# MetaX Self-developed Trade Aggregation

## Contract Audit Report

VER 1.0

9 Apr 2022

No. 2022040919051

# Project Summary

## 1. Project introduction

MetaX self-developed trade aggregation version 2.0 mainly call the unoswap contract to guide the intelligent path of user Dex transactions, acquiring the most suitable price and exchange path through the backend, Assembling the relevant parameters of unoswap and Returinng it to the user's wallet for signing transactions. The unoswap contract uses the assembly-like assembly to control the size of the memory space, so as to achieve the purpose of saving gas costs.

## 2. Audit Summary

| | | | |
|---|---|---|---|
| Project Name | MetaX self-developed trade aggregation ver 2.0 | Platform | N/A |
| Token | N/A | Token symbol | N/A |
| Start date | 28 Mar 2022 | Language | Solidity |
| End date | 9 Apr 2022 | Website | N/A |
| Github | https://github.com/okex/sor_smartcontract_dev/tree/3449df35523ee44b657c696c0ae927e63442cb96 | Introduction | N/A |

## 3. Audit Scope

| ID | File | SHA-256 |
|---|---|---|
| contracts/8 | DexRouter.sol | 764857b6d0a111f1a0ced545b643f92b07357280dc4841da2d167506c3af0896 |
| contracts/8 | TemplateCode.sol | 575ede11219ab04b46c4e4d0dc89e26a08213183006432f88bb5402b12653b7a |
| contracts/8 | TokenApprove.sol | 00aa8da5a7bafd7096a3bdef244a6d735dd3f19102f235363a49939973aa1e6e |
| contracts/8 | TokenApproveProxy.sol | e4afef05af86140fea22f15893b5c64753c1bcba0e2193f00fddff955b2b7c1b |

| ID | File | SHA-256 |
|---|---|---|
| contracts/8 | UnxswapRouter.sol | 9ef5cab6f9e3a64170d364bb33a495d80e91f9cccaf9a8059b4d219ef253dd02 |
| contracts/8 | WNativeRelayer.sol | 9dee3f4d993fee0801b19eac232ab2bb285a12fcb84a4373fc15a37c8c5deee4 |
| contracts/8/adapter | ApeAdapter.sol | 9ab2c4760382b74eea188bf58b7ebae47e32860a2bbbd3c42c6132902f6a9df2 |
| contracts/8/adapter | BakeryAdapter.sol | fb32261e5db033693494e5c7f6bead3c549d389cab0ff65170bb3c9dabda3528 |
| contracts/8/adapter | BalancerAdapter.sol | 23fd13455c5402b86fd18793a3e88c77c95bd4506fccbccadeb1f21499bc1bb9 |
| contracts/8/adapter | BalancerV2Adapter.sol | 0d0ec2028b93e58836881263a6afd63de5fa24675e6eeb7f16aca4f349918224 |
| contracts/8/adapter | BiAdapter.sol | 7b85a3610393afe3ca85e0b550e2e7b6c111cbcb2c2f96b16b293ef54b37296b |
| contracts/8/adapter | CurveAdapter.sol | 9bc346f948b202e6a159cc446d05970b674feb7adb067a54b827d873b2303f2a |
| contracts/8/adapter | DODOV2Adapter.sol | 924a8cf3f28c9bf24e9d16aebefe68f67fcfb85bbf0b05d5d4d486b8912cba56 |
| contracts/8/adapter | KyberAdapter.sol | bf17ffb24c85ec69b81b9b3b8526642af0c12a0b888d1e8dc50cfd9354a0f1bf |
| contracts/8/adapter | PancakeAdapter.sol | a5e7d17791898adce33562314d59f0d0e18153baf9ccf551ff6a2f8f421a7e67 |
| contracts/8/adapter | UniAdapter.sol | f9e001f48e49d07c9164a4849e37878fcc9161f6709493c781a65dd0b9989265 |
| contracts/8/adapter | UniV3Adapter.sol | b0fc4f24761d1fa26e2ea5d9727ef2654c8e3702a28fcb281b3a0b5345ca13c5 |
| contracts/8/libraries | Address.sol | a849d376bae9cd0ca314f4d5e457418e680a270e31e9cd7085ff20b73844cf97 |
| contracts/8/libraries | AddressSetLib.sol | 22986ba3c8fb072b13f37dd82fe9289f576f2a91a9c9ce423b5f6884bede807e |
| contracts/8/libraries | DecimalMath.sol | f9d5559ab312cc7c534386edb66493250765816489125d95cff74abfa9863203 |
| contracts/8/libraries | RevertReasonParser.sol | 29c90766559ad1d0d6f677f3f6bf7fde36d858753c2e3f8a1e450815347ba0c3 |
| contracts/8/libraries | SafeERC20.sol | dc6b90188809ff0cd7d7e9c7ccea53b8459fa929b1e727068e17f1646565db46 |
| contracts/8/libraries | SafeMath.sol | 80a3c28bfb9fd44113a933c19f6501f1fa01db0774643434fd16fe53db2dd5f1 |
| contracts/8/libraries | TickMath.sol | fe302c2050009fe429ad346dc9abef068df40b4e90f0f622889c1ecce0bc8d6d |
| contracts/8/libraries | UniversalERC20.sol | e1b01372b42877d85a8b88ef0f7bbccb53ab9ae7ff9255dc272463a34cb5c0aa |
| contracts/8/libraries | ZeroCopySink.sol | fc61fc798880c3fe0c003c7d7a0b776587697333e57d47ec4c855aa1c7d15413 |
| contracts/8/libraries | ZeroCopySource.sol | 4c3907626b4f52ba852858b854bd3ed004db8d8646db5cc0a39f4b37c8b72c7e |

# 4. Code Structure

```
├─── DexRouter.sol
├─── TemplateCode.sol
├─── TokenApprove.sol
├─── TokenApproveProxy.sol
├─── UnxswapRouter.sol
├─── WNativeRelayer.sol
├─── adapter
|    ├─── ApeAdapter.sol
|    ├─── BakeryAdapter.sol
|    ├─── BalancerAdapter.sol
|    ├─── BalancerV2Adapter.sol
|    ├─── BiAdapter.sol
|    ├─── CurveAdapter.sol
|    ├─── DODOV2Adapter.sol
|    ├─── KyberAdapter.sol
|    ├─── PancakeAdapter.sol
|    ├─── UniAdapter.sol
|    └─── UniV3Adapter.sol
├─── libraries
  ├─── Address.sol
  ├─── AddressSetLib.sol
  ├─── DecimalMath.sol
  ├─── RevertReasonParser.sol
  ├─── SafeERC20.sol
  ├─── SafeMath.sol
  ├─── TickMath.sol
  ├─── UniversalERC20.sol
  ├─── ZeroCopySink.sol
  └─── ZeroCopySource.sol
```

# Audit Report Summary

## 1. Audit Methodology

The audit was conducted to gain a clear understanding of how the project was implemented and how it works. The audit team conducted in-depth research, analysis, and testing of the project code and collected detailed data. In this report, the audit team will list in detail each issue identified, where it is located, the root cause of the issue, and a description of the issue, and will recommend changes to the issue accordingly.

| | |
|---|---|
| **Audit Methodology** | Static analysis, Manual Review |

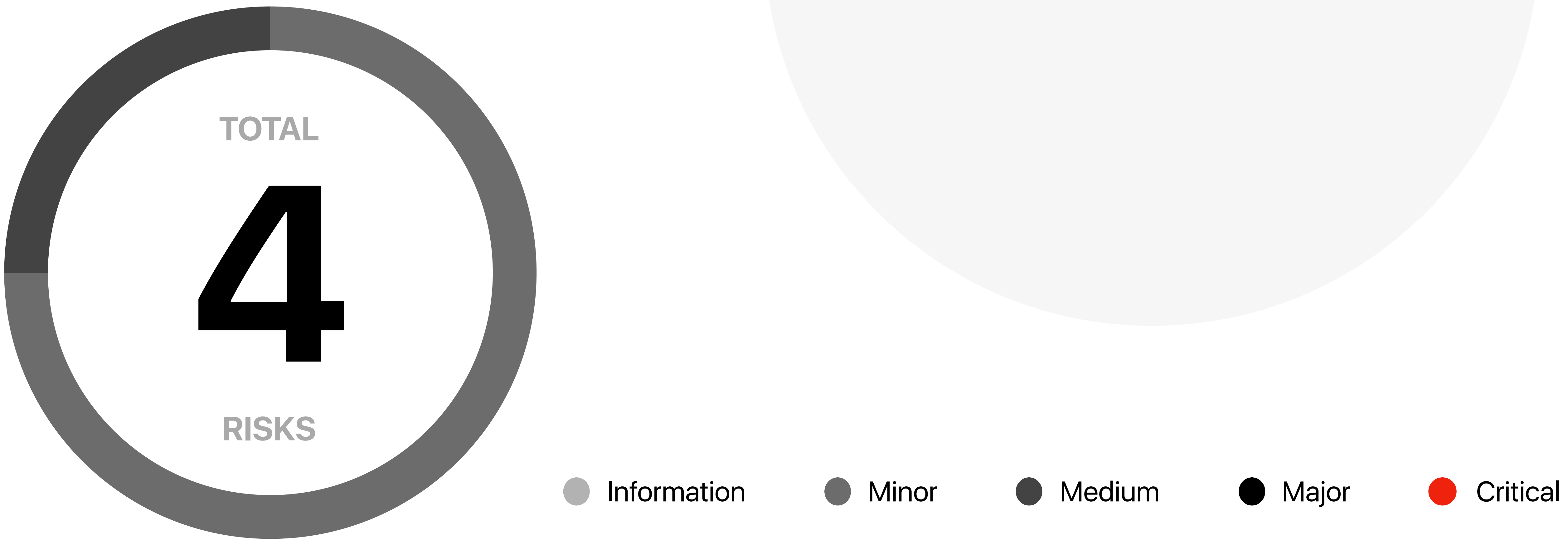
## 2. Audit Process

| Steps | Operation | Description |
|---|---|---|
| 1 | Background | Read project descriptions, white papers, contract source code, and other relevant information the project team provides to ensure a proper understanding of project functions. |
| 2 | Automated testing | Scanning source code mainly with automated tools to find common potential vulnerabilities. |
| 3 | Manual reveiw | Engineers read the code line by line to find potential vulnerabilities. |
| 4 | Logical proofread | The engineer will compare the understanding of the code with the information provided by the project and check whether the code implementation is in line with the project white paper information. |
| 5 | Test case | Including test case design, test scope analysis, symbolic execution, etc. |
| 6 | Optimization items | Review of projects in terms of maintainability, safety, and operability based on application scenarios, deployment methods, and latest research results. |

## 3. Risk Levels

| Risk level | Description |
|---|---|
| **Critical** | Fatal risks and hazards that need to fixed immediately. |
| **Major** | Some high risks and hazards that will lead to related problems that must be solved |
| **Medium** | Some moderate risks and pitfalls may lead to potential risks that will eventually need to be addressed |
| **Minor** | There are low risks and hazards, mainly details of various types of mishandling or warning messages, which can be set aside for the time being |
| **Information** | Some parts can be optimized, such problems can be shelved, but it is recommended that the final solution |

## 4. Audit Results



| ID | Audit Project | Risk Level | Status |
|---|---|---|---|
| 1 | Reentrancy | None | |
| 2 | Injection | None | |
| 3 | Authentication bypass | None | |
| 4 | MEV Possibility | None | |
| 5 | Revert | None | |
| 6 | Race condition | None | |
| 7 | Insufficient Gas Griefing | None | |
| 8 | The major impact of flash loan | None | |
| 9 | Unreasonable economic model | None | |
| 10 | Predictable random nubmber | None | |
| 11 | Voting rights management confusion | None | |

| ID | Audit project | Risk level | Status |
|---|---|---|---|
| 12 | Privacy leak | None | |
| 13 | Improper use of time on chain | None | |
| 14 | Improper codes in fallback function | None | |
| 15 | Improper identification | None | |
| 16 | Inappropriate opcode | None | |
| 17 | Inappropriate assembly | None | |
| 18 | Constructor irregularities | None | |
| 19 | Return value irregularity | None | |
| 20 | Event irregularity | None | |
| 21 | Keywords irregularity | None | |
| 22 | Not following ERC standards | None | |
| 23 | Irregularity of condition judgment | None | |
| 24 | Risk of liquidity drain | None | |
| 25 | Centralization Risk | Medium | Acknowledged |
| 26 | Logic change risk | None | |
| 27 | Integer overflow | None | |
| 28 | Improper function visibility | None | |
| 29 | Improper initialization of variables | Minor | Acknowledged |
| 30 | Improper contract calls | None | |
| 31 | Variable irregularities | None | |
| 32 | Replay | None | |
| 33 | Write to Arbitrary Storage Location | None | |
| 34 | Honeypot logic | None | |
| 35 | Hash collision | None | |
| 36 | Improper logic in receiving awards | None | |
| 37 | Use the not recommended method | None | |
| 38 | Basic coding principles were not followed | Minor | Acknowledged |

* In the above table, if the status column is “**Acknowledged**”, the audit team has informed the project owner of the vulnerability. Still, the project owner has not made any changes to the vulnerability or has not announced to the audit team the progress of the changes to the vulnerability. If the status column is “**Resolved**”, the project owner has made changes to the vulnerability, and the audit team has confirmed the changes.

## 5. Risk and Modification program

The following section provides detailed information about the risk items learned after the audit, including the type of risk, risk level, location of the issue, description of the problem, recommendations for changes, and feedback from the project owner.

| | | | |
|---|---|---|---|
| **Risk type** | **Centralization Risk** | **Risk Level** | **Medium** |
| **Location** | all | Contract File | DexRouter.sol |
| **Description** | The user can call the contract parameters to depend on the backend service | | |
| **Recommedation** | Signature verification of fields assembled by backend services | | |
| **Update** | 1.This is determined by the project architecture, which trusts the calldata data of the back-end assembly.<br>2.The dexRouter does not store any assets, and the target assets are immediately transferred to the caller according to the sliding range after the flashing. | | |

| | | | |
|---|---|---|---|
| **Risk type** | **Improper initialization of variables** | **Risk Level** | **Minor** |
| **Location** | Line 70、Line 74、Line 78 | Contract File | UnxswapRouter.sol |
| **Description** | There is a hard-coded address, which will make the contract unavailable when there is a risk of relying on the contract | | |
| **Recommedation** | Try to pass parameters from the administrator | | |
| **Update** | The address of the hardcode consists of the WETH address and our contract address, the WETH address does not change and our contract address can be controlled (the tokenApprove contract is functionally unique and does not contain business logic, only token authorization) | | |

| | | | |
|---|---|---|---|
| **Risk type** | **Basic coding principles were not followed** | **Risk Level** | **Minor** |
| **Location** | all | Contract File | UnxswapRouter.sol |
| **Description** | Extensive use of assembly language | | |
| **Recommedation** | Although reusing memory space through assembly can save gas, it will result in poor readability, and there will also be the risk of access and destruction of unsafe storage locations. It is recommended to add comments and documentation to strictly control storage access locations. | | |
| **Update** | In the main network, we still need to consider the cost of gas, so we use the convergence method to achieve the goal of saving money. Unxswap is mainly for small amount conversion, so it is more sensitive to gas. | | |

| | | | |
|---|---|---|---|
| **Risk type** | **Basic coding principles were not followed** | **Risk Level** | **Minor** |
| **Location** | all | Contract File | DexRouter.sol |
| **Description** | The calls of V1 and V2 are mixed together, which does not conform to the principle of one-to-one coding | | |
| **Recommedation** | Separate the contract entries of V1 and V2 | | |
| **Update** | Currently, DexRouter is used universally for route execution, and two business scenarios (unxswap and smartSwap) are implemented by calling different functions,<br>The two business scenarios (unxswap, smartSwap) are implemented using different functions. This is also a service design principle. There is no need to split into two contracts. | | |

## 6. Recommandation

N/A

# Disclaimer

i. This audit report focuses only on the types of audits identified in the final report issued. Other unknown security vulnerabilities are not part of this audit, and we do not accept responsibility for them.

ii. We shall only issue an audit report based on an attack or vulnerability that existed or occurred before the issuance of the audit report. We cannot determine the likely impact on the security posture of our projects for new attacks or vulnerabilities that may exist or occur in the future, and we are not responsible for them.

iii. The security audit analysis and other elements of our published audit report shall be based solely on documents and materials (including, but not limited to, contract codes) provided to us by the Project Party before the release of the audit report. Such documents and materials shall not be untrue, inaccurate, uninformative, altered, deleted, or concealed, and if the documents and materials provided by the Project Party are false, inaccurate, uninformative, changed, deleted or hidden, or if the documents and materials provided by the Project Party are untrue, inaccurate, uninformative, altered, deleted or concealed, or if the documents and materials provided by the Project Party are uninformative, uninformative, altered, deleted or hidden. If the records and information provided by the Project Party are untrue, inaccurate, uninformative, altered, deleted, or concealed, or if changes to such documents and information are made after the issuance of the audit report, we shall not be liable for any loss or adverse effect arising from any inconsistency between the reflected and actual conditions.

iv. The Project Parties are aware that our audit report is based on documents and information provided by the Project Parties and relies on the technology currently available. However, due to the technical limitations of any organization, there is a possibility that our audit report may not fully detect all risks. Our audit team encourages the project development team and any interested parties to conduct subsequent testing and audits of the project.

v. The project owner warrants that the project for which we are engaged to provide audit or testing services is legal, compliant, and does not violate applicable laws. The audit report is for the project owner's reference only, and the contents, manner of obtaining, use of, and any services or resources involved in the audit report shall not be relied upon for investment, tax, legal, regulatory, or advisory purposes of any kind, and we shall not be liable therefor. The Project Party shall not refer to, quote, display, or send the Audit Report in whole or in part to any third party without our prior written consent. The Project Party shall bear any loss or liability arising from that place. We assume no responsibility for any reliance on or use of the audit report for any purpose.

vi. This audit report does not cover the compiler of the contract or any areas beyond the programming language of the Smart Contract. The risk and liability of the audited Smart Contract arising from references to off-chain information or resources is the sole responsibility of the project party.

vii. Force Majeure. Force majeure means an unforeseen event whose occurrence and consequences cannot be avoided and cannot be overcome by the parties at the time of entering into the contract, including but not limited to natural disasters such as war, typhoon, flood, fire, earthquake, tidal wave, lightning, natural disaster, strike, nuclear explosion, epidemic and other unforeseen events such as changes in laws, regulations and policies and governmental acts, whose occurrence and consequences cannot be prevented or avoided, and which contains, affects or delays the performance by either party of all or part of its obligations under the contract.

viii. Suppose either party believes that the occurrence of force majeure affects the performance of its obligations under this Agreement. In that case, it shall promptly notify the other party and, depending on the extent of the effect of the event on the performance of the Agreement; the parties shall consult to determine whether to terminate the Agreement or partially relieve itself of its obligations to perform the Agreement, or to extend the performance of the Agreement.

ix. In force majeure, neither party shall be deemed in breach or non-performance of its obligations under this Agreement. Any financial commitments existing before the event shall not be affected, and the project party shall make payment for work performed by us.

# PASSED.

| | |
|---|---|
| **DATE** | 9 Apr 2022 |
| **AUDITOR** | 歐科雲鏈 |

This audit aims to review the metaX self-develped trade aggregation, written in Solidity language based on the function of trade aggregation, study its design architecture, discover potential security risks, and try to find possible vulnerabilities.