

Report

v. 1.0

Customer  
Term Structure Labs



Smart Contract Audit

# TermMax V2

8th October 2025

# Contents

<b>1 Changelog</b>	<b>3</b>
<b>2 Introduction</b>	<b>4</b>
<b>3 Project scope</b>	<b>5</b>
<b>4 Methodology</b>	<b>6</b>
<b>5 Our findings</b>	<b>7</b>
<b>6 Moderate Issues</b>	<b>8</b>
CVF-1. INFO .....	8
CVF-2. INFO .....	8
CVF-3. FIXED .....	9
CVF-4. FIXED .....	9
<b>7 Recommendations</b>	<b>10</b>
CVF-5. INFO .....	10
CVF-6. INFO .....	10
CVF-7. FIXED .....	10
CVF-8. INFO .....	11
CVF-9. INFO .....	11
CVF-10. INFO .....	11
CVF-11. INFO .....	11
CVF-12. INFO .....	12
CVF-13. INFO .....	12
CVF-14. INFO .....	12
CVF-15. INFO .....	13
CVF-16. INFO .....	13
CVF-17. INFO .....	13
CVF-18. INFO .....	14
CVF-19. INFO .....	14
CVF-20. INFO .....	14
CVF-21. INFO .....	15
CVF-22. INFO .....	15
CVF-23. INFO .....	16
CVF-24. INFO .....	16
CVF-25. INFO .....	16
CVF-26. INFO .....	17
CVF-27. INFO .....	17
CVF-28. INFO .....	17
CVF-29. INFO .....	18
CVF-30. INFO .....	18

# 1 Changelog

#	Date	Author	Description
0.1	08.10.25	A. Zveryanskaya	Initial Draft
0.2	08.10.25	A. Zveryanskaya	Minor revision
1.0	08.10.25	A. Zveryanskaya	Release

## 2 Introduction

All modifications to this document are prohibited. Violators will be prosecuted to the full extent of the U.S. law.

The following document provides the result of the audit performed by ABDK Consulting (Mikhail Vladimirov and Dmitry Khovratovich) at the customer request. The audit goal is a general review of the smart contracts structure, critical/major bugs detection and issuing the general recommendations.

TermMax is a DeFi protocol that simplifies DeFi borrowing, lending, and leveraging with one-click token trading, enabling predictable borrowing costs and stable returns.

# 3 Project scope

We have been asked to review the [code diff](#) between V1 and V2 of TermMax along with the related [fixes](#).

Files:

/	TermMaxMarketV2.sol	TermMaxOrderV2.sol
<b>factory/</b>		
	TermMax4626Factory.sol	TermMaxPrice FeedFactoryV2.sol
<b>oracle/priceFeeds/</b>		
	TermMaxConstant PriceFeed.sol	TermMaxERC4626 PriceFeed.sol
<b>oracle/</b>		
	OracleAggregatorV2.sol	OracleAggregator WithSequencerV2.sol
<b>router/swapAdapters/</b>		
	TermMaxSwapAdapter.sol	
<b>router/</b>		
	MakerHelper.sol	TermMaxRouterV2.sol
<b>vault/</b>		
	OrderManagerV2.sol	TermMaxVaultV2.sol

# 4 Methodology

The methodology is not a strict formal procedure, but rather a selection of methods and tactics combined differently and tuned for each particular project, depending on the project structure and technologies used, as well as on client expectations from the audit.

- **General Code Assessment.** The code is reviewed for clarity, consistency, style, and for whether it follows best code practices applicable to the particular programming language used. We check indentation, naming convention, commented code blocks, code duplication, confusing names, confusing, irrelevant, or missing comments etc. At this phase we also understand overall code structure.
- **Entity Usage Analysis.** Usages of various entities defined in the code are analysed. This includes both: internal usages from other parts of the code as well as potential external usages. We check that entities are defined in proper places as well as their visibility scopes and access levels are relevant. At this phase, we understand overall system architecture and how different parts of the code are related to each other.
- **Access Control Analysis.** For those entities, that could be accessed externally, access control measures are analysed. We check that access control is relevant and done properly. At this phase, we understand user roles and permissions, as well as what assets the system ought to protect.
- **Code Logic Analysis.** The code logic of particular functions is analysed for correctness and efficiency. We check whether the code actually does what it is supposed to do, whether the algorithms are optimal and correct, and whether proper data types are used. We also make sure that external libraries used in the code are up to date and relevant to the tasks they solve in the code. At this phase we also understand data structures used and the purposes they are used for.

We classify issues by the following severity levels:

- **Critical issue** directly affects the smart contract functionality and may cause a significant loss.
- **Major issue** is either a solid performance problem or a sign of misuse: a slight code modification or environment change may lead to loss of funds or data. Sometimes it is an abuse of unclear code behaviour which should be double checked.
- **Moderate issue** is not an immediate problem, but rather suboptimal performance in edge cases, an obviously bad code practice, or a situation where the code is correct only in certain business flows.
- **Recommendations** contain code style, best practices and other suggestions.

# 5 Our findings

We've provided the client with some recommendations.

**Moderate**

Info  
**2**

Fixed  
**2**

**Minor**

Info  
**25**

Fixed  
**1**



# 6 Moderate Issues

## CVF-1 INFO

- **Category** Flaw
- **Source** TermMaxSwapAdapter.sol

**Description** The output balance check is unreliable, as the recipient could implement a token transfer callback that synchronously forwards received tokens or changes the recipient's balance in some other way.

**Recommendation** Don't rely on the recipient's balance change.

**Client Comment** *Thank you for your suggestion. This check is only to enhance the security of interacting with TermMax. Neither EOA accounts nor our contracts will be affected. This is not a real issue.*

```
50 +uint256 outputTokenBalanceBefore = tokenOut.balanceOf(recipient);
```

```
86 +uint256 actualOutput = tokenOut.balanceOf(recipient) -  
    ↪ outputTokenBalanceBefore;  
+if (actualInput != finalInput || actualOutput != finalOutput) {  
+    revert ActualTokenBalanceNotMatch();
```

## CVF-2 INFO

- **Category** Procedural
- **Source** TermMaxERC4626PriceFeed.sol

**Description** In ERC20 the "decimals" property is used by UI to render token amounts in a human-readable way. Using this property in smart contracts is discouraged.

**Recommendation** Treat all token amounts as integers.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
36 +uint8 vaultDecimals = IERC20Metadata(_vault).decimals();  
+vaultDenominator = 10 ** vaultDecimals;  
+uint8 underlyingDecimals = IERC20Metadata(IERC4626(_vault).asset())  
    ↪ .decimals();  
+priceDenominator = 10 ** (assetPriceFeed.decimals() +  
    ↪ underlyingDecimals);
```



## CVF-3 FIXED

- **Category** Procedural
- **Source** TermMax4626Factory.sol

**Recommendation** Address parameters should be indexed.

- 13    +**event** TermMax4626FactoryInitialized(**address** aavePool, **uint16**  
      ↳ aaveReferralCode, **address** stableERC4626For4626Implementation,  
      ↳ **address** stableERC4626ForAaveImplementation, **address**  
      ↳ variableERC4626ForAaveImplementation);
- 15    +**event** StableERC4626For4626Created(**address indexed** caller, **address**  
      ↳ stableERC4626For4626);
- 17    +**event** StableERC4626ForAaveCreated(**address indexed** caller, **address**  
      ↳ stableERC4626ForAave);
- 19    +**event** VariableERC4626ForAaveCreated(**address indexed** caller, **address**  
      ↳ variableERC4626ForAave);

## CVF-4 FIXED

- **Category** Procedural
- **Source** TermMaxOrderV2.sol (ONLY YELLOW)

**Description** The comment isn't accurate anymore, as the function actually always reverts.

**Recommendation** Reflect in the comment that the function is actually disabled.

- 427    \* @notice Update fee configuration (only callable by market)
- 433    + **revert** OrderErrorsV2.FeeConfigCannotBeUpdated();

# 7 Recommendations

## CVF-5 INFO

- **Category** Suboptimal
- **Source** OrderManagerV2.sol

**Recommendation** It would be more efficient to pass a single array of structs with two fields, rather than two parallel arrays. This would also make the length check unnecessary.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

47    +**function** updateOrdersConfiguration(**address**[] **memory** orders,  
    ↳ OrderV2ConfigurationParams[] **memory** orderConfigs)

72    +**function** removeLiquidityFromOrders(IERC20 asset, **address**[] **memory**  
    ↳ orders, **uint256**[] **memory** removedLiquidities)

## CVF-6 INFO

- **Category** Bad datatype
- **Source** TermMaxVaultV2.sol

**Recommendation** The type for the “order” argument should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

401    +**function** withdrawFts(**address** order, **uint256** amount, **address**  
    ↳ recipient, **address** owner) **external** virtual nonReentrant  
    ↳ **returns** (**uint256** shares) {

## CVF-7 FIXED

- **Category** Suboptimal
- **Source** TermMaxVaultV2.sol

**Recommendation** This variable is redundant, as “msg.sender” is cheaper to access, then a local variable.

402    +**address** caller = **msg.sender**;



## CVF-8 INFO

- **Category** Procedural
- **Source** TermMaxSwapAdapter.sol

**Description** We didn't review this file.

**Client Comment** *It was audited in previous audit.*

9 +**import** {ArrayUtilsV2} from ".../lib/ArrayUtilsV2.sol";

## CVF-9 INFO

- **Category** Suboptimal
- **Source** TermMaxSwapAdapter.sol

**Recommendation** This error could be made more useful by adding certain parameters into it.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

38 +**error** ActualTokenBalanceNotMatch();

## CVF-10 INFO

- **Category** Procedural
- **Source** MakerHelper.sol

**Description** We didn't review this file.

**Client Comment** *It's the events need not be audited.*

21 +**import** {MakerHelperEvents} from ".../events/MakerHelperEvents.sol";

## CVF-11 INFO

- **Category** Procedural
- **Source** TermMaxRouterV2.sol

**Description** We didn't review this file.

**Client Comment** *It's the events need not be audited.*

31 +**import** {RouterEventsV2} from ".../events/RouterEventsV2.sol";



## CVF-12 INFO

- **Category** Bad datatype
- **Source**  
TermMaxERC4626PriceFeed.sol

**Recommendation** The type for the “\_underlyingPriceFeed” argument should be “AggregatorV3Interface”.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

33    +constructor(address \_underlyingPriceFeed, address \_vault) {

## CVF-13 INFO

- **Category** Bad datatype
- **Source**  
TermMaxERC4626PriceFeed.sol

**Recommendation** The type for the “\_vault” argument should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

33    +constructor(address \_underlyingPriceFeed, address \_vault) {

## CVF-14 INFO

- **Category** Documentation
- **Source**  
TermMaxERC4626PriceFeed.sol

**Description** The semantics of the returned values is unclear.

**Recommendation** Give descriptive names to the returned values and/or explain in a documentation comment.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

70    function latestRoundData() external view returns (uint80, int256,  
    ↳ uint256, uint256, uint80) {



## CVF-15 INFO

- **Category** Procedural

- **Source**

TermMaxConstantPriceFeed.sol

**Description** Consider specifying as “^0.8.0” unless there is something special regarding this particular version.

**Recommendation** See also: TermMax4626Factory.sol, OracleAggregatorWithSequencerV2.sol.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

2 +**pragma solidity** ^0.8.27;

## CVF-16 INFO

- **Category** Bad datatype

- **Source**

TermMaxConstantPriceFeed.sol

**Recommendation** The return type should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

43 +**function** asset() **external view override returns** (**address**) {

## CVF-17 INFO

- **Category** Documentation

- **Source** OracleAggregatorV2.sol

**Description** The semantics of the returned values is unclear.

**Recommendation** Give descriptive names to the returned values and/or explain in a documentation comment.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

203 +**function** getPrice(**address** asset) **public view virtual override**  
    **returns** (**uint256, uint8**) {



## CVF-18 INFO

- **Category** Bad naming
- **Source** OracleAggregatorWithSequencerV2.sol

**Recommendation** Events are usually named via nouns, such as "SequencerUptimeFeed".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
7 +event SequencerUptimeFeedUpdated(address indexed
    ↳ sequencerUptimeFeed, uint256 gracePeriodTime);
```

## CVF-19 INFO

- **Category** Bad datatype
- **Source** OracleAggregatorWithSequencerV2.sol

**Recommendation** The type for the "sequencerUptimeFeed" parameter should be "AggregatorV3Interface".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
7 +event SequencerUptimeFeedUpdated(address indexed
    ↳ sequencerUptimeFeed, uint256 gracePeriodTime);
```

## CVF-20 INFO

- **Category** Bad datatype
- **Source** OracleAggregatorWithSequencerV2.sol

**Recommendation** The type for the "\_sequencerUptimeFeed" argument should be "AggregatorV3Interface".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
17 +constructor(address owner, uint256 timeLock, address
    ↳ _sequencerUptimeFeed, uint256 _gracePeriodTime)
    ↳ OracleAggregatorV2(owner, timeLock)
```

```
22 +function setSequencerUptimeFeedAndGracePeriod(address
    ↳ _sequencerUptimeFeed, uint256 _gracePeriodTime) public
    ↳ onlyOwner {
```



## CVF-21 INFO

- **Category** Bad datatype
- **Source** OracleAggregatorWithSequencerV2.sol

**Recommendation** The type for the “asset” argument should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

29    +**function** getPrice(**address** asset) **public view virtual override**  
      ↳ **returns** (**uint256**, **uint8**) {

## CVF-22 INFO

- **Category** Documentation
- **Source** OracleAggregatorWithSequencerV2.sol

**Description** The semantics of the returned values is unclear.

**Recommendation** Give descriptive names to the returned values and/or explain in a documentation comment.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

29    +**function** getPrice(**address** asset) **public view virtual override**  
      ↳ **returns** (**uint256**, **uint8**) {

## CVF-23 INFO

- **Category** Bad naming
- **Source** TermMax4626Factory.sol

**Recommendation** Events are usually named via nouns, such as "TermMax4626Factpry" or "StableERC4626For4626".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
13 +event TermMax4626FactoryInitialized(address aavePool, uint16
    ↵ aaveReferralCode, address stableERC4626For4626Implementation,
    ↵ address stableERC4626ForAaveImplementation, address
    ↵ variableERC4626ForAaveImplementation);
```

```
15 +event StableERC4626For4626Created(address indexed caller, address
    ↵ stableERC4626For4626);
```

```
17 +event StableERC4626ForAaveCreated(address indexed caller, address
    ↵ stableERC4626ForAave);
```

```
19 +event VariableERC4626ForAaveCreated(address indexed caller, address
    ↵ variableERC4626ForAave);
```

## CVF-24 INFO

- **Category** Bad datatype
- **Source** TermMax4626Factory.sol

**Recommendation** The type for this variable should be "StableERC4626for4626".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
21 +address public immutable stableERC4626For4626Implementation;
```

## CVF-25 INFO

- **Category** Bad datatype
- **Source** TermMax4626Factory.sol

**Recommendation** The type for this variable should be "StableERC4626ForAave".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
22 +address public immutable stableERC4626ForAaveImplementation;
```



## CVF-26 INFO

- **Category** Bad datatype
- **Source** TermMax4626Factory.sol

**Recommendation** The type for this variable should be "VariableERC4626ForAave".

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

23    +**address public immutable** variableERC4626ForAaveImplementation;

## CVF-27 INFO

- **Category** Bad datatype
- **Source** TermMax4626Factory.sol

**Recommendation** The type for the "aavePool" argument should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

25    +**constructor(address aavePool, uint16 aaveReferralCode)** {

## CVF-28 INFO

- **Category** Bad datatype
- **Source** TermMax4626Factory.sol

**Recommendation** The type for the "thirdPool" argument should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

32    +**function createStableERC4626For4626(address admin, address**  
    ↪ **thirdPool, StakingBuffer.BufferConfig memory** bufferConfig)  
    ↪ **external returns** (StableERC4626For4626) {

## CVF-29 INFO

- **Category** Bad datatype
- **Source** TermMax4626Factory.sol

**Recommendation** The type for the “underlying” argument should be more specific.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
39 +function createStableERC4626ForAave(address admin, address
  ↪ underlying, StakingBuffer.BufferConfig memory bufferConfig)
  ↪ public returns (StableERC4626ForAave) {
```

```
46 +function createVariableERC4626ForAave(address admin, address
  ↪ underlying, StakingBuffer.BufferConfig memory bufferConfig)
  ↪ public returns (VariableERC4626ForAave) {
```

```
54 +     VariableERC4626ForAave variableInstance =
  ↪ createVariableERC4626ForAave(admin, underlying, bufferConfig);
```

## CVF-30 INFO

- **Category** Bad datatype
- **Source** TermMaxPriceFeedFactoryV2.sol

**Recommendation** The return type should be “TermMaxConstantPriceFeed”.

**Client Comment** *Thanks for your suggestion, we are not considering changing it for now as it is optional.*

```
83 +function createConstantPriceFeed(int256 _constantPrice) external
  ↪ returns (address) {
```



# ABDK Consulting

## About us

Established in 2016, is a leading service provider in the space of blockchain development and audit. It has contributed to numerous blockchain projects, and co-authored some widely known blockchain primitives like Poseidon hash function.

The ABDK Audit Team, led by Mikhail Vladimirov and Dmitry Khovratovich, has conducted over 40 audits of blockchain projects in Solidity, Rust, Circom, C++, JavaScript, and other languages.

## Contact

### ✉ Email

[dmitry@abdkconsulting.com](mailto:dmitry@abdkconsulting.com)

### 🌐 Website

[abdk.consulting](http://abdk.consulting)

### 🐦 Twitter

[twitter.com/ABDKconsulting](https://twitter.com/ABDKconsulting)

### LinkedIn

[linkedin.com/company/abdk-consulting](https://linkedin.com/company/abdk-consulting)