



# **Sky: Lockstake Engine**

## **Security Review**

Cantina Managed review by:

**Christoph Michel**, Lead Security Researcher

**M4rio.eth**, Lead Security Researcher

May 1, 2025

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	About Cantina . . . . .	2
1.2	Disclaimer . . . . .	2
1.3	Risk assessment . . . . .	2
1.3.1	Severity Classification . . . . .	2
<b>2</b>	<b>Security Review Summary</b>	<b>3</b>
<b>3</b>	<b>Findings</b>	<b>4</b>
3.1	Informational . . . . .	4
3.1.1	Missing fee check in the LSE . . . . .	4
3.1.2	Wrong README formatting . . . . .	4
3.1.3	Lockstake migration might bring users closer to liquidation . . . . .	5

# 1 Introduction

## 1.1 About Cantina

Cantina is a security services marketplace that connects top security researchers and solutions with clients. Learn more at [cantina.xyz](https://cantina.xyz)

## 1.2 Disclaimer

Cantina Managed provides a detailed evaluation of the security posture of the code at a particular moment based on the information available at the time of the review. While Cantina Managed endeavors to identify and disclose all potential security issues, it cannot guarantee that every vulnerability will be detected or that the code will be entirely secure against all possible attacks. The assessment is conducted based on the specific commit and version of the code provided. Any subsequent modifications to the code may introduce new vulnerabilities that were absent during the initial review. Therefore, any changes made to the code require a new security review to ensure that the code remains secure. Please be advised that the Cantina Managed security review is not a replacement for continuous security measures such as penetration testing, vulnerability scanning, and regular code reviews.

## 1.3 Risk assessment

Severity	Description
<b>Critical</b>	<i>Must fix as soon as possible (if already deployed).</i>
<b>High</b>	Leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
<b>Medium</b>	Global losses <10% or losses to only a subset of users, but still unacceptable.
<b>Low</b>	Losses will be annoying but bearable. Applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.
<b>Gas Optimization</b>	Suggestions around gas saving practices.
<b>Informational</b>	Suggestions around best practices or readability.

### 1.3.1 Severity Classification

The severity of security issues found during the security review is categorized based on the above table. Critical findings have a high likelihood of being exploited and must be addressed immediately. High findings are almost certain to occur, easy to perform, or not easy but highly incentivized thus must be fixed as soon as possible.

Medium findings are conditionally possible or incentivized but are still relatively likely to occur and should be addressed. Low findings a rare combination of circumstances to exploit, or offer little to no incentive to exploit but are recommended to be addressed.

Lastly, some findings might represent objective improvements that should be addressed but do not impact the project's overall security (Gas and Informational findings).

## 2 Security Review Summary

Sky Protocol is a decentralised protocol developed around the USDS stablecoin.

From Apr 18th to Apr 23rd the Cantina team conducted a review of [lockstake's src](#) and [deploy](#) directory contents on commit hash [ccc1c16b](#). The team identified a total of **3** issues in the following risk categories:

- Critical Risk: 0
- High Risk: 0
- Medium Risk: 0
- Low Risk: 0
- Gas Optimizations: 0
- Informational: 3

The Cantina Managed team reviewed Sky's lockstake holistically on commit hash [9cb25125](#) and concluded that all issues were addressed and no new vulnerabilities were identified.

## 3 Findings

### 3.1 Informational

#### 3.1.1 Missing `fee` check in the LSE

**Severity:** Informational

**Context:** [LockstakeEngine.sol#L131](#)

**Description:** The `LockstakeEngine` sets the `fee` used in `free` operations as an immutable field in the constructor. However, this value is not validated against `WAD`. In the previous version of the LSE, the `fee` was configurable via `file` and included the following check:

```
if (what == "fee") {  
    require(data < WAD, "LockstakeEngine/fee-equal-or-greater-wad");  
    fee = data;  
}
```

**Recommendation:** Consider adding the same validation check in the constructor of the new LSE.

**Sky:** Fixed in commit [cfb692c7](#).

**Cantina Managed:** Verified.

#### 3.1.2 Wrong README formatting

**Severity:** Informational

**Context:** [README.md](#)

**Description:** In the new README, the `Configurable Parameters` section of the `StakingRewards` is after the `Migrator` which is confusing

## 7. LockstakeMigrator

A contract which has the purpose to move `urn`s from a deprecated Lockstake version to a newer one, without having to pay the `exit` fee which would be required if the user would want to do this manually via the regular functions. This contract uses the `LockstakeEngine.freeNoFee` function ensuring the collateral will still remain locked in a `LockstakeEngine`. The migrator requires to be added to the `wards` mapping of the old `LockstakeEngine` and to the `wards` mapping of the `Vat`.

There are two paths that the user could take when calling the `migrate` function for the desired `urn`:

- If the `urn` doesn't have any debt. This is the simplest path where the collateral is just `free`d from the old engine and `lock`ed in the new one.
- If the `urn` has debt. This path uses the `DssFlash` module to `wipe` the debt in the old `urn` to be able to move the collateral. After doing so, the debt will be `draw`n in the new `urn` and the funds will be returned to the `DssFlash` module (all happens atomically).

The first path requires the migrator to be `hope`d in the old Engine for the `urn` being migrated. An authed address needs to call this `hope` function previously. It is also required that the caller of `migrate` be an authed address in the `urn` being migrated and in the recipient one. For the second path, apart from the same requirements of the simplest one, it is also necessary that an authed address in the `urn` that is receiving the position in the new Lockstake has `hope`d the migrator.

Note: The caller authed requirement for the recipient `urn` in the first path is just an extra safety measure to avoid migrating collateral to an undesired `urn`. However for the second path it is indeed mandatory as migrating debt increases the debt of the recipient `urn`.

Note 2: Even though migrating debt manually outside the migrator is not supported, it is not guaranteed that a `migrate` call would not revert. It depends on governance parameters such as liquidation ratios and dust, and the system state such as whether a position is under liquidation or should use the `reserveHatch` mechanism. Governance is assumed to configure the parameters in a user-friendly way. The user is of course assumed to be aware of the parameters (for example if after migrating they become closer to liquidation).

Note 3: It is assumed that the debt of the old engine does not exceed the amount filed in `onVatDaiFlashLoan` prior to the migrator being enabled, and that governance do not change the old ilk line from 0 throughout the process.

Note 4: It is assumed that after a certain period the migrator's permission over the `Vat` will be removed, and the debt ceiling will be managed as usual using the autoline.

Note 5: Migration won't transfer the `VoteDelegate` nor the farm selected in the old `urn` to the destination one. This needs to be manually done by an `urn` authed user directly in the new Engine (before or after the migration).

Note 6: Migrator assumes `MkrSky` is configured without a penalty. So as soon as, the penalty is set above 0, the migrator will generally stop working. It also expects MKR to SKY conversions are not blocked.

### Configurable Parameters:

- `rewardsDistribution` - The address which is allowed to start a rewards distribution. Will be set to the splitter.
- `rewardsDuration` - The amount of seconds each distribution should take.

**Recommendation:** Consider moving this in the right section.

**Sky:** Fixed in commit [9cb25125](#).

**Cantina Managed:** Verified.

### 3.1.3 Lockstake migration might bring users closer to liquidation

**Severity:** Informational

**Context:** [LockstakeMigrator.sol#L137](#)

**Description:** The `LockstakeEngine` (LSE) migration converts the old LSE CDP to the new LSE CDP by converting MKR to SKY. The new LSE uses a different oracle (SKY) which can be configured with a different `mat` (liquidation threshold) and can return a price not directly equivalent to the price of the old MKR oracle.

Users can end up closer to liquidation after migration and might get liquidated in the next block on a small price swing.

**Recommendation:** Users need to be aware of this potential risk before migrating their positions.

**Sky:** Added the following section to the README on commit [fe854c54](#):

It depends on governance parameters such as liquidation ratios and dust, and the system state such as whether a position is under liquidation or should use the reserveHatch mechanism. Governance is assumed to configure the parameters in a user-friendly way. The user is of course assumed to be aware of the parameters (for example if after migrating they become closer to liquidation).

**Cantina Managed:** Fix verified.