Quantstamp

Fragmetric Liquid Restaking Program - 2

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	Solana Restaking Protocol
Timeline	2025-08-18 through 2025-09-21
Language	Rust
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Restaking Program Changelog User Manual Fragmetric Docs Fragmetric Docs - Rewards Distribution
Source Code	• https://github.com/fragmetric-labs/fragmetric-contracts #7fb983e
Auditors	• Cameron Biniamow Auditing Engineer • Mostafa Yassin Auditing Engineer • Yamen Merhi Auditing Engineer

Documentation quality	High
Test quality	Medium
Total Findings	3 Fixed: 2 Acknowledged: 1
High severity findings	0
Medium severity findings	0
Low severity findings	1 Fixed: 1
Undetermined severity findings	0
Informational findings	2 Fixed: 1 Acknowledged: 1

Summary of Findings

Fragmetric is a decentralized liquid restaking program deployed on the Solana blockchain, enabling users to deposit SOL or supported tokens into the protocol and receive minted receipt tokens in return. Once deposited into the Fragmetric fund account, user funds are staked into various restaking vaults to generate rewards.

Quantstamp was tasked with a time-boxed review of Fragmetric's third iteration of their restaking contracts. Specifically, the fund, normalization, pricing, staking, restaking, reward, and swap modules were reviewed to identify deviations from the project specification, potential vulnerabilities, and proper integrations with external protocols. All external protocols and the Solv program wrapper were considered out of scope for this review. Due to the constrained time available for this review and the large size of the codebase, the audit team was unable to perform a comprehensive audit of the codebase. Therefore, uncaught bugs or vulnerabilities may remain in the code.

Fragmetric's codebase underwent substantial expansion and feature development. The protocol introduced significant new functionality, including enhanced fund configuration options (such as disabling receipt token transfers, fund operations, and token removal), pegged supported token capabilities, comprehensive restaking vault reward management with harvest thresholds and commission rates, and a 1:1 receipt token wrapping/unwrapping system. The team implemented a new token swap strategy framework with validation, expanded the reward system to include admin-created user accounts with delegation support and revenue account management, and integrated multiple new pricing sources, including Pegged Token, Solv BTC Vault, Sanctum Multi Validator Stake Pool, and Virtual Vault concepts. Performance optimizations were made through pricing value caching and account data access via offset rather than full deserialization, while new third-party integrations were added for Solv BTC Vault operations, Sanctum Multi Validator services, and Orca DEX liquidity pools.

Following the review, the audit team noted that the code quality remained high, largely adhering to best practices, and no significant security vulnerabilities were identified. However, the areas of centralization require clear documentation and user awareness. Users must understand and accept these trust assumptions when interacting with the protocol.

Fix Review Update: In the recent code updates, FRAG-1 has been fixed by enhancing the user reward pool synchronization process to mitigate potential denial-of-service risks. FRAG-2 has also been successfully fixed, with the client implementing a more secure reward account system to prevent unauthorized account creation. FRAG-3 has been acknowledged, with the client planning to transition fund manager authority to a multi-signature committee as part of their governance roadmap. Additionally, suggestions S1, S3, and S4 have been addressed and resolved, while S2 has been acknowledged with clarification on naming conventions.

After the initial review and before the fix commits, the Fragmetric team added code changes that are considered out of scope and were not reviewed by the audit team. Specifically, the out-of-scope commits range from `6cdcfdc77da1819d813ca83f9f80a5a0a4cb71cc` to `e85268f5c10077a96a9de019cc327f5bd9bc3db5`. Following the initial review, only the fix commits listed in the finding updates were reviewed to ensure the fix corrected the issue.

ID	DESCRIPTION	SEVERITY	STATUS
FRAG-1	**Potenial Risk for DoS For Dormant Users**	Low	Fixed
FRAG-2	**Admin Sets Fund Manager as Delegate on User Reward Account Creation**	Informational	Fixed

ID	DESCRIPTION	SEVERITY	STATUS
FRAG-3	**Fund Account State Can Be Instantly Updated by Fund Manager**	Informational	Acknowledged

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

> **Disclaimer**
> Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
    1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
    1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices,

recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

Files Included

Repo: `https://github.com/fragmetric-labs/fragmetric-contracts`

Included Paths: `programs/restaking`

Extensions: `rs`

Operational Considerations

1. The Fragmetric protocol relies on external protocols such as SPL Stake Pool, Marinade Stake Pool, Jito Restaking Vault, Orca Liquidity Pools, Sanctum Single Validator SPL Stake Pool, and Solv BTC Vault for restaking and pricing of assets. Exploits or configuration updates of these external protocols could result in unexpected behavior of the Fragmetric protocol.

Key Actors And Their Capabilities

There are five different actors in the protocol

The Admin can execute the instructions:

- `admin_initialize_fund_account()`
- `admin_update_fund_account_if_needed()`
- `admin_set_address_lookup_table_account()`
- `admin_initialize_normalized_token_pool_account()`
- `admin_update_normalized_token_pool_account_if_needed()`
- `admin_initialize_extra_account_meta_list()`
- `admin_update_extra_account_meta_list_if_needed()`
- `admin_initialize_reward_account()`
- `admin_update_reward_account_if_needed()`
- `admin_create_user_reward_account_idempotent()`

The Fund Manager can execute the instructions:

- `fund_manager_update_fund_strategy()`
- `fund_manager_update_sol_strategy()`
- `fund_manager_update_supported_token_strategy()`
- `fund_manager_update_restaking_vault_strategy()`
- `fund_manager_update_restaking_vault_delegation_strategy()`
- `fund_manager_update_restaking_vault_reward_token_harvest_threshold()`
- `fund_manager_add_restaking_vault_compounding_reward_token()`
- `fund_manager_remove_restaking_vault_compounding_reward_token()`
- `fund_manager_add_restaking_vault_distributing_reward_token()`
- `fund_manager_remove_restaking_vault_distributing_reward_token()`

- `fund_manager_add_token_swap_strategy()`
- `fund_manager_remove_token_swap_strategy()`
- `fund_manager_initialize_fund_normalized_token()`
- `fund_manager_initialize_fund_wrapped_token()`
- `fund_manager_reset_fund_wrap_account_reward_account_delegate()`
- `fund_manager_add_wrapped_token_holder()`
- `fund_manager_remove_wrapped_token_holder()`
- `fund_manager_reset_wrapped_token_holder_reward_account_delegate()`
- `fund_manager_initialize_fund_restaking_vault()`
- `fund_manager_initialize_fund_restaking_vault_delegation()`
- `fund_manager_add_supported_token()`
- `fund_manager_remove_supported_token()`
- `fund_manager_add_normalized_token_pool_supported_token()`
- `fund_manager_remove_normalized_token_pool_supported_token()`
- `fund_manager_add_reward()`
- `fund_manager_update_reward()`
- `fund_manager_settle_reward()`

The Operator can execute the instructions:

- `operator_log_message()`
- `operator_run_fund_command()`
- `operator_update_fund_prices()`
- `operator_donate_sol_to_fund()`
- `operator_donate_supported_token_to_fund()`
- `operator_update_reward_pools()`
- `operator_claim_remaining_reward()`
- `operator_update_normalized_token_pool_prices()`

The Slasher can execute the instructions:

- `slasher_initialize_normalized_token_withdrawal_account()`
- `slasher_withdraw_normalized_token()`

The User can execute the instructions:

- `user_create_fund_account_idempotent()`
- `user_deposit_sol()`
- `user_request_withdrawal()`
- `user_cancel_withdrawal_request()`
- `user_withdraw_sol()`
- `user_deposit_supported_token()`
- `user_withdraw_supported_token()`
- `user_deposit_vault_receipt_token()`
- `user_wrap_receipt_token()`
- `user_wrap_receipt_token_if_needed()`
- `user_unwrap_receipt_token()`
- `user_create_reward_account_idempotent()`
- `user_update_reward_pools()`
- `user_claim_reward()`
- `user_delegate_reward_account()`

Findings

FRAG-1 Potenial Risk for DoS For Dormant Users

Low Fixed

Update

The client fixed the issue in commit `6abadee9a9041ddc5f55524545910552e9ba202a` and provided the following explanation:

> We recently acknowledged the potential DoS risk for dormant users (e.g., inactive for more than two years). To mitigate this, we modified the process_update_user_reward_pools instruction to optionally accept a num_blocks_to_settle parameter as a per-call limit. With this change, dormant user reward pools can be safely synchronized without risk of DoS. A fix has already been committed.

File(s) affected: `programs/restaking/src/modules/reward/reward_account.rs`

Description: The reward update logic for a given user loops over all reward settlements, and for each reward settlement, it loops over all reward blocks to calculate the reward per block for the user. There can be multiple reward settlements with multiple reward blocks inside of them.

For dormant or highly inactive users, the nested loops can cause the transaction to consume all computation units and revert, which will prevent the user from interacting with the protocol.

Recommendation: Consider adding a paramater that specifies the a bound for how many iterations of looping at a given update. Then, for the next update, continue from the last updated block.

FRAG-2 Admin Sets Fund Manager as Delegate on User Reward Account Creation

Informational Fixed

Update

The client fixed the issue in commit `350be95d216ae3cc786e86de8c80ef1c31c54786` and provided the following explanation:

> This behavior was introduced when integrating the wrap/unwrap system with the existing reward mechanism. A “fund wrap” account is required to hold wrapped receipt tokens, and for DeFi integrations it is also necessary to track contributions (amount and holding period) of lock-in wrapped token accounts. To address this, we added functionality for the admin to create a user reward account for such non-user accounts and set the delegate. As noted, we have restricted this capability so that it only applies for the above system purposes and cannot be used to create arbitrary user reward accounts. A fix has been committed.

File(s) affected: `programs/restaking/src/lib.rs`

Description: The `admin_create_user_reward_account_idempotent` instruction provides a privileged admin with the ability to create a `UserRewardAccount` on behalf of any user. When this instruction is called, it hardcodes the `FUND_MANAGER_PUBKEY` as a delegate for the new account. This is a significant trust assumption in the protocol, as it grants the Fund Manager authority over the user's reward account without the user's direct signature for the delegation. This functionality is likely intended for administrative or onboarding purposes, but it implies that users must trust the Fund Manager not to misuse these delegated privileges (e.g., claiming rewards on their behalf).

Recommendation: Consider if this is intended, else consider having the delegate set to `None` instead.

FRAG-3
Fund Account State Can Be Instantly Updated by Fund Manager

Informational | Acknowledged

> **Update**
> The client acknowledged the issue and provided the following explanation:
>
> ```
> We acknowledge this behavior. The protocol will transition fund manager
> authority to a multi-sig committee once open-sourcing and governance
> activation are in place. While the protocol is nearly complete, at the current
> stage our priority is the ability to respond quickly to emergencies. This
> aspect will be clearly guided in the protocol roadmap.
> ```

File(s) affected: `programs/restaking/src/modules/fund/fund_configuration_service.rs`

Description: A fund manager can unilaterally and instantly update the state of a `FundAccount`. While this is likely for operational purposes, it may not be immediately obvious to users that this level of control is granted. This represents a significant trust assumption in the fund manager's operational security and integrity.

Recommendation: This behavior should be clearly documented in user-facing materials to ensure transparency. If this level of centralized control is not the long-term design, consider implementing a timelock or a multi-sig mechanism for critical state changes to decentralize trust.

Auditor Suggestions

S1 Misspellings

Fixed

> **Update**
> The client fixed the suggestion in commit `310ed2e9d848fa07ba1c74eac02fcf02feb4a444`.

Description: The following misspellings were found throughout the codebase:

1. `programs/restaking/src/errors.rs:204:18` - (**unauhorized**) fix: (**unauthorized**)
2. `programs/restaking/src/events/operator_donated_to_fund.rs:10:9` - (**offsetted**) fix: (**offset**)
3. `programs/restaking/src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs:37:9` - (**offsetted**) fix: (**offset**)

4. `programs/restaking/src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs:38:9` - (**offsetted**) fix: (**offset**)
5. `programs/restaking/src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs:462:29` - (**offsetted**) fix: (**offset**)
6. `programs/restaking/src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs:463:29` - (**offsetted**) fix: (**offset**)
7. `programs/restaking/src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs:555:21` - (**offsetted**) fix: (**offset**)
8. `programs/restaking/src/modules/fund/commands/cmd5_claim_unstaked_sol.rs:83:9` - (**offsetted**) fix: (**offset**)
9. `programs/restaking/src/modules/fund/commands/cmd5_claim_unstaked_sol.rs:85:9` - (**offsetted**) fix: (**offset**)
10. `programs/restaking/src/modules/fund/commands/cmd5_claim_unstaked_sol.rs:519:17` - (**offsetted**) fix: (**offset**)
11. `programs/restaking/src/modules/fund/commands/cmd5_claim_unstaked_sol.rs:520:17` - (**offsetted**) fix: (**offset**)
12. `programs/restaking/src/modules/fund/commands/cmd5_claim_unstaked_sol.rs:553:21` - (**offsetted**) fix: (**offset**)
13. `programs/restaking/src/modules/fund/commands/cmd6_process_withdrawal_batch.rs:47:9` - (**offsetted**) fix: (**offset**)
14. `programs/restaking/src/modules/fund/commands/cmd6_process_withdrawal_batch.rs:426:21` - (**offsetted**) fix: (**offset**)
15. `programs/restaking/src/modules/fund/commands/cmd6_process_withdrawal_batch.rs:441:25` - (**offsetted**) fix: (**offset**)
16. `programs/restaking/src/modules/fund/commands/cmd6_process_withdrawal_batch.rs:484:25` - (**offsetted**) fix: (**offset**)
17. `programs/restaking/src/modules/fund/commands/cmd6_process_withdrawal_batch.rs:498:25` - (**offsetted**) fix: (**offset**)
18. `programs/restaking/src/modules/fund/fund_account_asset_state.rs:149:36` - (**offsetted**) fix: (**offset**)
19. `programs/restaking/src/modules/fund/fund_account.rs:874:36` - (**offsetted**) fix: (**offset**)
20. `programs/restaking/src/modules/fund/fund_service.rs:553:154` - (**offsetted**) fix: (**offset**)
21. `programs/restaking/src/modules/fund/fund_service.rs:763:17` - (**offsetted**) fix: (**offset**)
22. `programs/restaking/src/modules/fund/fund_service.rs:879:44` - (**offsetted**) fix: (**offset**)
23. `programs/restaking/src/modules/fund/fund_service.rs:879:68` - (**offsetted**) fix: (**offset**)
24. `programs/restaking/src/modules/fund/fund_service.rs:896:44` - (**offsetted**) fix: (**offset**)
25. `programs/restaking/src/modules/pricing/pricing_service.rs:25:20` - (**mirco**) fix: (**micro**)
26. `programs/restaking/src/modules/pricing/pricing_service.rs:40:28` - (**mirco**) fix: (**micro**)
27. `programs/restaking/src/modules/pricing/pricing_service.rs:95:33` - (**mirco**) fix: (**micro**)
28. `programs/restaking/src/modules/pricing/pricing_service.rs:104:33` - (**mirco**) fix: (**micro**)
29. `programs/restaking/src/modules/pricing/pricing_service.rs:227:27` - (**mirco**) fix: (**micro**)
30. `programs/restaking/src/modules/pricing/pricing_service.rs:809:30` - (**vaule**) fix: (**value**)
31. `programs/restaking/src/modules/pricing/pricing_service.rs:811:72` - (**vaule**) fix: (**value**)
32. `programs/restaking/src/modules/pricing/pricing_service.rs:814:51` - (**vaule**) fix: (**value**)
33. `programs/restaking/src/modules/restaking/jito_restaking_vault_service.rs:729:104` - (**withdrawalble**) fix: (**withdrawable**)
34. `programs/restaking/src/modules/restaking/solv_btc_vault_service.rs:298:33` - (**incompleted**) fix: (**incomplete**)
35. `programs/restaking/src/modules/restaking/solv_btc_vault_service.rs:365:38` - (**incompleted**) fix: (**incomplete**)
36. `programs/restaking/src/modules/reward/token_allocated_amount.rs:262:5` - (**Auxillary**) fix: (**Auxiliary**)

37. `programs/restaking/src/modules/staking/marinade_stake_pool_service.rs:108:33` - (**wirtable**) fix: (**writable**)
38. `programs/restaking/src/modules/staking/spl_stake_pool_service.rs:265:46` - (**referer**) fix: (**referrer**)

Recommendation: Fix misspellings identified throughout the codebase.

S2 Inconsistency in Naming Conventions

Acknowledged

> **Update**
> The client acknowledged the suggestion and provided the following explanation:
>
> ```
> Our convention distinguishes between the two cases: _padding is strictly
> technical padding that will never be used in the future, whereas _reserved
> designates space that may be used for new features later. This differentiation
> is intentional, though we acknowledge the naming may appear inconsistent
> without this context.
> ```

Description: The program has inconsistent naming conventions for reserved space in account structs. Some fields use `_padding`, others use `_reserved`, and the presence of an underscore varies, despite all serving the same purpose.

Recommendation: Standardize naming conventions for reserved space across all structs. Use a single, consistent prefix (e.g., `_reserved`) to improve readability and maintainability.

S3 Division-by-Zero Discrepancy in Proportional Math (u64 vs u128)

Fixed

> **Update**
> The client fixed the suggestion in commit `80f8b9cb1ea61da49378591b9f966d004c91e258`.

File(s) affected: `programs/restaking/src/utils.rs`

Description: `get_proportional_amount_u64` performs a raw division without guarding against a zero denominator, which can panic at runtime. In contrast, `get_proportional_amount_u128` uses `checked_div` and returns a handled error. This inconsistency means the same logical operation is safe in the u128 path but crash-prone in the u64 path.

Recommendation: Normalize behavior across both helpers: add an explicit non-zero-denominator check to the u64 helper, mirror the u128 checked arithmetic semantics, and return a defined error instead of panicking.

S4

Presence of TODO Comments in Slasher Withdrawal Logic

Fixed

Update

The client fixed the suggestion in commit `2bc618c4c0c404424a93abfd315aa05bf29dde9c` and provided the following explanation:

```
Tests added, TODOs removed, slasher path verified.
```

File(s) affected: `programs/restaking/src/lib.rs`

Description: The functions `slasher_initialize_normalized_token_withdrawal_account` and `slasher_withdraw_normalized_token` in `program/src/lib.rs` contain `// TODO: untested` comments, indicating incomplete implementation or missing test coverage. Leaving TODOs in production code may signal unfinished logic and create uncertainty about correctness.

Recommendation: Address and resolve the TODOs by implementing the missing logic or adding the necessary tests, and remove the comments before production deployment.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not pose an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Files

Repo: `https://github.com/fragmetric-labs/fragmetric-contracts`

- `cdb...3e7 ./programs/restaking/src/constants/devnet.rs`
- `602...4d3 ./programs/restaking/src/constants/local.rs`
- `220...15e ./programs/restaking/src/constants/mainnet.rs`
- `9b9...a77 ./programs/restaking/src/constants/mod.rs`
- `bcf...c76 ./programs/restaking/src/errors.rs`
- `9cc...f93 ./programs/restaking/src/events/fund_manager_updated_fund.rs`
- `bf7...09c ./programs/restaking/src/events/fund_manager_updated_reward_pool.rs`
- `f2a...2eb ./programs/restaking/src/events/mod.rs`
- `f7f...d59 ./programs/restaking/src/events/operator_donated_to_fund.rs`
- `95b...83f ./programs/restaking/src/events/operator_ran_fund_command.rs`
- `597...a70 ./programs/restaking/src/events/operator_updated_fund_prices.rs`
- `8f3...26c ./programs/restaking/src/events/operator_updated_normalized_token_pool_prices.rs`
- `f97...1bc ./programs/restaking/src/events/operator_updated_reward_pools.rs`
- `394...ddd ./programs/restaking/src/events/user_canceled_withdrawal_request_from_fund.rs`
- `48e...558 ./programs/restaking/src/events/user_claimed_reward.rs`
- `eac...779 ./programs/restaking/src/events/user_created_or_updated_fund_account.rs`
- `73c...41a ./programs/restaking/src/events/user_created_or_updated_reward_account.rs`
- `8eb...017 ./programs/restaking/src/events/user_delegated_reward_account.rs`
- `0b2...e32 ./programs/restaking/src/events/user_deposited_to_fund.rs`
- `4ae...5b7 ./programs/restaking/src/events/user_deposited_to_vault.rs`
- `3d9...837 ./programs/restaking/src/events/user_requested_withdrawal_from_fund.rs`
- `fc1...b8d ./programs/restaking/src/events/user_transferred_receipt_token.rs`
- `d7d...a89 ./programs/restaking/src/events/user_unwrapped_receipt_token.rs`
- `06b...d91 ./programs/restaking/src/events/user_updated_reward_pool.rs`
- `6dd...1a4 ./programs/restaking/src/events/user_withdrew_from_fund.rs`
- `0a9...1c2 ./programs/restaking/src/events/user_wrapped_receipt_token.rs`
- `53c...7ee ./programs/restaking/src/instructions/admin_fund_context.rs`
- `9d7...bda ./programs/restaking/src/instructions/admin_normalized_token_pool_context.rs`
- `e7f...f96 ./programs/restaking/src/instructions/admin_receipt_token_mint_context.rs`

- `5de...030 ./programs/restaking/src/instructions/admin_reward_context.rs`
- `d08...e30 ./programs/restaking/src/instructions/admin_user_reward_context.rs`
- `ea7...348 ./programs/restaking/src/instructions/fund_manager_fund_context.rs`
- `5c9...507 ./programs/restaking/src/instructions/fund_manager_fund_normalized_token_context.rs`
- `0ed...955 ./programs/restaking/src/instructions/fund_manager_fund_restaking_vault_context.rs`
- `a82...c7f ./programs/restaking/src/instructions/fund_manager_fund_supported_token_context.rs`
- `239...fdf ./programs/restaking/src/instructions/fund_manager_fund_token_swap_strategy_context.rs`
- `b49...622 ./programs/restaking/src/instructions/fund_manager_fund_wrapped_token_context.rs`
- `5b9...d0d ./programs/restaking/src/instructions/fund_manager_normalized_token_pool_supported_token_context.rs`
- `fe1...490 ./programs/restaking/src/instructions/fund_manager_reward_context.rs`
- `af8...ca8 ./programs/restaking/src/instructions/mod.rs`
- `706...952 ./programs/restaking/src/instructions/operator_empty_context.rs`
- `b10...adb ./programs/restaking/src/instructions/operator_fund_context.rs`
- `015...cde ./programs/restaking/src/instructions/operator_normalized_token_pool_context.rs`
- `578...83c ./programs/restaking/src/instructions/operator_reward_context.rs`
- `b70...97f ./programs/restaking/src/instructions/slasher_normalized_token_context.rs`
- `add...2bb ./programs/restaking/src/instructions/user_fund_context.rs`
- `f25...f66 ./programs/restaking/src/instructions/user_fund_supported_token_context.rs`
- `c82...2cc ./programs/restaking/src/instructions/user_fund_vault_receipt_token_context.rs`
- `e56...214 ./programs/restaking/src/instructions/user_fund_wrapped_token_context.rs`
- `0ce...1f2 ./programs/restaking/src/instructions/user_receipt_token_transfer_context.rs`
- `e77...cd8 ./programs/restaking/src/instructions/user_reward_context.rs`
- `08e...cf7 ./programs/restaking/src/lib.rs`
- `588...177 ./programs/restaking/src/modules/ed25519/mod.rs`
- `853...79e ./programs/restaking/src/modules/ed25519/signature_verification_service.rs`
- `93e...407 ./programs/restaking/src/modules/fund/commands/cmd10_harvest_restaking_yield.rs`
- `bd0...ddb ./programs/restaking/src/modules/fund/commands/cmd11_stake_sol.rs`
- `a53...90d ./programs/restaking/src/modules/fund/commands/cmd12_normalize_st.rs`

- 6e3...e24 ./programs/restaking/src/modules/fund/commands/cmd13_restake_vst.rs
- e41...b31 ./programs/restaking/src/modules/fund/commands/cmd14_delegate_vst.rs
- 9ca...aa1 ./programs/restaking/src/modules/fund/commands/cmd1_initialize.rs
- 674...fe8 ./programs/restaking/src/modules/fund/commands/cmd2_enqueue_withdrawal_batch.rs
- eff...243 ./programs/restaking/src/modules/fund/commands/cmd3_claim_unrestaked_vst.rs
- eb6...76d ./programs/restaking/src/modules/fund/commands/cmd4_denormalize_nt.rs
- 66b...178 ./programs/restaking/src/modules/fund/commands/cmd5_claim_unstaked_sol.rs
- 721...65a ./programs/restaking/src/modules/fund/commands/cmd6_process_withdrawal_batch.rs
- 734...592 ./programs/restaking/src/modules/fund/commands/cmd7_unstake_lst.rs
- 36e...a3c ./programs/restaking/src/modules/fund/commands/cmd8_unrestake_vrt.rs
- 639...59a ./programs/restaking/src/modules/fund/commands/cmd9_undelegate_vst.rs
- acd...487 ./programs/restaking/src/modules/fund/commands/mod.rs
- 487...f16 ./programs/restaking/src/modules/fund/deposit_metadata.rs
- 89c...9e9 ./programs/restaking/src/modules/fund/fund_account.rs
- aa1...be6 ./programs/restaking/src/modules/fund/fund_account_asset_state.rs
- d82...9d8 ./programs/restaking/src/modules/fund/fund_account_normalized_token.rs
- a9a...26e ./programs/restaking/src/modules/fund/fund_account_operation_state.rs
- df7...d2b ./programs/restaking/src/modules/fund/fund_account_restaking_vault.rs
- efe...aca ./programs/restaking/src/modules/fund/fund_account_supported_token.rs
- 6da...0de ./programs/restaking/src/modules/fund/fund_account_token_swap_strategy.rs
- c45...c58 ./programs/restaking/src/modules/fund/fund_account_wrapped_token.rs
- a48...939 ./programs/restaking/src/modules/fund/fund_configuration_service.rs
- a1f...eae ./programs/restaking/src/modules/fund/fund_receipt_token_configuration_service.rs
- 4b5...637 ./programs/restaking/src/modules/fund/fund_receipt_token_value_provider.rs
- d73...3ac ./programs/restaking/src/modules/fund/fund_service.rs
- e36...809 ./programs/restaking/src/modules/fund/fund_withdrawal_batch_account.rs
- 55c...8cc ./programs/restaking/src/modules/fund/mod.rs
- 774...a23 ./programs/restaking/src/modules/fund/user_fund_account.rs
- 89c...816 ./programs/restaking/src/modules/fund/user_fund_configuration_service.rs
- 409...fef ./programs/restaking/src/modules/fund/user_fund_service.rs
- 536...2cf ./programs/restaking/src/modules/fund/user_fund_wrap_service.rs
- 54b...15b ./programs/restaking/src/modules/fund/weighted_allocation_strategy.rs
- ff1...78a ./programs/restaking/src/modules/mod.rs
- d0f...040 ./programs/restaking/src/modules/normalization/mod.rs

- 7e7...a65 ./programs/restaking/src/modules/normalization/normalized_token_pool_account.rs
- d91...e40 ./programs/restaking/src/modules/normalization/normalized_token_pool_configuration_service.rs
- d36...d92 ./programs/restaking/src/modules/normalization/normalized_token_pool_service.rs
- 7d7...317 ./programs/restaking/src/modules/normalization/normalized_token_pool_value_provider.rs
- b08...978 ./programs/restaking/src/modules/normalization/normalized_token_withdrawal_account.rs
- 790...f7c ./programs/restaking/src/modules/pricing/mod.rs
- 4e4...6c1 ./programs/restaking/src/modules/pricing/pricing_service.rs
- 92c...7ea ./programs/restaking/src/modules/pricing/token_pricing_source.rs
- b8f...5a8 ./programs/restaking/src/modules/pricing/token_value_provider.rs
- 3da...5df ./programs/restaking/src/modules/restaking/jito_restaking_vault_service.rs
- 396...89f ./programs/restaking/src/modules/restaking/jito_restaking_vault_value_provider.rs
- 6c1...0b5 ./programs/restaking/src/modules/restaking/mod.rs
- 700...629 ./programs/restaking/src/modules/restaking/solv_btc_vault_service.rs
- 125...60a ./programs/restaking/src/modules/restaking/solv_btc_vault_value_provider.rs
- 17a...d44 ./programs/restaking/src/modules/restaking/virtual_vault_service.rs
- 211...73a ./programs/restaking/src/modules/reward/mod.rs
- b37...b7e ./programs/restaking/src/modules/reward/reward.rs
- 933...6b0 ./programs/restaking/src/modules/reward/reward_account.rs
- 97f...98f ./programs/restaking/src/modules/reward/reward_configuration_service.rs
- 70c...4f6 ./programs/restaking/src/modules/reward/reward_pool.rs
- 232...e47 ./programs/restaking/src/modules/reward/reward_service.rs
- 99a...4d3 ./programs/restaking/src/modules/reward/reward_settlement.rs
- aa0...cba ./programs/restaking/src/modules/reward/token_allocated_amount.rs
- 019...ad8 ./programs/restaking/src/modules/reward/user_reward_account.rs
- f97...f8d ./programs/restaking/src/modules/reward/user_reward_configuration_service.rs
- fac...c50 ./programs/restaking/src/modules/reward/user_reward_pool.rs
- ff3...02c ./programs/restaking/src/modules/reward/user_reward_service.rs
- a60...816 ./programs/restaking/src/modules/reward/user_reward_settlement.rs
- 363...e33 ./programs/restaking/src/modules/staking/marinade_stake_pool_service.rs
- 0a8...61e ./programs/restaking/src/modules/staking/marinade_stake_pool_value_provider.rs

- `0bb...a6f ./programs/restaking/src/modules/staking/mod.rs`
- `a50...b02 ./programs/restaking/src/modules/staking/sanctum_multi_validator_spl_stake_pool_service.rs`
- `9ea...f83 ./programs/restaking/src/modules/staking/sanctum_single_validator_spl_stake_pool_service.rs`
- `e0b...ff5 ./programs/restaking/src/modules/staking/spl_stake_pool_service.rs`
- `60c...3bd ./programs/restaking/src/modules/staking/spl_stake_pool_value_provider.rs`
- `258...395 ./programs/restaking/src/modules/swap/mod.rs`
- `f43...eea ./programs/restaking/src/modules/swap/orca_dex_liquidity_pool_service.rs`
- `e4c...82e ./programs/restaking/src/modules/swap/orca_dex_liquidity_pool_value_provider.rs`
- `701...0ea ./programs/restaking/src/modules/swap/token_swap_source.rs`
- `4ec...f7e ./programs/restaking/src/utils.rs`

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Cargo Audit 0.16.0

Steps taken to run the tools:

Cargo Audit

- Installed via `cargo install cargo-audit`
- Ran `cargo audit`

Automated Analysis

Cargo Audit

1. Vulnerability: `curve25519-dalek`

```
Crate:     curve25519-dalek
Version:   3.2.0
Title:     Timing variability in `curve25519-dalek`'s
`Scalar29::sub`/`Scalar52::sub`
Date:      2024-06-18
ID:        RUSTSEC-2024-0344
```

```
URL:       https://rustsec.org/advisories/RUSTSEC-2024-0344
Solution:  Upgrade to >=4.1.3
```

2. Vulnerability: `ed25519-dalek`

```
Crate:     ed25519-dalek
Version:   1.0.1
Title:     Double Public Key Signing Function Oracle Attack on `ed25519-
dalek`
Date:      2022-06-11
ID:        RUSTSEC-2022-0093
URL:       https://rustsec.org/advisories/RUSTSEC-2022-0093
Solution:  Upgrade to >=2
```

3. Vulnerability: `openssl`

```
Crate:     openssl
Version:   0.10.71
Title:     Use-After-Free in `Md::fetch` and `Cipher::fetch`
Date:      2025-04-04
ID:        RUSTSEC-2025-0022
URL:       https://rustsec.org/advisories/RUSTSEC-2025-0022
Solution:  Upgrade to >=0.10.72
```

4. Warning: `derivative`

```
Crate:     derivative
Version:   2.2.0
Warning:   unmaintained
Title:     `derivative` is unmaintained; consider using an alternative
Date:      2024-06-26
ID:        RUSTSEC-2024-0388
URL:       https://rustsec.org/advisories/RUSTSEC-2024-0388
```

5. Warning: `paste`

```
Crate:     paste
Version:   1.0.15
Warning:   unmaintained
Title:     paste - no longer maintained
Date:      2024-10-07
ID:        RUSTSEC-2024-0436
URL:       https://rustsec.org/advisories/RUSTSEC-2024-0436
```

Test Suite Results

Setup

1. Install Rust v1.88.0

```
$ sh -c "$(curl -sSfL https://sh.rustup.rs)"
$ rustup install 1.88.0
$ rustup default 1.88.0
```

2. Install Solana CLI v2.1.21

```
$ sh -c "$(curl -sSfL https://release.anza.xyz/v2.1.21/install)"
```

3. Install Node v20.19.0

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/master/install.sh |
bash
$ nvm install node
$ nvm install 20.19.0
$ nvm use 20.19.0
```

4. Install AVM (Anchor Version Manager)

```
$ cargo install --git https://github.com/coral-xyz/anchor avm --force
$ avm install 0.31.1
$ avm use 0.31.1
```

5. Install PNPM

```
$ sh -c "$(curl -fsSL https://get.pnpm.io/install.sh)"
```

6. Install js Packages

```
$ pnpm i
```

7. Configure Program KeyPairs

```
$ pnpm keypairs:local
```

8. Build Program

```
$ anchor build
```

Unit Tests

```
$ cargo test-sbf -p restaking

running 28 tests
test constants::local::test_id ... ok
test modules::fund::commands::tests::size_command_buffer ... ok
test modules::fund::fund_account::tests::size_fund_account ... ok
test modules::fund::weighted_allocation_strategy::tests::test_capped_scenario
```

```
... ok
test modules::fund::fund_account_asset_state::tests::withdraw_basic_test ... ok
test modules::fund::weighted_allocation_strategy::tests::test_edge_scenario ...
ok
test modules::fund::weighted_allocation_strategy::tests::test_edge_scenario2 ...
ok
test modules::fund::weighted_allocation_strategy::tests::test_edge_scenario3 -
should panic ... ok
test modules::fund::weighted_allocation_strategy::tests::test_general_scenario
... ok
test modules::pricing::pricing_service::tests::size_token_pricing_source ... ok
test
modules::normalization::normalized_token_pool_value_provider::test::test_resolve
_from_buffer ... ok
test modules::fund::fund_account::tests::test_deposit_sol ... ok
test modules::pricing::pricing_service::tests::test_get_token_exchange_ratio ...
ok
test
modules::pricing::pricing_service::tests::test_get_token_exchange_ratio_with_peg
ged ... ok
test modules::fund::fund_account::tests::test_initialize_update_fund_account ...
ok
test modules::fund::fund_account::tests::test_update_token ... ok
test modules::pricing::pricing_service::tests::test_resolve_token_pricing_source
... ok
test modules::fund::fund_account::tests::test_deposit_token ... ok
test modules::reward::check_account_init_space::check_size ... ok
test modules::reward::reward_settlement::tests::test_claim_remaining_amount ...
ok
test modules::reward::reward_pool::tests::test_update_token_allocated_amount ...
ok
test
modules::reward::reward_settlement::tests::test_force_clear_when_block_is_full
... ok
test modules::reward::reward_settlement::tests::test_settlement ... ok
test
modules::pricing::pricing_service::tests::test_resolve_token_total_value_as_atom
ic ... ok
test modules::reward::token_allocated_amount::tests::test_clear_empty_records
... ok
test modules::reward::token_allocated_amount::tests::test_subtract ... ok
test modules::reward::user_reward_settlement::tests::test_settle_reward ... ok
test
modules::swap::orca_dex_liquidity_pool_value_provider::tests::test_resolve_token
_pricing_source ... ok

test result: ok. 28 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out;
finished in 0.03s

   Doc-tests restaking

running 0 tests
```

```
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out;
finished in 0.00s
```

Integration Tests

```
$ pnpm test ./programs/restaking/tests/*.test.ts

> @fragmetric-labs/contracts@ test /home/appuser/workspace/projects/AT-
3009/fragmetric_labs-fragmetric-contracts-github~full
> NODE_OPTIONS=--no-warnings=ExperimentalWarning pnpm vitest --run
"./programs/restaking/tests/frag2.test.ts"
"./programs/restaking/tests/fragbtc.test.ts"
"./programs/restaking/tests/fragjto.test.ts"
"./programs/restaking/tests/fragsol.test.ts"
"./programs/restaking/tests/fragsol.unit.test.ts"


 RUN  v3.1.1 /home/appuser/workspace/projects/AT-3009/fragmetric_labs-
fragmetric-contracts-github~full


 ✓ programs/restaking/tests/fragsol.unit.test.ts (11 tests) 51783ms
   ✓ restaking.fragSOL unit test > pricing source addresses field in fund
account updates correctly  1554ms
   ✓ restaking.fragSOL unit test > remove supported tokens  3050ms
   ✓ restaking.fragSOL unit test > remove token swap strategy  1278ms
   ✓ restaking.fragSOL unit test > user can update reward pools and sync with
global reward account anytime  425ms
   ✓ restaking.fragSOL unit test > user can deposit large amount of token  385ms
   ✓ restaking.fragSOL unit test > new supported token with new pricing source
deposits & withdraws without any issue  7702ms
   ✓ restaking.fragSOL unit test > operation disabled  15432ms
   ✓ restaking.fragSOL unit test > delegate reward account from user2 to user1
449ms
   ✓ restaking.fragSOL unit test > fails if trying to add wrong pricing source
when adding supported token  1089ms
   ✓ restaking.fragSOL unit test > fails to add compounding/distributing reward
token if provided reward token mint is not Mint account  558ms
   ✓ restaking.fragSOL unit test > Token should be pegged to non-pegging token
2464ms
 ✓ programs/restaking/tests/fragjto.test.ts (14 tests) 83336ms
   ✓ restaking.fragJTO test > rewards are settled based on the contribution
proportion  1481ms
   ✓ restaking.fragJTO test > rewards can be settled with custom contribution
accrual rate enabled  957ms
   ✓ restaking.fragJTO test > contribution is accumulated with users who have
user_reward_account  1649ms
   ✓ restaking.fragJTO test > wrapping FRAGXXX affects token allocated amount of
user, but global reward account maintains same amount  1582ms
   ✓ restaking.fragJTO test > record with low contribution rate is deleted first
1598ms
```

```
   ✓ restaking.fragJTO test > user can deposit JTO  381ms
   ✓ restaking.fragJTO test > user can withdraw receipt token as JTO  4119ms
   ✓ restaking.fragJTO test > jitoSOL reward is swapped to JTO then compounded
2190ms
   ✓ restaking.fragJTO test > reward is transferred to revenue account based on
commission rate during harvest command execution (swap reward)  32173ms
   ✓ restaking.fragJTO test > run operation cycles through multiple epoches to
test cash-in/out flows including (un)stake/(un)restake  28705ms
 ✓ programs/restaking/tests/frag2.test.ts (14 tests) 96843ms
   ✓ restaking.frag2 test > user can deposit frag  569ms
   ✓ restaking.frag2 test > user can withdraw receipt token as frag  1328ms
   ✓ restaking.frag2 test > virtual vault harvest/compound  1445ms
   ✓ restaking.frag2 test > virtual vault harvest/compound should not occur by
compounding threshold  3825ms
   ✓ restaking.frag2 test > virtual vault harvest/distribute  1110ms
   ✓ restaking.frag2 test > reward settlement clears one block before block
addition when block queue is full  2647ms
   ✓ restaking.frag2 test > operator also claims leftovers from cleared block
after settle  1364ms
   ✓ restaking.frag2 test > reward is transferred to revenue account based on
commission rate during harvest command execution (compound reward, distribute
reward)  74174ms
   ✓ restaking.frag2 test > run full operation cycle for regression  6019ms
 ✓ programs/restaking/tests/fragsol.test.ts (12 tests) 98131ms
   ✓ restaking.fragSOL test > user can deposit SOL  397ms
   ✓ restaking.fragSOL test > user can deposit token with SPLStakePool pricing
source  581ms
   ✓ restaking.fragSOL test > user can withdraw receipt tokens as SOL  4034ms
   ✓ restaking.fragSOL test > fund manager can add wrapped token holder  719ms
   ✓ restaking.fragSOL test > wrapped token holder amount is updated by operator
1867ms
   ✓ restaking.fragSOL test > wrapped token retained amount remains non-negative
2839ms
   ✓ restaking.fragSOL test > run operation cycles through multiple epoches to
test cash-in/out flows including (un)stake/(un)restake  61072ms
   ✓ restaking.fragSOL test > there could be remaining lamports in uninitialized
fund withdrawal stake accounts, due to jito tip  2658ms
 ✓ programs/restaking/tests/fragbtc.test.ts (16 tests) 128590ms
   ✓ restaking.fragBTC test > user can deposit token with OrcaDEXLiquidityPool
pricing source  561ms
   ✓ restaking.fragBTC test > user can deposit token with PeggedToken pricing
source  940ms
   ✓ restaking.fragBTC test > fund can settle distributing rewards by operation
3380ms
   ✓ restaking.fragBTC test > settle should not occur if threshold is not
matched  12979ms
   ✓ restaking.fragBTC test > settle -> everyone claim -> update reward pool ->
check the remaining amount  574ms
   ✓ restaking.fragBTC test > funds supporting only a single token and tokens
pegged to it must issue receipt tokens at a 1:1 ratio until additional yield is
compounded  9833ms
   ✓ restaking.fragBTC test > fragBTC does restake/unrestake assets into/from
solvBTC vault  37258ms
```

```
   ✓ restaking.fragBTC test > fragBTC APY can be estimated through vault
supported token compounded amount  32684ms
   ✓ restaking.fragBTC test > duplicate withdrawal requests can be prevented by
adopting pending_supported_token_unrestaking_amount  20852ms
   ✓ restaking.fragBTC test > user can deposit srt and receive rt  884ms
   ✓ restaking.fragBTC test > new supported token should be pegged to registered
token if pricing source of the registered one is manipulatable  1455ms

 Test Files  5 passed (5)
      Tests  67 passed (67)
   Start at  15:04:27
   Duration  131.83s (transform 1.60s, setup 0ms, collect 14.56s, tests 458.68s,
environment 0ms, prepare 322ms)
```

Changelog

- 2025-09-24 - Initial report
- 2025-10-08 - Final report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

assumes no obligation to update any information following publication or other making available of the report to you by Quantstamp.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.