - _getCurrencyPair(string) (private)
  - _memcmp(bytes,bytes) (internal)
  - _removeCurrency(string) (private)
  - _removeCurrencyPair(string) (private)
  - _setFiatDecimals(uint8) (private)
  - _strcmp(string,string) (internal)
  - addCryptoCurrency(string,address,uint8) (external)
  - addCurrencyPair(string,string,string,bool,uint8,address) (external)
  - addFiatCurrency(string) (external)
  - convert(string,string,uint256) (external)
  - getCurrency(string) (external)
  - getCurrencyByIndex(uint256) (external)
  - getCurrencyPair(string) (external)
  - getCurrencyPairByIndex(uint256) (external)
  - getFiatDecimals() (external)
  - getNativeCurrency() (external)
  - initializeCurrencyManager(string) (internal)
  - removeCurrency(string) (external)
  - removeCurrencyPair(string) (external)
  - setFiatDecimals(uint8) (external)
  - supportedCurrencies() (external)
  - supportedCurrencyPairs() (external)

- `supportsCurrency(string)` (external)
- `supportsCurrencyPair(string)` (external)
- `totalSupportedCurrencies()` (external)
- `totalSupportedCurrencyPairs()` (external)

- From RarioCurrencyManager
  - `initialize(address,string)` (external)

As this contract was really simple our manual tests focused on checking that all the getter and setter functions were working as expected:

```
Contract RarioCurrencyManager (rarioCurrencyManager) deployed at 0x1AF81ED448b89aa0C5224051dbe6ED02f471Cd2B

Transaction sent: 0x4ec69e368c5c99101a1ed72de81e62ad7df288b7fd326eb3e0231c1e984d697
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 4
  RarioCurrencyManager.initialize confirmed   Block: 13492144   Gas used: 620118 (9.23%)

Contract RarioCurrencyManager initialized

Calling -> rarioCurrencyManager.addFiatCurrency("EURO", {'from': owner})
Transaction sent: 0xa00723377fd8cfed0d65513d2d3bca5b990f11ef30e8848266448ca0bbbe7b5a9
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 5
  RarioCurrencyManager.addFiatCurrency confirmed   Block: 13492145   Gas used: 123829 (1.84%)

Calling -> rarioCurrencyManager.addFiatCurrency("DOLLAR", {'from': owner})
Transaction sent: 0xa04407cdcb0fe4052aa81e0f8bfd1a39c66fad0cf21f76e4b77c41ef16db9b3e
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 6
  RarioCurrencyManager.addFiatCurrency confirmed   Block: 13492146   Gas used: 123853 (1.84%)

rarioCurrencyManager.getNativeCurrency() -> MATIC

Calling -> rarioCurrencyManager.addCryptoCurrency("MATIC", usdc.address, 18), {'from': owner})
Transaction sent: 0x9fa1a3898447f055555511668da17752a63e348239f02b551f93750cf8183820
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 7
  RarioCurrencyManager.addCryptoCurrency confirmed   Block: 13492147   Gas used: 146210 (2.18%)

Calling -> rarioCurrencyManager.addCurrencyPair("MATICUSD", "MATIC", "USD", False, 8, maticFeed.address, {'from': owner})
Transaction sent: 0x4503cb00d7351930906fb77ed460dc2e0accbebd7e81959bef92db4a9fcfe724
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 8
  RarioCurrencyManager.addCurrencyPair confirmed   Block: 13492148   Gas used: 184994 (2.75%)

rarioCurrencyManager.getFiatDecimals() -> 2
Calling -> rarioCurrencyManager.setFiatDecimals(5, {'from': owner})
Transaction sent: 0x0d535a9f33b0a9c024b7a6abe97a1e0b9788e94c5a7f68c89b6c9a44aa37b9ba
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 9
  RarioCurrencyManager.setFiatDecimals confirmed   Block: 13492149   Gas used: 73443 (1.09%)

rarioCurrencyManager.getFiatDecimals() -> 5
rarioCurrencyManager.getCurrency("EURO") -> ('EURO', 0, '0x0000000000000000000000000000000000000000', 5)
rarioCurrencyManager.getCurrency("DOLLAR") -> ('DOLLAR', 0, '0x0000000000000000000000000000000000000000', 5)
rarioCurrencyManager.getCurrency("MATIC") -> ('MATIC', 1, '0x0000000000000000000000000000000000000000', 18)
rarioCurrencyManager.getCurrencyByIndex(0) -> ('USD', 0, '0x0000000000000000000000000000000000000000', 5)
rarioCurrencyManager.getCurrencyByIndex(1) -> ('EURO', 0, '0x0000000000000000000000000000000000000000', 5)
rarioCurrencyManager.getCurrencyByIndex(1) -> ('DOLLAR', 0, '0x0000000000000000000000000000000000000000', 5)
rarioCurrencyManager.getCurrencyByIndex(1) -> ('MATIC', 1, '0x0000000000000000000000000000000000000000', 18)
rarioCurrencyManager.getCurrencyPair("MATICUSD") -> ('MATICUSD', 'MATIC', 'USD', False, 8, '0x07C9b67329C6b4f4E03fb4C7cc3C09183C7141dE')
maticFeed.address -> 0x07C9b67329C6b4f4E03fb4C7cc3C09183C7141dE
rarioCurrencyManager.getCurrencyPairByIndex(0) -> ('MATICUSD', 'MATIC', 'USD', False, 8, '0x07C9b67329C6b4f4E03fb4C7cc3C09183C7141dE')
rarioCurrencyManager.supportedCurrencies() -> ('USD', 'EURO', 'DOLLAR', 'MATIC')
rarioCurrencyManager.supportedCurrencyPairs() -> ('MATICUSD',)
rarioCurrencyManager.supportsCurrency("EURO") -> True
rarioCurrencyManager.supportsCurrencyPair("MATICUSD") -> True
rarioCurrencyManager.totalSupportedCurrencies() -> 4
rarioCurrencyManager.totalSupportedCurrencyPairs() -> 1
```

The setter functions can only be called by RARIO_ADMIN. The getter functions can be called by anyone, except `getFiatDecimals()` which can only be called by OPERATORS.

4.5 RARIODATA CONTRACT

The contract RarioData inherits from MetadataDB. This contract is used to store all the different attributes related to the tokens.

The contract contains the following functions:

- From MetadataDB
  - _canSetValue(uint256,string) (private)
  - _getAttribute(string) (private)
  - addAttribute(string,string,MetadataDB.AttributeType,bool) (external)
  - depricate(string) (external)
  - existsAttribute(string) (external)
  - getAttribute(string) (external)
  - getAttributeByIndex(uint256) (external)
  - getAttributeCount() (external)
  - getBooleanValue(uint256,string) (external)
  - getNumberValue(uint256,string) (external)
  - getTextValue(uint256,string) (external)
  - setBooleanValue(uint256,string,bool) (external)
  - setDescription(string,string) (external)
  - setNumberValue(uint256,string,int256) (external)
  - setTextValue(uint256,string,string) (external)
- From RarioData
  - initialize(address) (external)

All the setter functions in this contract can only be called by users with the OPERATOR role except the function existsAttribute(string) which can be called by anyone. On the other hand, the setter functions can only be called by users with the RARIO_ADMIN role.

We can see below how the different getter and setter functions are working as expected and also how they are restricted with the right access control policies:

MANUAL TESTING

```
Calling -> rariodata.addAttribute("label0", "description0", 0, False, {'from': owner})
Transaction sent: 0xc8da004d86598bc3e1db527b05a34e4d8389291272811298b39a1a9e60c60a7e
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 2
  RarioData.addAttribute confirmed   Block: 13492589   Gas used: 122780 (1.83%)

Calling -> rariodata.addAttribute("label1", "description1", 1, True, {'from': owner})
Transaction sent: 0xf65e680c2d49d62d67ff54547043a78cfbafc39dd488355731fae842e9e86284
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 3
  RarioData.addAttribute confirmed   Block: 13492590   Gas used: 146204 (2.18%)

Calling -> rariodata.addAttribute("label2", "description2", 2, True, {'from': owner})
Transaction sent: 0x650073c2ec5e864a3fc810468b8bf2b042fdf7c191db9efaf3e0b7be5556314
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 4
  RarioData.addAttribute confirmed   Block: 13492591   Gas used: 146204 (2.18%)

rariodata.existsAttribute("label0", {'from': owner}) -> True
rariodata.existsAttribute("label0", {'from': owner}) -> True
rariodata.getAttribute("label0", {'from': owner}) -> ('label0', 'description0', 'text', False, False)
rariodata.getAttribute("label0", {'from': owner}) -> ('label1', 'description1', 'boolean', True, False)
rariodata.getAttribute("label0", {'from': owner}) -> ('label2', 'description2', 'number', True, False)
rariodata.getAttributeByIndex(0, {'from': owner}) -> ('label0', 'description0', 'text', False, False)
rariodata.getAttributeByIndex(1, {'from': owner}) -> ('label1', 'description1', 'boolean', True, False)
rariodata.getAttributeByIndex(2, {'from': owner}) -> ('label2', 'description2', 'number', True, False)
rariodata.getAttributeCount({'from': owner}) -> 3
Calling -> rariodata.depricate("label1", {'from': owner})
Transaction sent: 0x9958c394069dbcebf59e765d0b5ef1db42a4537a322d71bdca7162b87d2e9a91
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 5
  RarioData.depricate confirmed   Block: 13492592   Gas used: 36481 (0.54%)

rariodata.getAttribute("label0", {'from': owner}) -> ('label0', 'description0', 'text', False, False)
rariodata.getAttribute("label0", {'from': owner}) -> ('label1', 'description1', 'boolean', True, True)
rariodata.getAttribute("label0", {'from': owner}) -> ('label2', 'description2', 'number', True, False)
Calling -> rariodata.setBooleanValue(1, "label0", True, {'from': owner})
Transaction sent: 0x1c1a1708a0940667aaaf215358bbd304bef4cbf527b0946c7a3790288c0ba2ef
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 6
  RarioData.setBooleanValue confirmed   Block: 13492593   Gas used: 152308 (2.27%)

rariodata.getBooleanValue(1, "label0") -> True
Calling -> rariodata.setBooleanValue(1, "label0", False, {'from': owner})
Transaction sent: 0x216cfd7e8c70b6121810c11d95a87090ab52dba58d38aa31994214ebdb8b8830
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 7
  RarioData.setBooleanValue confirmed   Block: 13492594   Gas used: 40326 (0.60%)

rariodata.getBooleanValue(1, "label0") -> False
Calling -> rariodata.setDescription("label0", "NEWDESCRIPTION", {'from': owner})
Transaction sent: 0x33aa6eee98ab2cc935d58d2b56ec6c415547803ef6ae2789b7db67de7aa226d
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 8
  RarioData.setDescription confirmed   Block: 13492595   Gas used: 42480 (0.63%)

rariodata.getAttribute("label0", {'from': owner}) -> ('label0', 'NEWDESCRIPTION', 'text', False, False)
rariodata.getAttribute("label0", {'from': owner}) -> ('label1', 'description1', 'boolean', True, True)
rariodata.getAttribute("label0", {'from': owner}) -> ('label2', 'description2', 'number', True, False)
Calling -> rariodata.setNumberValue(1, "label0", 1337, {'from': owner})
Transaction sent: 0x6da9b4cf83e3bea3c9c5826f70425184a6d8dfbd476ad425cb23a35c031c5edb
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 9
  RarioData.setNumberValue confirmed   Block: 13492596   Gas used: 69458 (1.03%)

rariodata.getNumberValue(1, "label0") -> 1337
Calling -> rariodata.setTextValue(1, "label0", "NEWVALUE", {'from': owner})
Transaction sent: 0x073d3cd0b3575187a21d1b5b2cacf6897fc5ffcad1f4897e3b9500f2c41371b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 10
  RarioData.setTextValue confirmed   Block: 13492597   Gas used: 73559 (1.09%)

rariodata.getTextValue(1, "label0") -> NEWVALUE
>>> rariodata.getAttribute("label1", {'from': user1})
  File "<console>", line 1, in <module>
  File "brownie/network/contract.py", line 1728, in __call__
    return self.call(*args, block_identifier=block_identifier)
  File "brownie/network/contract.py", line 1532, in call
    raise VirtualMachineError(e) from None
VirtualMachineError: revert: AccessControl: account 0x41d8d960d58ealebf4f8f7487fab323e177e76a3 is missing role 0x523a704056dcd17bcf83bed8b68c59416dac1119be77755efe3bde0a64e46e0c
>>> rariodata.getAttributeByIndex(1, {'from': user1})
  File "<console>", line 1, in <module>
  File "brownie/network/contract.py", line 1728, in __call__
    return self.call(*args, block_identifier=block_identifier)
  File "brownie/network/contract.py", line 1532, in call
    raise VirtualMachineError(e) from None
VirtualMachineError: revert: AccessControl: account 0x41d8d960d58ealebf4f8f7487fab323e177e76a3 is missing role 0x523a704056dcd17bcf83bed8b68c59416dac1119be77755efe3bde0a64e46e0c
```

4.6 RARIOMARKETPLACE CONTRACT

The contract RarioMarketplace inherits from Marketplace.

The contract contains the following functions:

- From Marketplace
  - _cancelListing(uint256) (private)
  - _createListing(uint256,uint256,string,address,address,address,uint8) (private)
  - _distributeFunds(address,string,uint256) (private)
  - _finaliseListing(uint256) (private)
  - _getListingByListingId(uint256,bool) (private)
  - _isValidWallet(address) (private)
  - _reloadConfiguration() (private)
  - _sendValue(address,uint256,uint256) (private)
  - _transferToken(address,address,uint256) (private)
  - cancelListing(uint256) (external)
  - createAuction(uint256,uint256,string,address) (external)
  - createPrivateSale(uint256,uint256,string,address,address) (external)
  - createPublicSale(uint256,uint256,string,address) (external)
  - finalizeListing(uint256) (external)
  - getBidPriceInCurrency(uint256,string) (external)
  - getFinalisedListing(uint256) (external)
  - getListing(uint256) (external)
  - getListingByTokenId(uint256) (external)
  - getPendingWithdrawal(address) (public)
  - initializeMarketplace(address,address) (internal)
  - placeBid(uint256,address,string,uint256) (external)
  - reloadConfiguration() (external)
  - setMarketplaceConfig(address) (external)
  - updateListing(uint256,uint256) (external)
  - withdraw() (public)
  - withdrawFor(address) (public)
- From RarioMarketplace

- initialize(address,address,address) (external)

Test 1: Creating a public sale

In this test, user1 created a public sale by calling the createPublicSale function. Initially, the price set was 1337. But then user1 called updateListing and updated the price to 3000 USD. Then, user2 bought the token successfully by calling the placeBid function. In the image below, we can see how the 95% of the tokens were transferred to user1 and the 5% to the treasury address:

On the other hand, it is worth mentioning that the parameter bidValue is useless as long as the user calling placeBid is providing a msg.value > 0:

Listing 18: Marketplace.sol (Lines 308,310)

```
297 function placeBid(
298     uint256 listingId,
299     address buyerTokenWallet,
300     string calldata currency,
301     uint256 bidValue
```

```
302 ) external payable {
303     Listing storage listing = _listings[listingId];
304     require(listing.price > 0, ERR_LISTING_NOT_EXISTS);
305     require(listing.sellerTokenWallet != buyerTokenWallet,
            ERR_CANNOT_BUY_OWN_LISTING);
306     require(_isValidWallet(buyerTokenWallet),
            ERR_UNSUPPORTED_TOKEN_WALLET);
307
308     if (msg.value > 0) {
309         require(currency.equals(_nativeCurrency), "NFTMarketplace:
                Currency Symbol is not Native currency");
310         bidValue = msg.value;
311     }
```

```
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user2.address, "MATIC", 0, {'from': user2, 'value': msgvalue})
>>> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user2.address, "MATIC", 0, {'from': user2, 'value': msgvalue})
Transaction sent: 0xa92316f630b9e33bcf8d16b766a82f0e542fc1ad6022a009a6b58172b4a699c3
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioMarketplace.placeBid confirmed   Block: 13494351   Gas used: 337576 (5.02%)

<Transaction '0xa92316f630b9e33bcf8d16b766a82f0e542fc1ad6022a009a6b58172b4a699c3'>
>>> output.redd("user1.balance() -> " + str(user1.balance()))
output.redd("treasury.balance() -> " + str(treasury.balance()))
output.redd("user2.balance() -> " + str(user2.balance()))
output.redd("token.ownerOf(1) -> " + str(token.ownerOf(1)))
output.redd("user2.address -> " + str(user2.address))
user1.balance() -> 119000000000000000000
treasury.balance() -> 101000000000000000000
user2.balance() -> 80000000000000000000
token.ownerOf(1) -> 0xf799573dA457653AB8f5f8E11c108eb6Aa0c29DF
user2.address -> 0xf799573dA457653AB8f5f8E11c108eb6Aa0c29DF
```

The payment can also be done with an ERC20 token, in this case, with USDC ERC20 tokens:

```
Calling -> token.approve(rarioMarketplace.address, 1, {'from': user1})
Transaction sent: 0xd4a7e2aad81a9ef7730ba91d9252e2b850915576292 7f68ea86ac54665d5cf15
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioPolygonToken.approve confirmed   Block: 13494566   Gas used: 48878 (0.73%)

token.ownerOf(1) -> 0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE
user1.address -> 0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE
rarioMarketplace.address -> 0x1Db79b800DD96689B3477f4e07CeBf6F77376baB
Calling -> rarioMarketplace.createPublicSale(1, 1337, "USDC", user1.address, {'from': user1})
Transaction sent: 0x00407a762725437af4f2d2fd575fc1fc8cbad7e4655440dd5e7fb32e1de16b98
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.createPublicSale confirmed   Block: 13494567   Gas used: 325166 (4.84%)

token.ownerOf(1) -> 0x1Db79b800DD96689B3477f4e07CeBf6F77376baB
user1.address -> 0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE
rarioMarketplace.address -> 0x1Db79b800DD96689B3477f4e07CeBf6F77376baB
rarioMarketplace.getListingByTokenId(1) -> (1, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE', '0xFEc4da8636
00000000000000000000000000', '0x0000000000000000000000000000000000000000', 1337, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (1, 78541660797044910968829902406342334108369226379826116161446442989268089806
AE', '0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE', '0x0000000000000000000000000000000000000000', '0x0000000000000000000000000000000000000000', 1337, 2592000, 900, 0)
Calling -> rarioMarketplace.updateListing(78541660797044910968829902406342334108369226379826116161446442989268089806461, 3000, {'from': user1})
Transaction sent: 0x38d79280135c6b97cad8c616c0a0a239e4a76129063fa92f1e9a3a6ca4ebae18
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 2
  RarioMarketplace.updateListing confirmed   Block: 13494568   Gas used: 35420 (0.53%)

rarioMarketplace.getListingByTokenId(1) -> (1, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE', '0xFEc4da8636
00000000000000000000000000', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (1, 78541660797044910968829902406342334108369226379826116161446442989268089806
AE', '0xFEc4da86367cfc6f7C52e3F8cc2a8bc6C67601AE', '0x0000000000000000000000000000000000000000', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
USDC_AMOUNT = rarioCurrencyManager.convert("USD", "USDC", 3000) -> 29994001
Transaction sent: 0x62d1ed7a7d7935a8023cd880e2ba3ac4337c3c7212998ab23d9df1c394b031fe
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 29
  ERC20Token.transfer confirmed   Block: 13494569   Gas used: 51061 (0.76%)

Transaction sent: 0xe435d29828daa6338459ce89ecc091efa4ccc7b7e5d639107f9cad06b3f02ba0
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  ERC20Token.approve confirmed   Block: 13494570   Gas used: 44127 (0.66%)

usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user2.address) -> 29994002
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user2.address, "USDC", USDC_AMOUNT + 1, {'from': user2})
Transaction sent: 0x3472ec06f134d8c83cab1087dcb9d6add3fd512e421b14a186eb550cc99c4e0a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed   Block: 13494571   Gas used: 361694 (5.38%)

usdc.balanceOf(user1.address) -> 28494302
usdc.balanceOf(treasury.address) -> 1499700
usdc.balanceOf(user2.address) -> 0
token.ownerOf(1) -> 0x55C48a7251F2185d5651B07482B35da5F19BC154
user2.address -> 0x55C48a7251F2185d5651B07482B35da5F19BC154
```

Test 2: Creating a private sale

In this test, user1 created a private sale by calling the `createPrivateSale` function setting the user2 as the buyer. The price set was 3000. Then, user2 bought the token successfully by calling the `placeBid` function:

```
Minting tokenid 1 to user1
Transaction sent: 0xe9c08e1371c2f110e0ee1b897a3f1d272ab9abaacb9f3f5a9e3352808119855e
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 26
  RarioPolygonToken.mint confirmed   Block: 13498056   Gas used: 134496 (2.00%)

token.ownerOf(1) -> 0x1D01835A7de31eAcFd16FFbF2307263eAE095218
user1.address -> 0x1D01835A7de31eAcFd16FFbF2307263eAE095218

Calling -> rariousers.addUser(1, user1.address, {'from': owner})
Transaction sent: 0xdf72b2d3ebbd1d751a1ddf486660caa39e379cad035f215c836b137f524c1aca
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 27
  RarioUsers.addUser confirmed   Block: 13498057   Gas used: 132294 (1.97%)

Calling -> rariousers.addUser(2, user2.address, {'from': owner})
Transaction sent: 0xa827ec1f0e50e235a3de008c1f4fcdd390b0bbd98aef8195c192dcc5ac0d748a
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 28
  RarioUsers.addUser confirmed   Block: 13498058   Gas used: 117294 (1.74%)

Calling -> token.approve(rarioMarketplace.address, 1, {'from': user1})
Transaction sent: 0xfa0836105d16fde27048deded6c9acb31ea3e4a6840962a5e053ab578cf06f0f
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioPolygonToken.approve confirmed   Block: 13498059   Gas used: 48878 (0.73%)

token.ownerOf(1) -> 0x1D01835A7de31eAcFd16FFbF2307263eAE095218
user1.address -> 0x1D01835A7de31eAcFd16FFbF2307263eAE095218
rarioMarketplace.address -> 0xDF9843DEC2d08fa64fC5989ffc721eF1d9047b0d
Calling -> rarioMarketplace.createPrivateSale(1, 3000, "USD", user1.address, user2.address, {'from': user1})
Transaction sent: 0xafbbc5f29627d9417d038870e4259eae438c6e3f40f08db5b2e03b78df8c26ac
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.createPrivateSale confirmed   Block: 13498060   Gas used: 325581 (4.84%)

token.ownerOf(1) -> 0xDF9843DEC2d08fa64fC5989ffc721eF1d9047b0d
user1.address -> 0x1D01835A7de31eAcFd16FFbF2307263eAE095218
rarioMarketplace.address -> 0xDF9843DEC2d08fa64fC5989ffc721eF1d9047b0d
rarioMarketplace.getListingByTokenId(1) -> (0, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x1D01835A7de31eAcFd16FFbF2307263eAE095218', '0x1D01835A7
51cb686566A8767C0a6dfbb8202', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (0, 78541660797044910968829902406342334108369226379826116161446442989268089806
18', '0x1D01835A7de31eAcFd16FFbF2307263eAE095218', '0xF003d5e00907d51cb686566A8767C0a6dfbb8202', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
USDC_AMOUNT = rarioCurrencyManager.convert("USD", "USDC", 3000) -> 29994001
Transaction sent: 0xbea6c5b3449fae57d593ec87e70201ce042cf04ad00f2af2c5e7e50d5cb34f26
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 29
  ERC20Token.transfer confirmed   Block: 13498061   Gas used: 51061 (0.76%)

Transaction sent: 0x1842e7adf8237bec36befdc141d66dbaf0b9ab84706daa8f7ff230197ef6d999
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  ERC20Token.approve confirmed   Block: 13498062   Gas used: 44127 (0.66%)

usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user2.address) -> 29994002
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user2.address, "USDC", USDC_AMOUNT + 1, {'from': user2})
Transaction sent: 0x79d8eaee558222efb9d335d879a166ded3f1fd2feb7387d1a7b80f31cfe8c5b6
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed   Block: 13498063   Gas used: 342563 (5.10%)

usdc.balanceOf(user1.address) -> 28494302
usdc.balanceOf(treasury.address) -> 1499700
usdc.balanceOf(user2.address) -> 0
token.ownerOf(1) -> 0xF003d5e00907d51cb686566A8767C0a6dfbb8202
user2.address -> 0xF003d5e00907d51cb686566A8767C0a6dfbb8202
```

We can see how the sale is restricted just for the user2:

```
usdc.balanceOf(user1.address) -> 0
usdc.balanceOf(treasury.address) -> 0
usdc.balanceOf(user3.address) -> 29994002
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user3.address, "USDC", USDC_AMOUNT + 1, {'from': user3})
Transaction sent: 0x827695dda278e2f2f1bca123243dd2dc8a62de7c73fd5375ab3fe26f37629480
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.placeBid confirmed (NFTMarketplace: Private Sale)   Block: 13498197   Gas used: 93500 (1.39%)
```

Test 3: Creating an auction, payment with NATIVE currency

In this test, user1 created an auction sale by calling the createAuction function. Initially the price set was 3000. But then user1 called updateListing and updated the price to 5000 USD. Then multiple users placed a bid using NATIVE currency:


```
Calling -> rarioMarketplace.createAuction(1, 3000, "USD", user1.address, {'from': user1})
Transaction sent: 0xb2fbd7e040c685c0fed95d1fa936e452baafc03613fcc213dc70c6a12918a18
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.createAuction confirmed   Block: 13501203   Gas used: 325220 (4.84%)

token.ownerOf(1) -> 0x45E1eC087ee7EaAli8E9d66b2F96512955e74C3f
user1.address -> 0x2487Af7590848cfd2FD309d054B3AE35248c2740
rarioMarketplace.address -> 0x45E1eC087ee7EaAli8E9d66b2F96512955e74C3f
rarioMarketplace.getListingByTokenId(1) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
Calling -> rarioMarketplace.updateListing(listingID, 5000, {'from': user1})
Transaction sent: 0x5b5833c935b54e77bc7201c6b7947b3c607a6c665c77ca967a2be0f5ee362
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 2
  RarioMarketplace.updateListing confirmed   Block: 13501204   Gas used: 35420 (0.53%)

rarioMarketplace.getListingByTokenId(1) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x0000000000000000000000000000000000000000', 5000, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x0000000000000000000000000000000000000000', 5000, 2592000, 900, 0)


MSGVALUE1 = rarioCurrencyManager.convert("USD", "MATIC", 5000) -> 3333333333333333333
user1.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 100000000000000000000
rarioMarketplace.balance() -> 0
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user2.address, "MATIC", 0, {'from': user2, 'value': MSGVALUE1 + 1})
Transaction sent: 0x307b196d2fa7cc6d3c4037c95b9df88c3c5fe9bd28f4faa66dbf10d9a514e973
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioMarketplace.placeBid confirmed   Block: 13501205   Gas used: 134941 (2.01%)

rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x45C46f68C017B61c2B194d39DEed4B15c36E10BE', 5000, 2592000, 900, 1637954065)
user1.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 666666666666666666
rarioMarketplace.balance() -> 3333333333333333334
token.ownerOf(1) -> 0x45E1eC087ee7EaAli8E9d66b2F96512955e74C3f
user2.address -> 0x45C46f68C017B61c2B194d39DEed4B15c36E10BE


MSGVALUE2 = rarioCurrencyManager.convert("USD", "MATIC", 6000) -> 4000000000000000000
user2.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 666666666666666666
user3.balance() -> 100000000000000000000
rarioMarketplace.balance() -> 3333333333333333334
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user3.address, "MATIC", 0, {'from': user3, 'value': MSGVALUE2 + 1})
Transaction sent: 0x02d0dbf5430c2891bf2ca37593e6ca16edcbdc72866ed67ca19b8af24908d1d
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioMarketplace.placeBid confirmed   Block: 13501206   Gas used: 146190 (2.17%)

rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0xf4823F090939De3034aaa7f4A6a5f3219605d08A', '0xf4823F090939De3034aaa7f4A6a5f3219605d08A', 6000, 2592000, 900, 1637954065)
user1.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 100000000000000000000
user3.balance() -> 599999999999999999
rarioMarketplace.balance() -> 4000000000000000001
token.ownerOf(1) -> 0x45E1eC087ee7EaAli8E9d66b2F96512955e74C3f
user3.address -> 0xf4823F090939De3034aaa7f4A6a5f3219605d08A


MSGVALUE3 = rarioCurrencyManager.convert("USD", "MATIC", 7000) -> 4666666666666666666
user1.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 100000000000000000000
user3.balance() -> 599999999999999999
user4.balance() -> 100000000000000000000
rarioMarketplace.balance() -> 4000000000000000001
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user4.address, "MATIC", 0, {'from': user4, 'value': MSGVALUE3 + 1})
Transaction sent: 0x16afab804e87ea23570c5fcbd84afdafd386bba95007cd39cf25e1011e39dbb
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioMarketplace.placeBid confirmed   Block: 13501207   Gas used: 135398 (2.01%)

rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0x2487Af7590848cfd2FD309d054B3AE35248c2740', '0x9015B120BeB9EAdbDcF2C4b3f50CeaaD104CD7c4', '0x9015B120BeB9EAdbDcF2C4b3f50CeaaD104CD7c4', 7000, 2592000, 900, 1637954065)
user1.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 100000000000000000000
user3.balance() -> 100000000000000000000
user4.balance() -> 5333333333333333333
rarioMarketplace.balance() -> 4666666666666666667
token.ownerOf(1) -> 0x45E1eC087ee7EaAli8E9d66b2F96512955e74C3f
user3.address -> 0x9015B120BeB9EAdbDcF2C4b3f50CeaaD104CD7c4
```

In the image above we can appreciate how the ETH amount was discounted from their balance and sent to the smart contract right after placing the highest bid. We can also see how the bid was returned also to the outbid users. We have also made sure that the seller cannot place any bid in his listing:

MANUAL TESTING

```
Calling -> rarioMarketplace.createAuction(1, 3000, "USD", user1.address, {'from': user1})
Transaction sent: 0x2b1970598146155ce3e378e2335bd048d9e9e85f6c7be0646fca085091d2681b
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.createAuction confirmed   Block: 13504346   Gas used: 325220 (4.84%)

token.ownerOf(1) -> 0x96c776120088314F568ec4757fb41EB75F869b4e
user1.address -> 0xa0299fA741579BAE9C525c2574A055790C39E8ab
rarioMarketplace.address -> 0x96c776120088314F568ec4757fb41EB75F869b4e
rarioMarketplace.getListingByTokenId(1) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0xa0299fA741579BAE9C525c2574A055790C39E8ab', '0xa0299fA74
0000000000000000000000000000', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806
ab', '0xa0299fA741579BAE9C525c2574A055790C39E8ab', '0x0000000000000000000000000000000000000000', '0x0000000000000000000000000000000000000000', 3000, 2592000, 900, 0)
Calling -> rarioMarketplace.updateListing(listingID, 5000, {'from': user1})
Transaction sent: 0x94438c139cdbc26bbca84a454f8967b9884e851fb5a0a6d850191481d44e41ff
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 2
  RarioMarketplace.updateListing confirmed   Block: 13504347   Gas used: 35420 (0.53%)

rarioMarketplace.getListingByTokenId(1) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806461, 1, '0xa0299fA741579BAE9C525c2574A055790C39E8ab', '0xa0299fA74
0000000000000000000000000000', '0x0000000000000000000000000000000000000000', 5000, 2592000, 900, 0)
rarioMarketplace.getListing(78541660797044910968829902406342334108369226379826116161446442989268089806461) -> (2, 78541660797044910968829902406342334108369226379826116161446442989268089806
ab', '0xa0299fA741579BAE9C525c2574A055790C39E8ab', '0x0000000000000000000000000000000000000000', '0x0000000000000000000000000000000000000000', 5000, 2592000, 900, 0)


MSGVALUE1 = rarioCurrencyManager.convert("USD", "MATIC", 5000) -> 3333333333333333333
user1.balance() -> 100000000000000000000
treasury.balance() -> 100000000000000000000
user2.balance() -> 100000000000000000000
rarioMarketplace.balance() -> 0
Calling -> rarioMarketplace.placeBid(78541660797044910968829902406342334108369226379826116161446442989268089806461, user1.address, "MATIC", 0, {'from': user1, 'value': MSGVALUE1 + 1})
Transaction sent: 0xdce02b97e0cc4d09d1c0470a6cba3e1803f6f4eab61ca6e5b694ed0c079c2c9e
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 3
  RarioMarketplace.placeBid confirmed (NFTMarketplace: Cannot buy own listing)   Block: 13504348   Gas used: 25555 (0.38%)
```

Test 4: Creating an auction, payment with TOKEN currency

In this test, user1 created an auction sale by calling the createAuction function. Initially, the price set was 3000. But then user1 called updateListing and updated the price to 5000 USD. Then multiple users placed a bid using TOKEN currency. 2 different issues were found which are explained in:

- INCORRECT LOGIC LEADS TO DOS IN AUCTION SALES
- OUTBID USER DOES NOT RECEIVE BACK HIS FUNDS

4.7 RARIOMARKETPLACECONFIG CONTRACT

The contract RarioMarketplaceConfig inherits from MarketplaceConfig and is used to configure different parameters used by the RarioMarketplace contract.

The contract RarioMarketplaceConfig contains the following functions:

- From MarketplaceConfig
  - _existsUserWallet(address) (internal)
  - _setAllowExternalWallets(bool) (private)
  - _setAuctionExtensionDuration(uint256) (private)
  - _setAuctionPercentIncrementInBasisPoints(uint256) (private)
  - _setGasLimits(uint256,uint256,uint256) (private)
  - _setListingExpiry(uint256) (private)
  - _setPlatformFees(uint256) (private)
  - _setTokenContract(address) (private)
  - _setTreasury(address) (private)
  - _setUserDB(address) (private)
  - getAllowExternalWallets() (external)
  - getAuctionExtensionDuration() (external)
  - getAuctionPercentIncrementInBasisPoints() (external)
  - getGasLimits() (external)
  - getListingExpiry() (external)
  - getPlatformFees() (external)
  - getTokenAddress() (external)
  - getTreasuryAddress() (external)
  - getUserDBAddress() (external)
  - initializeMarketplaceConfig(address,address,address) (internal)
  - isValidWallet(address) (external)
  - setAllowExternalWallets(bool) (external)
  - setAuctionExtensionDuration(uint256) (external)
  - setAuctionPercentIncrementInBasisPoints(uint256) (external)
  - setGasLimits(uint256,uint256,uint256) (external)
  - setListingExpiry(uint256) (external)

- setPlatformFees(uint256) (external)
- setTokenAddress(address) (external)
- setTreasuryAddress(address) (external)
- setUserDBAddress(address) (external)

- From RarioMarketplaceConfig
  - initialize(address,address,address,address) (external)

Test: setAllowExternalWallets - True, setPlatformFees - 5000 (50%)

In this test, we have set the following configuration in the MarketplaceConfig contract:

- marketplaceConfig.setAllowExternalWallets(True, {'from': owner})
- marketplaceConfig.setPlatformFees(5000, {'from': owner})

Below we can see, how the platform fee is now 50% as the 50% of the payment goes to the treasury address. On the other hand, we can also see how external wallets are allowed to bid without having added them previously in the RarioUsers contract:

```
Calling -> marketplaceConfig.setAllowExternalWallets(True, {'from': owner})
Transaction sent: 0xd4f20dff3de876b233b367333802965f9d7c19a1d4d715a996led6e21f45ac45
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 27
  RarioMarketplaceConfig.setAllowExternalWallets confirmed   Block: 13504733   Gas used: 31845 (0.47%)

Calling -> marketplaceConfig.setPlatformFees(5000, {'from': owner})
Transaction sent: 0x82d0629691e7a9eccce93283ca1f6b639bfdad100201e15275ecb262c9168ecb
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 28
  RarioMarketplaceConfig.setPlatformFees confirmed   Block: 13504734   Gas used: 31875 (0.47%)

Calling -> rarioMarketplace.reloadConfiguration({'from': owner})
Transaction sent: 0x3be0feb8f35b734f2f0e843b81e1c175356e99c756ab02ef074df13a178e491c
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 29
  RarioMarketplace.reloadConfiguration confirmed   Block: 13504735   Gas used: 157029 (2.34%)

Calling -> rarioMarketplace.createPublicSale(1, 1337, "USD", user1.address, {'from': user1})
Transaction sent: 0x34625c88532d210bc5a5ab70564e94eefecc5120e23b351779ad7bc9c9ddc661
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 1
  RarioMarketplace.createPublicSale confirmed   Block: 13504736   Gas used: 318241 (4.73%)


msg.value = rarioCurrencyManager.convert("USD", "MATIC", 3000) -> 2000000000000000000
50% for user1 = 1000000000000000000.000000
50% for treasury = 1000000000000000000.000000

token.ownerOf(1) -> 0x1798Bb594bf8963FAf3d2b40663506a402Ee79E9
rarioMarketplace.address -> 0x1798Bb594bf8963FAf3d2b40663506a402Ee79E9

Calling -> rarioMarketplace.placeBid(78541660797044911096882990240634233411083692263798261161614464429892680898064 61, user2.address, "MATIC", 0, {'from': user2, 'value': msgvalue})
Transaction sent: 0xcf145bb4c057eb77f8ca9f47ba41066f5b4f064df9fa8fa1fc54f24b001841c3
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 0
  RarioMarketplace.placeBid confirmed   Block: 13504737   Gas used: 330651 (4.92%)

user1.balance() -> 11000000000000000000
treasury.balance() -> 11000000000000000000
user2.balance() -> 8000000000000000000
rarioMarketplace.balance() -> 0
token.ownerOf(1) -> 0x4D7dFE6B6C32C29968d4953BbDA011b2c1B54B05
user2.address -> 0x4D7dFE6B6C32C29968d4953BbDA011b2c1B54B05
```

4.8 RARIOPOLYGONTOKEN CONTRACT

The contract RarioPolygonToken inherits from RarioPolygonNFT. RarioPolygonNFT at the same time inherits from RarioNFT contract.

The contract RarioPolygonToken contains the following functions:

- From RarioPolygonNFT
  - deposit(address,bytes) (external)
  - encodeTokenMetadata(uint256) (external)
  - initializeRarioPolygonNFT(string,string,address,address) (internal)
  - withdraw(uint256) (external)
  - withdrawBatch(uint256[]) (external)
  - withdrawWithMetadata(uint256) (external)
- From RarioNFT
  - existsAttribute(string) (external)
  - getAttribute(string) (external)
  - getAttributeBooleanValue(uint256,string) (external)
  - getAttributeByIndex(uint256) (external)
  - getAttributeCount() (external)
  - getAttributeNumberValue(uint256,string) (external)
  - getAttributeTextValue(uint256,string) (external)
  - initializeRarioNFT(string,string,address) (internal)
  - isApprovedForAll(address,address) (public)
  - isContractApprovedForAll(address) (external)
  - mint(address,uint256) (external)
  - safeMint(address,uint256) (external)
  - safeMint(address,uint256,bytes) (external)
  - setContractApprovalForAll(address,bool) (external)
- From RarioPolygonToken
  - initialize(address,address,address,string,string) (external)

The contract RarioPolygonNFT correctly follows the Polygon Mintable Assets documentation:

1. The contract RarioPolygonNFT has the same structure as ChildMintableERC721.sol
2. The contract RarioPolygonNFT contains the functions deposit, withdraw and mint. The mint function is inherited, though, from RarioNFT contract and can only be called by an OPERATOR.

Listing 19: RarioNFT.sol (Lines 40)

```
39     function mint(address to, uint256 tokenId) external virtual {
40         _checkRole(OPERATOR, msgSender());
41         _mint(to, tokenId);
42     }
```

It is worth mentioning, as described in the MINT FUNCTION IN RARIOPOLYGONNFT DOES NOT VERIFY IF TOKEN WAS WITHDRAWN findings page that the function mint in RarioPolygonNFT does not verify if the token was withdrawn previously as it is done in ChildMintableERC721.sol

4.9 RARIOETHEREUMTOKEN CONTRACT

The contract RarioEthereumToken inherits from RarioEthereumNFT. RarioEthereumNFT at the same time inherits from RarioNFT contract.

The contract RarioEthereumToken contains the following functions:

- From RarioEthereumNFT
  - exists(uint256) (external)
  - initializeRarioEthereumNFT(string,string,address,address) (internal)
  - mint(address,uint256) (external)
  - mint(address,uint256,bytes) (external)
  - setTokenMetadata(uint256,bytes) (internal)
  - version() (external)
- From RarioNFT
  - existsAttribute(string) (external)
  - getAttribute(string) (external)
  - getAttributeBooleanValue(uint256,string) (external)
  - getAttributeByIndex(uint256) (external)
  - getAttributeCount() (external)
  - getAttributeNumberValue(uint256,string) (external)
  - getAttributeTextValue(uint256,string) (external)
  - initializeRarioNFT(string,string,address) (internal)
  - isApprovedForAll(address,address) (public)
  - isContractApprovedForAll(address) (external)
  - safeMint(address,uint256) (external)
  - safeMint(address,uint256,bytes) (external)
  - setContractApprovalForAll(address,bool) (external)
- From RarioEthereumToken
  - initialize(address,address,address,string,string) (external)

The contract RarioEthereumNFT correctly follows the Polygon Mintable Assets documentation:

1. The contract RarioEthereumNFT has the same structure as DummyMintableERC721.sol

4.10 RARIOUSERS CONTRACT

The contract RarioUsers inherits from UserDB. This contract is used like a database to keep control of all the users and wallets.

The contract RarioUsers contains the following functions which are all only callable by the OPERATOR role:

- From UserDB
  - addUser(uint256,address) (external)
  - existsUserId(uint256) (external)
  - existsUserWallet(address) (external)
  - getUserByAddress(address) (external)
  - getUserById(uint256) (external)
  - getUserByIndex(uint256) (external)
  - getUserCount() (external)
  - updateUser(uint256,address) (external)
- From RarioUsers
  - initialize(address) (external)

```
>>> rariousers.addUser(1, user1.address, {'from': owner})
Transaction sent: 0xf7773e83162c90aa8cd0304a45ff88b185975d8b693c92134e16e18a9c5f1af9
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 27
  RarioUsers.addUser confirmed   Block: 13511177   Gas used: 132294 (1.97%)

<Transaction '0xf7773e83162c90aa8cd0304a45ff88b185975d8b693c92134e16e18a9c5f1af9'>
>>> rariousers.addUser(2, user2.address, {'from': owner})
Transaction sent: 0xdb7081182905b407737c2bd8f98924e8adbca47160d22f7fbe38cf50af02079e
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 28
  RarioUsers.addUser confirmed   Block: 13511178   Gas used: 117294 (1.74%)

<Transaction '0xdb7081182905b407737c2bd8f98924e8adbca47160d22f7fbe38cf50af02079e'>
>>> rariousers.existsUserId(0)
False
>>> rariousers.existsUserId(1)
True
>>> rariousers.existsUserId(2)
True
>>> rariousers.existsUserWallet(user1.address)
True
>>> rariousers.existsUserWallet(user2.address)
True
>>> rariousers.existsUserWallet(user3.address)
False
>>> rariousers.getUserCount()
2
>>> rariousers.getUserByIndex(1)
(2, "0x3BDE98b2F0f540b3Bda46ca679bb288E01D0D61B")
>>> rariousers.getUserByIndex(0)
(1, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserById(1)
(1, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserById(2)
(2, "0x3BDE98b2F0f540b3Bda46ca679bb288E01D0D61B")
>>> rariousers.updateUser(2, user1.address)
Transaction sent: 0xb1e17a4a9c3e28486698a4a5fb0b1abf3ca1d1bb4c45bb1b5d13df1216b5053f
  Gas price: 0.0 gwei   Gas limit: 6721975   Nonce: 29
  RarioUsers.updateUser confirmed   Block: 13511179   Gas used: 29009 (0.43%)

<Transaction '0xb1e17a4a9c3e28486698a4a5fb0b1abf3ca1d1bb4c45bb1b5d13df1216b5053f'>
>>> rariousers.getUserById(1)
(1, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserById(2)
(2, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
>>> rariousers.getUserByAddress(user1.address)
(1, "0x1B1c0E70CBFc1d01b25c667a48bF018b7E5196C3")
```

All the functions are working as expected as can be seen above except updateUser which is not validating if the address of the wallet was

already added, as described in the findings page UPDATEUSER FUNCTION MISSING REQUIRE STATEMENT.

AUTOMATED TESTING

5.1 STATIC ANALYSIS REPORT

Description:

Halborn used automated testing techniques to enhance the coverage of certain areas of the scoped contracts. Among the tools used was Slither, a Solidity static analysis framework. After Halborn verified all the contracts in the repository and was able to compile them correctly into their abi and binary formats, Slither was run on the all-scoped contracts. This tool can statically verify mathematical relationships between Solidity variables to detect invalid or inconsistent usage of the contracts' APIs across the entire code-base.

Slither results:

RarioCurrencyManager.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

RarioCurrencyManager (contracts/RarioCurrencyManager.sol#13-24) is an upgradeable contract that does not protect its initiliaze functions: RarioCurrencyManager.initialize(address,string) (contracts/RarioCurrencyManager.sol#20-23). Anyone can delete the contract with: ERC1822UUPS.upgradeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

CurrencyManager._convert(string,string,uint256).price (contracts/traits/CurrencyManager.sol#347) is a local variable never initialized
CurrencyManager._convert(string,string,uint256).currencyPairDecimals (contracts/traits/CurrencyManager.sol#346) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
CurrencyManager.addCurrencyPair(string,string,string,bool,uint8,address) (contracts/traits/CurrencyManager.sol#137-172) ignores return value by _currencyPairSymbols.add(pairSymbol) (contracts/traits/CurrencyManager.sol#161)
CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) ignores return value by IPriceFeed(feedContractAddress).decimals() (contracts/traits/CurrencyManager.sol#344-363)
CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) ignores return value by IPriceFeed(feedContractAddress).latestRoundData() (contracts/traits/CurrencyManager.sol#345-360)
CurrencyManager._addCurrency(CurrencyManager.CurrencyType,string,address,uint8) (contracts/traits/CurrencyManager.sol#366-383) ignores return value by _symbols.add(symbol) (contracts/traits/CurrencyManager.sol#380)
CurrencyManager._removeCurrency(string) (contracts/traits/CurrencyManager.sol#385-395) ignores return value by _symbols.remove(symbol) (contracts/traits/CurrencyManager.sol#392)
CurrencyManager._removeCurrencyPair(string) (contracts/traits/CurrencyManager.sol#397-418) ignores return value by _currencyPairSymbols.remove(symbol) (contracts/traits/CurrencyManager.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'CurrencyManager._convert(string,string,uint256).currencyPairDecimals (contracts/traits/CurrencyManager.sol#346)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used before declaration: ((amount * (10 ** (toCurrencyDecimals + currencyPairDecimals - fromCurrencyDecimals))) / uint256(price)) (contracts/traits/CurrencyManager.sol#352-357)
Variable 'CurrencyManager._convert(string,string,uint256).price (contracts/traits/CurrencyManager.sol#347)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used before declaration: ((amount * (10 ** (toCurrencyDecimals + currencyPairDecimals - fromCurrencyDecimals))) / uint256(price)) (contracts/traits/CurrencyManager.sol#352-357)
Variable 'CurrencyManagre._convert(string,string,uint256).currencyPairDecimals (contracts/traits/CurrencyManager.sol#346)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used before declaration: (uint256(price) * amount) / (10 ** (fromCurrencyDecimals + currencyPairDecimals - toCurrencyDecimals)) (contracts/traits/CurrencyManager.sol#352-357)
Variable 'CurrencyManager._convert(string,string,uint256).price (contracts/traits/CurrencyManager.sol#347)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used before declaration: (uint256(price) * amount) / (10 ** (fromCurrencyDecimals + currencyPairDecimals - toCurrencyDecimals)) (contracts/traits/CurrencyManager.sol#352-357)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

RarioData.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

RarioData (contracts/RarioData.sol#13-23) is an upgradeable contract that does not protect its initiliaze functions: RarioData.initialize(address) (contracts/RarioData.sol#20-22). Anyone can delete the contract with: ERC1822UUPS.upgradeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
MetadataDB.addAttribute(string,string,MetadataDB.AttributeType,bool) (contracts/traits/MetadataDB.sol#172-193) ignores return value by _attributeNames.add(attribute) (contracts/traits/MetadataDB.sol#191)
MetadataDB.setTextValue(uint256,string,string) (contracts/traits/MetadataDB.sol#200-221) ignores return value by _tokenIds.add(tokenId) (contracts/traits/MetadataDB.sol#217)
MetadataDB.setNumberValue(uint256,string,int256) (contracts/traits/MetadataDB.sol#223-240) ignores return value by _tokenIds.add(tokenId) (contracts/traits/MetadataDB.sol#236)
MetadataDB.setBooleanValue(uint256,string,bool) (contracts/traits/MetadataDB.sol#242-259) ignores return value by _tokenIds.add(tokenId) (contracts/traits/MetadataDB.sol#255)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
EnumerableStringSet.remove(EnumerableStringSet.StringSet,string) (contracts/lib/EnumerableStringSet.sol#57-89) is never used and should be removed
EnumerableStringSet.values(EnumerableStringSet.StringSet) (contracts/lib/EnumerableStringSet.sol#127-129) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

RarioEthereumToken.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

RarioEthereumToken (contracts/RarioEthereumToken.sol#13-25) is an upgradeable contract that does not protect its initiliaze functions: RarioEthereumToken.initialize(address,address,address,string,string) (contracts/RarioEthereumToken.sol
#14-23). Anyone can delete the contract with: ERC1822UUPS.upgradeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic
/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) ignores return value by IERC721Receiver(to).onERC721Received(msgSender(),from,tokenId,_data) (contracts/traits/ERC721NFT.sol#445-455
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

RarioNFT.getAttributeByIndex(uint256).name (contracts/traits/RarioNFT.sol#66) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
RarioNFT.getAttribute(string).name (contracts/traits/RarioNFT.sol#80) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).retval (contracts/traits/ERC721NFT.sol#445)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: retval == IERC721Receiver.onERC721Received.selector (contracts/traits/ERC721NFT.sol#446)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: reason.length == 0 (contracts/traits/ERC721NFT.sol#448)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (contracts/traits/ERC721NFT.sol#452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3


Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) uses assembly
        - INLINE ASM (contracts/traits/ERC721NFT.sol#451-453)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#78-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Variable Context._trustedForwarder (contracts/traits/Context.sol#6) is too similar to RarioEthereumToken.initialize(address,address,address,string,string).trustedForwarder_ (contracts/RarioEthereumToken.sol#15)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

ERC721NFT.E_NFT_TRANSFER_TO_NON_RECEIVER (contracts/traits/ERC721NFT.sol#30) is never used in RarioEthereumToken (contracts/RarioEthereumToken.sol#13-25)
RarioEthereumNFT._gap (contracts/traits/RarioEthereumNFT.sol#75) is never used in RarioEthereumToken (contracts/RarioEthereumToken.sol#13-25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

RarioMarketplace.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Reentrancy in Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615):
        External calls:
        - _sendValue(_treasury,platformPayableAmount,_gasLimitMedium) (contracts/traits/Marketplace.sol#607)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        - _sendValue(seller,sellerPayableAmount,_gasLimitMedium) (contracts/traits/Marketplace.sol#608)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        State variables written after the call(s):
        - _sendValue(seller,sellerPayableAmount,_gasLimitMedium) (contracts/traits/Marketplace.sol#608)
                - _pendingWithdrawals[user] = _pendingWithdrawals[user] + amount (contracts/traits/Marketplace.sol#632)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377) ignores return value by IERC20(currencyContractAddress).transferFrom(msgSender(),address(this),bidValue) (contracts/traits/Marketplace.sol#37
9)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) ignores return value by IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#6
11)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) ignores return value by IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

RarioMarketplace (contracts/RarioMarketplace.sol#13-24) is an upgradeable contract that does not protect its initiliaze functions: RarioMarketplace.initialize(address,address,address) (contracts/RarioMarketplace.sol#17-23). Anyone can de
lete the contract with: ERC1822UUPS.upgradeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic/slither/wiki/Detector
-Documentation#unprotected-upgradeable-contract

Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563) uses a dangerous strict equality:
        - require(bool,string)(listing.sellerTokenWallet == msgSender(),ERR_LISTING_WRONG_OWNER) (contracts/traits/Marketplace.sol#553)
Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563) uses a dangerous strict equality:
        - require(bool,string)(listing.endTime == 0,ERR_ADCTION_IN_PROGRESS) (contracts/traits/Marketplace.sol#554)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) uses a dangerous strict equality:
        - currencyType == 1 (contracts/traits/Marketplace.sol#605)
Marketplace._sendValue(address,uint256,uint256) (contracts/traits/Marketplace.sol#617-634) uses a dangerous strict equality:
        - amount == 0 (contracts/traits/Marketplace.sol#622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Marketplace._createListing(uint256,uint256,string,address,address,address,uint8) (contracts/traits/Marketplace.sol#442-498):
        External calls:
        - require(bool,string)(_isValidWallet(msgSender()),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#459)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _transferToken(msgSender(),address(this),tokenId) (contracts/traits/Marketplace.sol#460)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        State variables written after the call(s):
        - _listedTokens[tokenId] = listingId (contracts/traits/Marketplace.sol#481)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
```

```
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        State variables written after the call(s):
        - listing.endTime = block.timestamp (contracts/traits/Marketplace.sol#324)
        - listing.buyerTokenWallet = address(buyerTokenWallet) (contracts/traits/Marketplace.sol#325)
        - listing.buyerPaymentWallet = address(msgSender()) (contracts/traits/Marketplace.sol#326)
        - listing.bidCurrency = currency (contracts/traits/Marketplace.sol#327)
        - listing.bidAmount = bidValue (contracts/traits/Marketplace.sol#328)
        - listing.price = bidValueInListingCurrency (contracts/traits/Marketplace.sol#337)
        - listing.buyerPaymentWallet = address(msgSender()) (contracts/traits/Marketplace.sol#338)
        - listing.buyerTokenWallet = address(buyerTokenWallet) (contracts/traits/Marketplace.sol#339)
        - listing.bidCurrency = currency (contracts/traits/Marketplace.sol#340)
        - listing.bidAmount = bidValue (contracts/traits/Marketplace.sol#341)
        - listing.endTime = block.timestamp + listing.duration (contracts/traits/Marketplace.sol#343)
        - listing.price = bidValueInListingCurrency (contracts/traits/Marketplace.sol#358)
        - listing.bidCurrency = currency (contracts/traits/Marketplace.sol#359)
        - listing.bidAmount = bidValue (contracts/traits/Marketplace.sol#360)
        - listing.buyerPaymentWallet = address(msgSender()) (contracts/traits/Marketplace.sol#361)
        - listing.buyerTokenWallet = address(buyerTokenWallet) (contracts/traits/Marketplace.sol#362)
        - listing.endTime = block.timestamp + listing.extensionDuration (contracts/traits/Marketplace.sol#366)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Marketplace._isValidWallet(address).isValid (contracts/traits/Marketplace.sol#501) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
Marketplace._isValidWallet(address) (contracts/traits/Marketplace.sol#500-506) ignores return value by _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'Marketplace._isValidWallet(address).isValid (contracts/traits/Marketplace.sol#501)' in Marketplace._isValidWallet(address) (contracts/traits/Marketplace.sol#500-506) potentially used before declaration: isValid (contracts/trait
s/Marketplace.sol#502)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in Marketplace._createListing(uint256,uint256,string,address,address,address,uint8) (contracts/traits/Marketplace.sol#442-498):
        External calls:
        - require(bool,string)(_isValidWallet(msgSender()),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#459)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _transferToken(msgSender(),address(this),tokenId) (contracts/traits/Marketplace.sol#460)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        State variables written after the call(s):
        - _listings[listingId] = listing (contracts/traits/Marketplace.sol#482)
Reentrancy in Marketplace._reloadConfiguration() (contracts/traits/Marketplace.sol#423-440):
        External calls:
        - (_gasLimitHigh,_gasLimitMedium,_gasLimitLow) = _config.getGasLimits() (contracts/traits/Marketplace.sol#428)
        State variables written after the call(s):
        - _allowExternalWallets = _config.getAllowExternalWallets() (contracts/traits/Marketplace.sol#435)
        - _auctionExtensionDuration = _config.getAuctionExtensionDuration() (contracts/traits/Marketplace.sol#432)
        - _auctionPercentIncrementInBasisPoints = _config.getAuctionPercentIncrementInBasisPoints() (contracts/traits/Marketplace.sol#433)
        - _listingExpiry = _config.getListingExpiry() (contracts/traits/Marketplace.sol#431)
        - _nativeCurrency = _currencyManager.getNativeCurrency() (contracts/traits/Marketplace.sol#437)
        - _platformFeesBasisPoints = _config.getPlatformFees() (contracts/traits/Marketplace.sol#430)
        - delete _supportedCurrencies (contracts/traits/Marketplace.sol#438)
        - _supportedCurrencies = _currencyManager.supportedCurrencies() (contracts/traits/Marketplace.sol#439)
Reentrancy in Marketplace._sendValue(address,uint256,uint256) (contracts/traits/Marketplace.sol#617-634):
        External calls:
        - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        State variables written after the call(s):
        - _pendingWithdrawals[user] = _pendingWithdrawals[user] + amount (contracts/traits/Marketplace.sol#632)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
                - IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#611)
                - IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
        External calls sending eth:
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        State variables written after the call(s):
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - _finalisedListings[listingId] = listing (contracts/traits/Marketplace.sol#572)
Reentrancy in Marketplace.reloadConfiguration() (contracts/traits/Marketplace.sol#154-158):
        External calls:
        - _reloadConfiguration() (contracts/traits/Marketplace.sol#156)
                - (_gasLimitHigh,_gasLimitMedium,_gasLimitLow) = _config.getGasLimits() (contracts/traits/Marketplace.sol#428)
        State variables written after the call(s):
        - _hasConfiguration = true (contracts/traits/Marketplace.sol#157)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563):
        External calls:
        - IERC721(_trustedToken).transferFrom(address(this),listing.sellerTokenWallet,tokenId) (contracts/traits/Marketplace.sol#560)
        Event emitted after the call(s):
        - ListingCanceled(uint8(listing.listingType),tokenId,listingId) (contracts/traits/Marketplace.sol#562)
Reentrancy in Marketplace._createListing(uint256,uint256,string,address,address,address,uint8) (contracts/traits/Marketplace.sol#442-498):
        External calls:
        - require(bool,string)(_isValidWallet(msgSender()),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#459)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _transferToken(msgSender(),address(this),tokenId) (contracts/traits/Marketplace.sol#460)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        Event emitted after the call(s):
        - ListingCreated(listingType,listingId,tokenId,msgSender(),sellerPaymentWallet,buyerTokenWallet,buyerPaymentWallet,listing.price,listing.duration,listing.extensionDuration,listing.endTime) (contracts/traits/Marketplace.sol#485-49
7)
Reentrancy in Marketplace._finaliseListing(uint256) (contracts/traits/Marketplace.sol#565-594):
        External calls:
        - _transferToken(address(this),listing.buyerTokenWallet,listing.tokenId) (contracts/traits/Marketplace.sol#576)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        - (platformFee,sellerPayableAmount) = _distributeFunds(listing.sellerPaymentWallet,listing.bidCurrency,listing.bidAmount) (contracts/traits/Marketplace.sol#577-581)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
                - IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#611)
                - IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
        External calls sending eth:
        - (platformFee,sellerPayableAmount) = _distributeFunds(listing.sellerPaymentWallet,listing.bidCurrency,listing.bidAmount) (contracts/traits/Marketplace.sol#577-581)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
                - IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#611)
                - IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
        External calls sending eth:
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        Event emitted after the call(s):
        - ListingFinalized(uint8(listing.listingType),listingId,listing.tokenId,listing.sellerTokenWallet,listing.sellerPaymentWallet,listing.buyerTokenWallet,listing.buyerPaymentWallet,platformFee,sellerPayableAmount) (contracts/traits/
Marketplace.sol#583-593)
                - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        Event emitted after the call(s):
        - AuctionBidPlaced(listing.listingId,msgSender(),bidValue,listing.endTime) (contracts/traits/Marketplace.sol#344)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - IERC20(currencyContractAddress).transferFrom(msgSender(),address(this),bidValue) (contracts/traits/Marketplace.sol#370)
        Event emitted after the call(s):
        - AuctionBidPlaced(listing.listingId,msgSender(),bidValue,listing.endTime) (contracts/traits/Marketplace.sol#373)
Reentrancy in Marketplace.withdrawFor(address) (contracts/traits/Marketplace.sol#401-407):
        External calls:
        - _sendValue(user,amount,_gasLimitHigh) (contracts/traits/Marketplace.sol#405)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        Event emitted after the call(s):
        - Withdrawal(user,amount) (contracts/traits/Marketplace.sol#406)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(listing.price > 0,ERR_LISTING_NOT_EXISTS) (contracts/traits/Marketplace.sol#304)
        - require(bool,string)(listing.sellerTokenWallet != buyerTokenWallet,ERR_CANNOT_BUY_OWN_LISTING) (contracts/traits/Marketplace.sol#305)
        - require(bool,string)(listing.buyerTokenWallet == buyerTokenWallet,ERR_PRIVATE_SALE_ONLY) (contracts/traits/Marketplace.sol#320)
        - require(bool,string)(listing.price <= bidValueInListingCurrency,ERR_BID_PRICE_LOW) (contracts/traits/Marketplace.sol#323)
        - require(bool,string)(listing.price <= bidValueInListingCurrency,NFTMarketReserveAuction: Bid must be at least the reserve price) (contracts/traits/Marketplace.sol#333-336)
        - require(bool,string)(listing.endTime >= block.timestamp,NFTMarketReserveAuction: Auction is over) (contracts/traits/Marketplace.sol#347)
        - require(bool,string)(listing.buyerPaymentWallet != msgSender(),NFTMarketReserveAuction: You already have an outstanding bid) (contracts/traits/Marketplace.sol#348-351)
        - listing.endTime - block.timestamp < listing.extensionDuration (contracts/traits/Marketplace.sol#365)
Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(listing.listingId != uint256(0),ERR_LISTING_NOT_EXISTS) (contracts/traits/Marketplace.sol#552)
        - require(bool,string)(listing.sellerTokenWallet == msgSender(),ERR_LISTING_WRONG_OWNER) (contracts/traits/Marketplace.sol#553)
        - require(bool,string)(listing.endTime == 0,ERR_AUCTION_IN_PROGRESS) (contracts/traits/Marketplace.sol#554)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) uses timestamp for comparisons
        Dangerous comparisons:
        - currencyType == 1 (contracts/traits/Marketplace.sol#605)
Marketplace._sendValue(address,uint256,uint256) (contracts/traits/Marketplace.sol#617-634) uses timestamp for comparisons
        Dangerous comparisons:
        - amount == 0 (contracts/traits/Marketplace.sol#622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-55) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Counters.decrement(Counters.Counter) (contracts/lib/Counters.sol#31-37) is never used and should be removed
Counters.reset(Counters.Counter) (contracts/lib/Counters.sol#39-41) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
ReentrancyGuard.initializeReentrancyGuard() (contracts/traits/ReentrancyGuard.sol#39-41) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-24) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

ReentrancyGuard.__gap (contracts/traits/ReentrancyGuard.sol#63) is never used in RarioMarketplace (contracts/RarioMarketplace.sol#13-24)
Marketplace.ERR_UNSUPPORTED_CURRENCY (contracts/traits/Marketplace.sol#56) is never used in RarioMarketplace (contracts/RarioMarketplace.sol#13-24)
Marketplace.ERR_CURRENCY_MANAGER (contracts/traits/Marketplace.sol#57) is never used in RarioMarketplace (contracts/RarioMarketplace.sol#13-24)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

getPendingWithdrawal(address) should be declared external:
        - Marketplace.getPendingWithdrawal(address) (contracts/traits/Marketplace.sol#387-389)
withdraw() should be declared external:
        - Marketplace.withdraw() (contracts/traits/Marketplace.sol#394-396)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

RarioMarketplaceConfig.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

RarioMarketplaceConfig (contracts/RarioMarketplaceConfig.sol#13-23) is an upgradeable contract that does not protect its initiliaze functions: RarioMarketplaceConfig.initialize(address,address,address,address) (contracts/RarioMarketplaceConfig.sol#14-22). Anyone can delete the contract with: ERC1822UUPS.upgradeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

MarketplaceConfig._existsUserWallet(address).isValid (contracts/traits/MarketplaceConfig.sol#145) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
MarketplaceConfig._existsUserWallet(address) (contracts/traits/MarketplaceConfig.sol#144-150) ignores return value by IUserDB(_rarioUserDB).existsUserWallet(wallet) (contracts/traits/MarketplaceConfig.sol#145-149)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'MarketplaceConfig._existsUserWallet(address).isValid (contracts/traits/MarketplaceConfig.sol#145)' in MarketplaceConfig._existsUserWallet(address) (contracts/traits/MarketplaceConfig.sol#144-150) potentially used before declaration: retval = isValid (contracts/traits/MarketplaceConfig.sol#146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

RarioPolygonToken.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

RarioPolygonToken (contracts/RarioPolygonToken.sol#13-25) is an upgradeable contract that does not protect its initiliaze functions: RarioPolygonToken.initialize(address,address,address,string,string) (contracts/RarioPolygonToken.sol#14-23). Anyone can delete the contract with: ERC1822UUPS.upgradeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-contract

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) ignores return value by IERC721Receiver(to).onERC721Received(msgSender(),from,tokenId,_data) (contracts/traits/ERC721NFT.sol#445-455)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

RarioNFT.getAttributeByIndex(uint256).name (contracts/traits/RarioNFT.sol#66) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/ERC721Metadata.sol#15) (function)
RarioNFT.getAttribute(string).name (contracts/traits/RarioNFT.sol#80) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/ERC721Metadata.sol#15) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).retval (contracts/traits/ERC721NFT.sol#445)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (contracts/traits/ERC721NFT.sol#446)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentially used before declaration: reason.length == 0 (contracts/traits/ERC721NFT.sol#448)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (contracts/traits/ERC721NFT.sol#452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) uses assembly
        - INLINE ASM (contracts/traits/ERC721NFT.sol#451-453)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Variable Context._trustedForwarder (contracts/traits/Context.sol#6) is too similar to RarioPolygonToken.initialize(address,address,address,string,string).trustedForwarder_ (contracts/RarioPolygonToken.sol#15)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

ERC721NFT.E_NFT_TRANSFER_TO_NON_RECEIVER (contracts/traits/ERC721NFT.sol#30) is never used in RarioPolygonToken (contracts/RarioPolygonToken.sol#13-25)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

RarioUsers.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

RarioUsers (contracts/RarioUsers.sol#13-23) is an upgradeable contract that does not protect its initiliaze functions: RarioUsers.initialize(address) (contracts/RarioUsers.sol#20-22). Anyone can delete the contract with: ERC1822UUPS.upgr
adeTo(address) (contracts/traits/ERC1822UUPS.sol#32-35)ERC1822UUPS.upgradeToAndCall(address,bytes) (contracts/traits/ERC1822UUPS.sol#45-48)Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unprotected-upgradeable-c
ontract

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
UserDB.addUser(uint256,address) (contracts/traits/UserDB.sol#41-51) ignores return value by rarioIds.add(rarioId) (contracts/traits/UserDB.sol#47)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage


Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/AccessControl.sol

```
AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setRoleAdmin(bytes32,bytes32) (contracts/traits/AccessControl.sol#221-225) is never used and should be removed
AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/Context.sol

```
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Context.isTrustedForwarder(address) (contracts/traits/Context.sol#12-14) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.msgSender() (contracts/traits/Context.sol#16-25) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/CurrencyManager.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

CurrencyManager._convert(string,string,uint256).price (contracts/traits/CurrencyManager.sol#347) is a local variable never initialized
CurrencyManager._convert(string,string,uint256).currencyPairDecimals (contracts/traits/CurrencyManager.sol#344) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
CurrencyManager.addCurrencyPair(string,string,string,bool,uint8,address) (contracts/traits/CurrencyManager.sol#157-172) ignores return value by _currencyPairSymbols.add(pairSymbol) (contracts/traits/CurrencyManager.sol#161)
CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) ignores return value by IPriceFeed(feedContractAddress).decimals() (contracts/traits/CurrencyManager.sol#344-363)
CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) ignores return value by IPriceFeed(feedContractAddress).latestRoundData() (contracts/traits/CurrencyManager.sol#345-360)
CurrencyManager._addCurrency(CurrencyManager.CurrencyType,string,address,uint8) (contracts/traits/CurrencyManager.sol#366-383) ignores return value by _symbols.add(symbol) (contracts/traits/CurrencyManager.sol#380)
CurrencyManager._removeCurrency(string) (contracts/traits/CurrencyManager.sol#385-395) ignores return value by _symbols.remove(symbol) (contracts/traits/CurrencyManager.sol#392)
CurrencyManager._removeCurrencyPair(string) (contracts/traits/CurrencyManager.sol#397-418) ignores return value by _currencyPairSymbols.remove(symbol) (contracts/traits/CurrencyManager.sol#408)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'CurrencyManager._convert(string,string,uint256).currencyPairDecimals (contracts/traits/CurrencyManager.sol#344)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used
 before declaration: ((amount * (10 ** (toCurrencyDecimals + currencyPairDecimals - fromCurrencyDecimals))) / uint256(price)) (contracts/traits/CurrencyManager.sol#352-357)
Variable 'CurrencyManager._convert(string,string,uint256).price (contracts/traits/CurrencyManager.sol#347)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used before declara
tion: ((amount * (10 ** (toCurrencyDecimals + currencyPairDecimals - fromCurrencyDecimals))) / uint256(price)) (contracts/traits/CurrencyManager.sol#352-357)
Variable 'CurrencyManager._convert(string,string,uint256).currencyPairDecimals (contracts/traits/CurrencyManager.sol#344)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used
 before declaration: (uint256(price) * amount) / (10 ** (fromCurrencyDecimals + currencyPairDecimals - toCurrencyDecimals)) (contracts/traits/CurrencyManager.sol#352-357)
Variable 'CurrencyManager._convert(string,string,uint256).price (contracts/traits/CurrencyManager.sol#347)' in CurrencyManager._convert(string,string,uint256) (contracts/traits/CurrencyManager.sol#328-364) potentially used before declara
tion: (uint256(price) * amount) / (10 ** (fromCurrencyDecimals + currencyPairDecimals - toCurrencyDecimals)) (contracts/traits/CurrencyManager.sol#352-357)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
            - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
            - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
            - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#18-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
CurrencyManager.initializeCurrencyManager(string) (contracts/traits/CurrencyManager.sol#100-104) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-146) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-176) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Upgradeable.initializeUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/ERC165Interface.sol

```
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/ERC1822UUPS.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
	- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
	External calls:
	- _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
		- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
	- _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
		- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
	Event emitted after the call(s):
	- Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
		- _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
	- INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
	- INLINE ASM (contracts/lib/Address.sol#180-183)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

ERC1822UUPS (contracts/traits/ERC1822UUPS.sol#20-63) does not implement functions:
	- ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
```

traits/ERC1967Upgrade.sol

```
Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
	External calls:
	- _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
		- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
	- _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
		- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
	Event emitted after the call(s):
	- Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
		- _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
	- INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
	- INLINE ASM (contracts/lib/Address.sol#180-183)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.isContract(address) (contracts/lib/Address.sol#26-36) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._getImplementation() (contracts/traits/ERC1967Upgrade.sol#39-41) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._setImplementation(address) (contracts/traits/ERC1967Upgrade.sol#46-49) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeTo(address) (contracts/traits/ERC1967Upgrade.sol#56-59) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) is never used and should be removed
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/ERC721NFT.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
	- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) ignores return value by IERC721Receiver(to).onERC721Received(msgSender(),from,tokenId,_data) (contracts/traits/ERC721NFT.sol#445-455
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).retval (contracts/traits/ERC721NFT.sol#445)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: retval == IERC721Receiver.onERC721Received.selector (contracts/traits/ERC721NFT.sol#446)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: reason.length == 0 (contracts/traits/ERC721NFT.sol#448)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (contracts/traits/ERC721NFT.sol#452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
	External calls:
	- _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
		- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
	- _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
		- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
	Event emitted after the call(s):
	- Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
		- _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
	- INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
	- INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
	- INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
	- INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
	- INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
	- INLINE ASM (contracts/traits/Context.sol#19-21)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) uses assembly
	- INLINE ASM (contracts/traits/ERC721NFT.sol#451-453)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initialiseERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#126-128) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initialiseERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
ERC721NFT._mint(address,uint256) (contracts/traits/ERC721NFT.sol#351-361) is never used and should be removed
ERC721NFT._safeMint(address,uint256) (contracts/traits/ERC721NFT.sol#319-321) is never used and should be removed
ERC721NFT._safeMint(address,uint256,bytes) (contracts/traits/ERC721NFT.sol#327-337) is never used and should be removed
ERC721NFT.initialiseERC721NFT(string,string) (contracts/traits/ERC721NFT.sol#66-77) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initialisePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Upgradeable.initialiseUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

ERC721NFT.E_NFT_TRANSFER_TO_NON_RECEIVER (contracts/traits/ERC721NFT.sol#30) is never used in ERC721NFT (contracts/traits/ERC721NFT.sol#19-567)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

traits/Initializable.sol

No results from Slither.

traits/Marketplace.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
	- (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

Reentrancy in Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615):
	External calls:
	- _sendValue(_treasury,platformPayableAmount,_gasLimitMedium) (contracts/traits/Marketplace.sol#607)
		- (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
	- _sendValue(seller,sellerPayableAmount,_gasLimitMedium) (contracts/traits/Marketplace.sol#608)
		- (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
	State variables written after the call(s):
	- _sendValue(seller,sellerPayableAmount,_gasLimitMedium) (contracts/traits/Marketplace.sol#608)
		- _pendingWithdrawals[user] = _pendingWithdrawals[user] + amount (contracts/traits/Marketplace.sol#632)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities

Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377) ignores return value by IERC20(currencyContractAddress).transferFrom(msgSender(),address(this),bidValue) (contracts/traits/Marketplace.sol#37
0)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) ignores return value by IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#6
11)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) ignores return value by IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unchecked-transfer

Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563) uses a dangerous strict equality:
	- require(bool,string)(listing.sellerTokenWallet == msgSender(),ERR_LISTING_WRONG_OWNER) (contracts/traits/Marketplace.sol#553)
Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563) uses a dangerous strict equality:
	- require(bool,string)(listing.endTime == 0,ERR_AUCTION_IN_PROGRESS) (contracts/traits/Marketplace.sol#554)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) uses a dangerous strict equality:
	- currencyType == 1 (contracts/traits/Marketplace.sol#605)
Marketplace._sendValue(address,uint256,uint256) (contracts/traits/Marketplace.sol#617-634) uses a dangerous strict equality:
	- amount == 0 (contracts/traits/Marketplace.sol#622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dangerous-strict-equalities

Reentrancy in Marketplace._createListing(uint256,uint256,string,address,address,address,uint8) (contracts/traits/Marketplace.sol#442-498):
	External calls:
	- require(bool,string)(_isValidWallet(msgSender()),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#459)
		- _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
	- _transferToken(msgSender(),address(this),tokenId) (contracts/traits/Marketplace.sol#460)
		- IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
	State variables written after the call(s):
	- _listedTokens[tokenId] = listingId (contracts/traits/Marketplace.sol#481)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
	External calls:
	- require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
		- _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
```

```
        State variables written after the call(s):
        - listing.endTime = block.timestamp (contracts/traits/Marketplace.sol#324)
        - listing.buyerTokenWallet = address(buyerTokenWallet) (contracts/traits/Marketplace.sol#325)
        - listing.buyerPaymentWallet = address(msgSender()) (contracts/traits/Marketplace.sol#326)
        - listing.bidCurrency = currency (contracts/traits/Marketplace.sol#327)
        - listing.bidAmount = bidValue (contracts/traits/Marketplace.sol#328)
        - listing.price = bidValueInListingCurrency (contracts/traits/Marketplace.sol#337)
        - listing.buyerPaymentWallet = address(msgSender()) (contracts/traits/Marketplace.sol#338)
        - listing.buyerTokenWallet = address(buyerTokenWallet) (contracts/traits/Marketplace.sol#339)
        - listing.bidCurrency = currency (contracts/traits/Marketplace.sol#340)
        - listing.bidAmount = bidValue (contracts/traits/Marketplace.sol#341)
        - listing.endTime = block.timestamp + listing.duration (contracts/traits/Marketplace.sol#343)
        - listing.price = bidValueInListingCurrency (contracts/traits/Marketplace.sol#358)
        - listing.bidCurrency = currency (contracts/traits/Marketplace.sol#359)
        - listing.bidAmount = bidValue (contracts/traits/Marketplace.sol#360)
        - listing.buyerPaymentWallet = address(msgSender()) (contracts/traits/Marketplace.sol#361)
        - listing.buyerTokenWallet = address(buyerTokenWallet) (contracts/traits/Marketplace.sol#362)
        - listing.endTime = block.timestamp + listing.extensionDuration (contracts/traits/Marketplace.sol#366)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-1

Marketplace._isValidWallet(address).isValid (contracts/traits/Marketplace.sol#501) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
Marketplace._isValidWallet(address) (contracts/traits/Marketplace.sol#500-506) ignores return value by _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'Marketplace._isValidWallet(address).isValid (contracts/traits/Marketplace.sol#501)' in Marketplace._isValidWallet(address) (contracts/traits/Marketplace.sol#500-506) potentially used before declaration: isValid (contracts/trait
s/Marketplace.sol#502)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables


Reentrancy in Marketplace._createListing(uint256,uint256,string,address,address,address,uint8) (contracts/traits/Marketplace.sol#442-498):
        External calls:
        - require(bool,string)(_isValidWallet(msgSender()),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#459)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _transferToken(msgSender(),address(this),tokenId) (contracts/traits/Marketplace.sol#460)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        State variables written after the call(s):
        - _listings[listingId] = listing (contracts/traits/Marketplace.sol#482)
Reentrancy in Marketplace._reloadConfiguration() (contracts/traits/Marketplace.sol#423-440):
        External calls:
        - (_gasLimitHigh,_gasLimitMedium,_gasLimitLow) = _config.getGasLimits() (contracts/traits/Marketplace.sol#428)
        State variables written after the call(s):
        - _allowExternalWallets = _config.getAllowExternalWallets() (contracts/traits/Marketplace.sol#435)
        - _auctionExtensionDuration = _config.getAuctionExtensionDuration() (contracts/traits/Marketplace.sol#432)
        - _auctionPercentIncrementInBasisPoints = _config.getAuctionPercentIncrementInBasisPoints() (contracts/traits/Marketplace.sol#433)
        - _listingExpiry = _config.getListingExpiry() (contracts/traits/Marketplace.sol#431)
        - _nativeCurrency = _currencyManager.getNativeCurrency() (contracts/traits/Marketplace.sol#437)
        - _platformFeesBasisPoints = _config.getPlatformFees() (contracts/traits/Marketplace.sol#430)
        - delete _supportedCurrencies (contracts/traits/Marketplace.sol#438)
        - _supportedCurrencies = _currencyManager.supportedCurrencies() (contracts/traits/Marketplace.sol#439)
Reentrancy in Marketplace._sendValue(address,uint256,uint256) (contracts/traits/Marketplace.sol#617-634):
        External calls:
        - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        State variables written after the call(s):
        - _pendingWithdrawals[user] = _pendingWithdrawals[user] + amount (contracts/traits/Marketplace.sol#632)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
                - IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#611)
                - IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
        External calls sending eth:
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        State variables written after the call(s):
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - _finalizedListings[listingId] = listing (contracts/traits/Marketplace.sol#572)
Reentrancy in Marketplace.reloadConfiguration() (contracts/traits/Marketplace.sol#154-158):
        External calls:
        - _reloadConfiguration() (contracts/traits/Marketplace.sol#156)
                - (_gasLimitHigh,_gasLimitMedium,_gasLimitLow) = _config.getGasLimits() (contracts/traits/Marketplace.sol#428)
        State variables written after the call(s):
        - _hasConfiguration = true (contracts/traits/Marketplace.sol#157)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563):
        External calls:
        - IERC721(_trustedToken).transferFrom(address(this),listing.sellerTokenWallet,tokenId) (contracts/traits/Marketplace.sol#560)
        Event emitted after the call(s):
        - ListingCanceled(uint8(listing.listingType),tokenId,listingId) (contracts/traits/Marketplace.sol#562)
Reentrancy in Marketplace._createListing(uint256,uint256,string,address,address,address,uint8) (contracts/traits/Marketplace.sol#442-498):
        External calls:
        - require(bool,string)(_isValidWallet(msgSender()),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#459)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _transferToken(msgSender(),address(this),tokenId) (contracts/traits/Marketplace.sol#460)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        Event emitted after the call(s):
        - ListingCreated(listingType,listingId,tokenId,msgSender(),sellerPaymentWallet,buyerTokenWallet,buyerPaymentWallet,listing.price,listing.duration,listing.extensionDuration,listing.endTime) (contracts/traits/Marketplace.sol#485-49
7)
Reentrancy in Marketplace._finaliseListing(uint256) (contracts/traits/Marketplace.sol#565-594):
        External calls:
        - _transferToken(address(this),listing.buyerTokenWallet,listing.tokenId) (contracts/traits/Marketplace.sol#576)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
        - (platformFee,sellerPayableAmount) = _distributeFunds(listing.sellerPaymentWallet,listing.bidCurrency,listing.bidAmount) (contracts/traits/Marketplace.sol#577-581)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
                - IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#611)
                - IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
        External calls sending eth:
        - (platformFee,sellerPayableAmount) = _distributeFunds(listing.sellerPaymentWallet,listing.bidCurrency,listing.bidAmount) (contracts/traits/Marketplace.sol#577-581)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        Event emitted after the call(s):
        - ListingFinalized(uint8(listing.listingType),listingId,listing.tokenId,listing.sellerTokenWallet,listing.sellerPaymentWallet,listing.buyerTokenWallet,listing.buyerPaymentWallet,platformFee,sellerPayableAmount) (contracts/traits/
Marketplace.sol#583-593)
Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - IERC721(_trustedToken).transferFrom(from,to,tokenId) (contracts/traits/Marketplace.sol#513-515)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
                - IERC20(contractAddress).transferFrom(msgSender(),_treasury,platformPayableAmount) (contracts/traits/Marketplace.sol#611)
                - IERC20(contractAddress).transferFrom(msgSender(),seller,sellerPayableAmount) (contracts/traits/Marketplace.sol#612)
        External calls sending eth:
        - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        Event emitted after the call(s):
        - ListingFinalized(uint8(listing.listingType),listingId,listing.tokenId,listing.sellerTokenWallet,listing.sellerPaymentWallet,listing.buyerTokenWallet,listing.buyerPaymentWallet,platformFee,sellerPayableAmount) (contracts/traits/
Marketplace.sol#583-593)
                - _finaliseListing(listingId) (contracts/traits/Marketplace.sol#329)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        Event emitted after the call(s):
        - AuctionBidPlaced(listing.listingId,msgSender(),bidValue,listing.endTime) (contracts/traits/Marketplace.sol#344)
Reentrancy in Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377):
        External calls:
        - require(bool,string)(_isValidWallet(buyerTokenWallet),ERR_UNSUPPORTED_TOKEN_WALLET) (contracts/traits/Marketplace.sol#306)
                - _config.isValidWallet(wallet) (contracts/traits/Marketplace.sol#501-505)
        - IERC20(currencyContractAddress).transferFrom(msgSender(),address(this),bidValue) (contracts/traits/Marketplace.sol#370)
        Event emitted after the call(s):
        - AuctionBidPlaced(listing.listingId,msgSender(),bidValue,listing.endTime) (contracts/traits/Marketplace.sol#373)
Reentrancy in Marketplace.withdrawFor(address) (contracts/traits/Marketplace.sol#401-407):
        External calls:
        - _sendValue(user,amount,_gasLimitHigh) (contracts/traits/Marketplace.sol#405)
                - (success) = user.call{gas: gasLimit,value: amount}() (contracts/traits/Marketplace.sol#627)
        Event emitted after the call(s):
        - Withdrawal(user,amount) (contracts/traits/Marketplace.sol#406)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

AUTOMATED TESTING

```
Marketplace.placeBid(uint256,address,string,uint256) (contracts/traits/Marketplace.sol#297-377) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(listing.price > 0,ERR_LISTING_NOT_EXISTS) (contracts/traits/Marketplace.sol#304)
        - require(bool,string)(listing.sellerTokenWallet != buyerTokenWallet,ERR_CANNOT_BUY_OWN_LISTING) (contracts/traits/Marketplace.sol#305)
        - require(bool,string)(listing.buyerTokenWallet == buyerTokenWallet,ERR_PRIVATE_SALE_ONLY) (contracts/traits/Marketplace.sol#320)
        - require(bool,string)(listing.price <= bidValueInListingCurrency,ERR_BID_PRICE_LOW) (contracts/traits/Marketplace.sol#323)
        - require(bool,string)(listing.price <= bidValueInListingCurrency,NFTMarketReserveAuction: Bid must be at least the reserve price) (contracts/traits/Marketplace.sol#333-336)
        - require(bool,string)(listing.endTime >= block.timestamp,NFTMarketReserveAuction: Auction is over) (contracts/traits/Marketplace.sol#347)
        - require(bool,string)(listing.buyerPaymentWallet != msgSender(),NFTMarketReserveAuction: You already have an outstanding bid) (contracts/traits/Marketplace.sol#348-351)
        - listing.endTime < block.timestamp + listing.extensionDuration (contracts/traits/Marketplace.sol#365)
Marketplace._cancelListing(uint256) (contracts/traits/Marketplace.sol#550-563) uses timestamp for comparisons
        Dangerous comparisons:
        - require(bool,string)(listing.listingId != uint256(0),ERR_LISTING_NOT_EXISTS) (contracts/traits/Marketplace.sol#552)
        - require(bool,string)(listing.sellerTokenWallet == msgSender(),ERR_LISTING_WRONG_OWNER) (contracts/traits/Marketplace.sol#553)
        - require(bool,string)(listing.endTime == 0,ERR_AUCTION_IN_PROGRESS) (contracts/traits/Marketplace.sol#554)
Marketplace._distributeFunds(address,string,uint256) (contracts/traits/Marketplace.sol#596-615) uses timestamp for comparisons
        Dangerous comparisons:
        - currencyType == 1 (contracts/traits/Marketplace.sol#605)
Marketplace._sendValue(address,uint256,uint256) (contracts/traits/Marketplace.sol#617-634) uses timestamp for comparisons
        Dangerous comparisons:
        - amount == 0 (contracts/traits/Marketplace.sol#622)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#block-timestamp

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#78-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
Counters.decrement(Counters.Counter) (contracts/lib/Counters.sol#31-37) is never used and should be removed
Counters.reset(Counters.Counter) (contracts/lib/Counters.sol#39-41) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Marketplace.initializeMarketplace(address,address) (contracts/traits/Marketplace.sol#143-147) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
ReentrancyGuard.initializeReentrancyGuard() (contracts/traits/ReentrancyGuard.sol#39-41) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Upgradeable.initializeUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

ReentrancyGuard.__gap (contracts/traits/ReentrancyGuard.sol#63) is never used in Marketplace (contracts/traits/Marketplace.sol#18-635)
Marketplace.ERR_UNSUPPORTED_CURRENCY (contracts/traits/Marketplace.sol#56) is never used in Marketplace (contracts/traits/Marketplace.sol#18-635)
Marketplace.ERR_CURRENCY_MANAGER (contracts/traits/Marketplace.sol#57) is never used in Marketplace (contracts/traits/Marketplace.sol#18-635)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

getPendingWithdrawal(address) should be declared external:
        - Marketplace.getPendingWithdrawal(address) (contracts/traits/Marketplace.sol#387-389)
withdraw() should be declared external:
        - Marketplace.withdraw() (contracts/traits/Marketplace.sol#394-396)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

traits/MarketplaceConfig.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

MarketplaceConfig._existsUserWallet(address).isValid (contracts/traits/MarketplaceConfig.sol#145) is a local variable never initialized
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#uninitialized-local-variables

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
MarketplaceConfig._existsUserWallet(address) (contracts/traits/MarketplaceConfig.sol#144-150) ignores return value by IUserDB(_rarioUserDB).existsUserWallet(wallet) (contracts/traits/MarketplaceConfig.sol#145-149)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Variable 'MarketplaceConfig._existsUserWallet(address).isValid (contracts/traits/MarketplaceConfig.sol#145)' in MarketplaceConfig._existsUserWallet(address) (contracts/traits/MarketplaceConfig.sol#144-150) potentially used before declara
tion: retval = isValid (contracts/traits/MarketplaceConfig.sol#146)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
MarketplaceConfig.initialiseMarketplaceConfig(address,address,address) (contracts/traits/MarketplaceConfig.sol#65-79) is never used and should be removed
Pausable.initialisePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Upgradeable.initialiseUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/MetadataDB.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
MetadataDB.addAttribute(string,string,MetadataDB.AttributeType,bool) (contracts/traits/MetadataDB.sol#172-193) ignores return value by _attributeNames.add(attribute) (contracts/traits/MetadataDB.sol#191)
MetadataDB.setTextValue(uint256,string,string) (contracts/traits/MetadataDB.sol#204-221) ignores return value by _tokenIds.add(tokenId) (contracts/traits/MetadataDB.sol#217)
MetadataDB.setNumberValue(uint256,string,int256) (contracts/traits/MetadataDB.sol#223-240) ignores return value by _tokenIds.add(tokenId) (contracts/traits/MetadataDB.sol#236)
MetadataDB.setBooleanValue(uint256,string,bool) (contracts/traits/MetadataDB.sol#242-259) ignores return value by _tokenIds.add(tokenId) (contracts/traits/MetadataDB.sol#255)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
EnumerableStringSet.remove(EnumerableStringSet.StringSet,string) (contracts/lib/EnumerableStringSet.sol#57-89) is never used and should be removed
EnumerableStringSet.values(EnumerableStringSet.StringSet) (contracts/lib/EnumerableStringSet.sol#127-129) is never used and should be removed
Pausable.initialisePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Upgradeable.initialiseUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/Pausable.sol

```
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Context.isTrustedForwarder(address) (contracts/traits/Context.sol#12-14) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.msgSender() (contracts/traits/Context.sol#16-25) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
Pausable._pause() (contracts/traits/Pausable.sol#72-75) is never used and should be removed
Pausable._unpause() (contracts/traits/Pausable.sol#84-87) is never used and should be removed
Pausable.initialisePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

traits/RarioEthereumNFT.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) ignores return value by IERC721Receiver(to).onERC721Received(msgSender(),from,tokenId,_data) (contracts/traits/ERC721NFT.sol#445-455
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

RarioNFT.getAttributeByIndex(uint256).name (contracts/traits/RarioNFT.sol#66) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
RarioNFT.getAttribute(string).name (contracts/traits/RarioNFT.sol#80) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).retval (contracts/traits/ERC721NFT.sol#465)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: retval == IERC721Receiver.onERC721Received.selector (contracts/traits/ERC721NFT.sol#446)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: reason.length == 0 (contracts/traits/ERC721NFT.sol#448)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (contracts/traits/ERC721NFT.sol#452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) uses assembly
        - INLINE ASM (contracts/traits/ERC721NFT.sol#451-453)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#126-128) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
ERC721NFT.initializeERC721NFT(string,string) (contracts/traits/ERC721NFT.sol#66-77) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
RarioEthereumNFT.initializeRarioEthereumNFT(string,string,address,address) (contracts/traits/RarioEthereumNFT.sol#12-16) is never used and should be removed
RarioNFT.initializeRarioNFT(string,string,address) (contracts/traits/RarioNFT.sol#19-26) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Upgradeable._authorizeUpgrade(address) (contracts/traits/Upgradeable.sol#43-45) is never used and should be removed
Upgradeable.initializeUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

RarioEthereumNFT (contracts/traits/RarioEthereumNFT.sol#8-80) does not implement functions:
        - ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ERC721NFT.E_NFT_TRANSFER_TO_NON_RECEIVER (contracts/traits/ERC721NFT.sol#30) is never used in RarioEthereumNFT (contracts/traits/RarioEthereumNFT.sol#8-80)
RarioEthereumNFT.__gap (contracts/traits/RarioEthereumNFT.sol#79) is never used in RarioEthereumNFT (contracts/traits/RarioEthereumNFT.sol#8-80)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

traits/RarioNFT.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) ignores return value by IERC721Receiver(to).onERC721Received(msgSender(),from,tokenId,_data) (contracts/traits/ERC721NFT.sol#445-455
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

RarioNFT.getAttributeByIndex(uint256).name (contracts/traits/RarioNFT.sol#66) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
RarioNFT.getAttribute(string).name (contracts/traits/RarioNFT.sol#80) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).retval (contracts/traits/ERC721NFT.sol#445)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: retval == IERC721Receiver.onERC721Received.selector (contracts/traits/ERC721NFT.sol#446)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: reason.length == 0 (contracts/traits/ERC721NFT.sol#448)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (contracts/traits/ERC721NFT.sol#452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3
```

```
Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) uses assembly
        - INLINE ASM (contracts/traits/ERC721NFT.sol#451-453)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
ERC721NFT.initializeERC721NFT(string,string) (contracts/traits/ERC721NFT.sol#66-77) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
RarioNFT.initializeRarioNFT(string,string,address) (contracts/traits/RarioNFT.sol#19-26) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Upgradeable.initializeUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

ERC721NFT.E_NFT_TRANSFER_TO_NON_RECEIVER (contracts/traits/ERC721NFT.sol#30) is never used in RarioNFT (contracts/traits/RarioNFT.sol#11-132)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

traits/RarioPolygonNFT.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) ignores return value by IERC721Receiver(to).onERC721Received(msgSender(),from,tokenId,_data) (contracts/traits/ERC721NFT.sol#445-455
)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

RarioNFT.getAttributeByIndex(uint256).name (contracts/traits/RarioNFT.sol#66) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
RarioNFT.getAttribute(string).name (contracts/traits/RarioNFT.sol#80) shadows:
        - ERC721NFT.name() (contracts/traits/ERC721NFT.sol#82-84) (function)
        - IERC721Metadata.name() (contracts/interfaces/IERC721Metadata.sol#15) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).retval (contracts/traits/ERC721NFT.sol#445)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: retval == IERC721Receiver.onERC721Received.selector (contracts/traits/ERC721NFT.sol#446)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: reason.length == 0 (contracts/traits/ERC721NFT.sol#448)
Variable 'ERC721NFT._checkOnERC721Received(address,address,uint256,bytes).reason (contracts/traits/ERC721NFT.sol#447)' in ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) potentiall
y used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (contracts/traits/ERC721NFT.sol#452)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
ERC721NFT._checkOnERC721Received(address,address,uint256,bytes) (contracts/traits/ERC721NFT.sol#438-459) uses assembly
        - INLINE ASM (contracts/traits/ERC721NFT.sol#451-453)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

```
AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
ERC721NFT.initializeERC721NFT(string,string) (contracts/traits/ERC721NFT.sol#66-77) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
RarioNFT.initializeRarioNFT(string,string,address) (contracts/traits/RarioNFT.sol#19-26) is never used and should be removed
RarioPolygonNFT.initializeRarioPolygonNFT(string,string,address,address) (contracts/traits/RarioPolygonNFT.sol#19-22) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Upgradeable._authorizeUpgrade(address) (contracts/traits/Upgradeable.sol#43-45) is never used and should be removed
Upgradeable.initializeUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

RarioPolygonNFT (contracts/traits/RarioPolygonNFT.sol#8-126) does not implement functions:
        - ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions

ERC721NFT.E_NFT_TRANSFER_TO_NON_RECEIVER (contracts/traits/ERC721NFT.sol#30) is never used in RarioPolygonNFT (contracts/traits/RarioPolygonNFT.sol#8-126)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

traits/ReentrancyGuard.sol

```
ReentrancyGuard.initializeReentrancyGuard() (contracts/traits/ReentrancyGuard.sol#39-41) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

ReentrancyGuard.__gap (contracts/traits/ReentrancyGuard.sol#63) is never used in ReentrancyGuard (contracts/traits/ReentrancyGuard.sol#22-64)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable
```

traits/Upgradeable.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#79-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.add(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#296-298) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.at(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#334-336) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.contains(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#313-315) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.length(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#320-322) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Upgradeable._authorizeUpgrade(address) (contracts/traits/Upgradeable.sol#43-45) is never used and should be removed
Upgradeable.initializeUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Upgradeable (contracts/traits/Upgradeable.sol#10-46) does not implement functions:
        - ERC1822UUPS._authorizeUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unimplemented-functions
```

traits/UserDB.sol

```
ERC1967Upgrade._functionDelegateCall(address,bytes) (contracts/traits/ERC1967Upgrade.sol#200-206) uses delegatecall to a input-controlled function id
        - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#controlled-delegatecall

AccessControl._grantRole(bytes32,address) (contracts/traits/AccessControl.sol#227-233) ignores return value by _roleMembers[role].add(account) (contracts/traits/AccessControl.sol#230)
AccessControl._revokeRole(bytes32,address) (contracts/traits/AccessControl.sol#235-241) ignores return value by _roleMembers[role].remove(account) (contracts/traits/AccessControl.sol#238)
UserDB.addUser(uint256,address) (contracts/traits/UserDB.sol#41-51) ignores return value by rarioIds.add(rarioId) (contracts/traits/UserDB.sol#47)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

Reentrancy in ERC1967Upgrade._upgradeToAndCallSecure(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#82-107):
        External calls:
        - _functionDelegateCall(newImplementation,data) (contracts/traits/ERC1967Upgrade.sol#92)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        - _functionDelegateCall(newImplementation,abi.encodeWithSignature(upgradeTo(address),oldImplementation)) (contracts/traits/ERC1967Upgrade.sol#100)
                - (success,returndata) = target.delegatecall(data) (contracts/traits/ERC1967Upgrade.sol#204)
        Event emitted after the call(s):
        - Upgraded(newImplementation) (contracts/traits/ERC1967Upgrade.sol#58)
                - _upgradeTo(newImplementation) (contracts/traits/ERC1967Upgrade.sol#105)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.isContract(address) (contracts/lib/Address.sol#26-36) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#32-34)
Address.verifyCallResult(bool,bytes,string) (contracts/lib/Address.sol#168-188) uses assembly
        - INLINE ASM (contracts/lib/Address.sol#180-183)
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#277-279)
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) uses assembly
        - INLINE ASM (contracts/lib/EnumerableSet.sol#350-352)
StorageSlot.getAddressSlot(bytes32) (contracts/lib/StorageSlot.sol#51-55) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#52-54)
StorageSlot.getBooleanSlot(bytes32) (contracts/lib/StorageSlot.sol#60-64) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#61-63)
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#70-72)
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) uses assembly
        - INLINE ASM (contracts/lib/StorageSlot.sol#79-81)
Context.msgSender() (contracts/traits/Context.sol#16-25) uses assembly
        - INLINE ASM (contracts/traits/Context.sol#19-21)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

AccessControl._setupRole(bytes32,address) (contracts/traits/AccessControl.sol#212-214) is never used and should be removed
AccessControl.initializeAccessControl() (contracts/traits/AccessControl.sol#56-59) is never used and should be removed
Address.functionCall(address,bytes) (contracts/lib/Address.sol#78-81) is never used and should be removed
Address.functionCall(address,bytes,string) (contracts/lib/Address.sol#89-95) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (contracts/lib/Address.sol#108-114) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (contracts/lib/Address.sol#122-133) is never used and should be removed
Address.functionStaticCall(address,bytes) (contracts/lib/Address.sol#141-143) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (contracts/lib/Address.sol#151-160) is never used and should be removed
Address.sendValue(address,uint256) (contracts/lib/Address.sol#54-59) is never used and should be removed
Context.msgData() (contracts/traits/Context.sol#27-33) is never used and should be removed
Context.setTrustedForwarder(address) (contracts/traits/Context.sol#8-10) is never used and should be removed
ERC165Interface.initializeERC165Interface() (contracts/traits/ERC165Interface.sol#20-22) is never used and should be removed
ERC165Interface.registerInterface(bytes4) (contracts/traits/ERC165Interface.sol#42-45) is never used and should be removed
ERC1822UUPS._authoriseUpgrade(address) (contracts/traits/ERC1822UUPS.sol#60) is never used and should be removed
ERC1822UUPS.initializeERC1822UUPS() (contracts/traits/ERC1822UUPS.sol#21-23) is never used and should be removed
ERC1967Upgrade._changeAdmin(address) (contracts/traits/ERC1967Upgrade.sol#141-144) is never used and should be removed
ERC1967Upgrade._getAdmin() (contracts/traits/ERC1967Upgrade.sol#124-126) is never used and should be removed
ERC1967Upgrade._getBeacon() (contracts/traits/ERC1967Upgrade.sol#160-162) is never used and should be removed
ERC1967Upgrade._setAdmin(address) (contracts/traits/ERC1967Upgrade.sol#131-134) is never used and should be removed
ERC1967Upgrade._setBeacon(address) (contracts/traits/ERC1967Upgrade.sol#167-174) is never used and should be removed
ERC1967Upgrade._upgradeBeaconToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#182-192) is never used and should be removed
ERC1967Upgrade._upgradeToAndCall(address,bytes,bool) (contracts/traits/ERC1967Upgrade.sol#66-75) is never used and should be removed
ERC1967Upgrade.initializeERC1967Upgrade() (contracts/traits/ERC1967Upgrade.sol#19) is never used and should be removed
EnumerableSet._values(EnumerableSet.Set) (contracts/lib/EnumerableSet.sol#141-143) is never used and should be removed
EnumerableSet.add(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#157-159) is never used and should be removed
EnumerableSet.at(EnumerableSet.Bytes32Set,uint256) (contracts/lib/EnumerableSet.sol#195-197) is never used and should be removed
EnumerableSet.contains(EnumerableSet.AddressSet,address) (contracts/lib/EnumerableSet.sol#240-242) is never used and should be removed
EnumerableSet.contains(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#174-176) is never used and should be removed
EnumerableSet.length(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#181-183) is never used and should be removed
EnumerableSet.remove(EnumerableSet.Bytes32Set,bytes32) (contracts/lib/EnumerableSet.sol#167-169) is never used and should be removed
EnumerableSet.remove(EnumerableSet.UintSet,uint256) (contracts/lib/EnumerableSet.sol#306-308) is never used and should be removed
EnumerableSet.values(EnumerableSet.AddressSet) (contracts/lib/EnumerableSet.sol#273-282) is never used and should be removed
EnumerableSet.values(EnumerableSet.Bytes32Set) (contracts/lib/EnumerableSet.sol#207-209) is never used and should be removed
EnumerableSet.values(EnumerableSet.UintSet) (contracts/lib/EnumerableSet.sol#346-355) is never used and should be removed
Pausable.initializePausable() (contracts/traits/Pausable.sol#30-32) is never used and should be removed
StorageSlot.getBytes32Slot(bytes32) (contracts/lib/StorageSlot.sol#69-73) is never used and should be removed
StorageSlot.getUint256Slot(bytes32) (contracts/lib/StorageSlot.sol#78-82) is never used and should be removed
Strings.equals(string,string) (contracts/lib/Strings.sol#67-71) is never used and should be removed
Strings.toHexString(uint256) (contracts/lib/Strings.sol#39-50) is never used and should be removed
Strings.toString(uint256) (contracts/lib/Strings.sol#14-34) is never used and should be removed
Upgradeable.initialiseUpgradeable(address) (contracts/traits/Upgradeable.sol#16-26) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

- The reentrancies flagged in Marketplace.sol are false positives although it's recommended to add the nonReentrant modifier in all the external functions.
- The unprotected-upgradeable-contract alerts are false positives. Although the contracts should be deployed using a factory pattern to prevent front-running.

THANK YOU FOR CHOOSING

// HALBORN