



Sky: PAU NFAT Facet

Security Review

Cantina Managed review by:

Christoph Michel, Lead Security Researcher

M4rio.eth, Lead Security Researcher

July 6, 2026

Contents

1	Introduction	2
1.1	About Cantina	2
1.2	Disclaimer	2
1.3	Risk assessment	2
1.3.1	Severity Classification	2
2	Security Review Summary	3
3	Findings	4
3.1	Low Risk	4
3.1.1	NFAT position state can desync when repayments bypass facet	4
3.1.2	Some off-chain NFAT contracts may not be fully repayable	4
3.2	Informational	5
3.2.1	Halo issue rate-limit key omits the facility gem	5
3.2.2	PAU cannot transfer NFTs	5
3.2.3	Minor issues, Typos & Documentation	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

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 level	Impact: High	Impact: Medium	Impact: Low
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

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 are 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 Jun 22nd to Jun 25th the Cantina team conducted a review of [diamond-pau](#) on commit hash [4812e320 \(v1.14.0-beta.0\)](#). The team identified a total of **5** issues:

Issues Found

Severity	Count	Fixed	Acknowledged
Critical Risk	0	0	0
High Risk	0	0	0
Medium Risk	0	0	0
Low Risk	2	0	2
Gas Optimizations	0	0	0
Informational	3	0	3
Total	5	0	5

The Cantina Managed team reviewed Sky's [diamond-pau](#) holistically on commit hash [cbf71b2a \(v1.14.0\)](#) and concluded that all findings were addressed and no new vulnerabilities were identified.

3 Findings

3.1 Low Risk

3.1.1 NFAT position state can desync when repayments bypass facet

Severity: Low Risk.

Context: `NFATHaloFacet.sol#L134`.

Description: Repayments can be made directly through `NFATFacility.repay` without going through `NFATHaloFacet`. Because the facet tracks its own `position` state, direct repayments can cause the stored position data to become stale or inconsistent with the facility. The NFAT token can be over-repaid, which is exactly what the position accounting is trying to avoid.

The impact is lowered as there is no economic incentive for a third-party that is not the Halo PAU to repay - and the Halo proxy itself should only be configured to be repayable through its controller's facet.

Recommendation: There is currently no way to avoid this without changing the `NFATFacility` contract. Ensure the Halo proxy itself is configured to only be able to perform repayments through this `NFATHaloFacet.repay*` code path.

Sky: Acknowledged. The PAU is responsible for itself and cannot prevent others from interacting with repay as the function is open.

The halo proxy will have to go through the facet (or a future facet as well) in order to execute repayments to correctly checkpoint the state.

Cantina Managed: Acknowledged.

3.1.2 Some off-chain NFAT contracts may not be fully repayable

Severity: Low Risk.

Context: `NFATHaloFacet.sol#L164`.

Description: The current repayment model splits position state into principal and interest, and uses the computed principal plus interest as the max repayable amount for an NFAT token. Because allowed interest is computed from the current *outstanding principal*, this assumes the off-chain contract charges linear interest only on outstanding principal, and that repaying principal early reduces future interest accrual.

Some valid off-chain contracts may instead charge fixed interest or fees over the full term regardless of principal repayments. In those cases, repaying principal early can reduce `outstandingPrincipal` to zero, preventing further interest accrual on-chain and making the full off-chain obligation impossible to repay through HALO.

Example: To demonstrate the issue, assume a term sheet provides \$1M for 2 years at 100% annual fixed interest, so the expected total collectible amount at maturity is \$3M (regardless of when the principal is repaid). The contract would model it as follows:

- Initial principal is \$1M with a 100% annual growth rate.
- After 1 year, `outstandingPrincipal` is \$1M and `outstandingInterest` is \$1M.
- When \$1M of principal is repaid, `outstandingPrincipal` becomes zero.
- No interest accrues during year 2 because interest is based on current outstanding principal.
- At the end of year 2, the remaining second year \$1M owed under the term sheet cannot be repaid through HALO.

Recommendation: Avoid making interest repayment assumptions about the off-chain NFAT contract. As the stated intended goal of the current per-position accounting is to provide per-position repayment rate-limits in case of a malicious allocator, consider alternative admin-controlled per-NFAT-token rate limits that provide an upperbound on the repayment.

Sky: Acknowledged. This can be considered the V1 for NFAT Facets and the structure closely maps to many investments that primes have made. Therefore, we will keep it as is and accept the constraint that

the facet in its current state may not be suitable for every type of investment. As NFATs are explored more deeply, it's likely that related facets will be introduced.

Cantina Managed: Acknowledged.

3.2 Informational

3.2.1 Halo issue rate-limit key omits the facility gem

Severity: Informational.

Context: `NFATHaloFacet#L103-L124` .

Description: Halo `issue` consumes gem amount, but the issue rate-limit key does not include the `gem` . The docs explain why the key should not be per `tokenId` , but they do not justify omitting `gem` ; repayment keys include `gem` specifically so a configured limit is bound to the token units being spent.

Recommendation: Include `gem` in Halo `issue` key derivation, for example `getIssueRateLimitKey(addressfacility,addressgem,addresssto)` .

Sky: Acknowledged. Throughout the codebase, we don't key on `gem` if we're not directly interacting with the underlying `gem` . In order to keep consistency across the codebase we will acknowledge this.

Cantina Managed: Acknowledged.

3.2.2 PAU cannot transfer NFTs

Severity: Informational.

Context: `NFATPrimeFacet.sol#L1`.

Description: PAUs cannot currently transfer NFTs, including NFATs, because no ERC-721 transfer function is exposed. This is relevant to NFATs as the owner of the NFT can `collect` the repayments.

Recommendation: Decide whether PAUs should support NFT transfers. If so, add ERC-721 `transferFrom` support to `TransferAssetFacet` or a dedicated NFT transfer facet.

Sky: Yes, this is desired for now. We had a facet but removed it as it needed more discussion. If we need to move NFTs we can do it through a spell for now.

Cantina Managed: Acknowledged.

3.2.3 Minor issues, Typos & Documentation

Severity: Informational.

Context: (See each case below).

Description:

1. `docs/NFAT_INTEGRATION.md#L102-L115` , `NFATDeploy.sol#L31-L34` , and `NFATInit.sol#L51-L58` : `annualGrowthRate` docs should say how to think about sUSDS appreciation. If the facility gem is `SUSDS` , operators need to know how the configurable annual growth rate should be set.
2. `docs/NFAT_INTEGRATION.md#L7-L10` and `NFATHaloFacet.sol#L37-L41` : Docs say one PAU maps to one facility, but `NFATHaloFacet` is not immutable-bound to one facility. It is facility-keyed, so the one-facility rule is enforced by governance config and rate limits, not by the facet itself. The docs should say that directly.
3. `docs/NFAT_INTEGRATION.md#L29-L39` , `NFATPrimeFacet.sol#L51-L72` , and `NFATFacility.sol#L191-L198` : If the facility uses an `identityNetwork` , the ALMProxy also needs to be a member in any flow where it is expected to receive or own NFATs. The docs mention `bud` and `recipient` , but not this membership requirement.

4. `INFATHaloFacet.sol#L153-L164` and `NFATHaloFacet.sol#L186-L195` : `getFacilityState` Nat-Spec says it returns the current live index including time accrual. It actually returns the stored checkpointed `interestIndex` and `lastUpdated` . The wording should match the implementation.
5. `NFATHaloFacet.sol#L118`: The facility recipient is documented as expected to be the proxy, but this invariant is not enforced onchain. If the Halo PAU or facility is misconfigured, funds may be directed to an unexpected recipient. Add a validation that requires `facility.recipient()==proxy` .
6. `NFATPrimeFacet.sol#L93`: After determining the actual `received` amount, consider calling `_tryIncreaseRateLimit(getSubscribeRateLimitKey(facility,gem),received)` so subscribe capacity is restored consistently with the withdrawn amount (as `withdraw` is effectively "cancel subscribe").
7. `NFATHaloFacet.sol#L235`: `outstandingInterest+=outstandingPrincipal*deltaIndex/1e18` can round to zero when `gem` has low decimals or `outstandingPrincipal` is small. In practice, only an authorized allocator/admin path can force position checkpointing. The grieving case would be a malicious allocator repeatedly making tiny nonzero repayments to checkpoint the position and lose fractional interest. However, this seems unlikely/low-impact for expected assets and position sizes.

Recommendation: Consider addressing the issues mentioned above.

Sky: Adressed 1, 2 and 4 in [PR 208](#). Acknowledged the rest.

Cantina Managed: Acknowledged.