



SMART CONTRACT SECURITY AUDIT

Fluid Protocol

Core Protocol

CLIENT

Fluid Protocol

REVIEW

October 2024

Document information

Project	Fluid Protocol, Core Protocol
Client	Fluid Protocol
Platform	Not recorded in the original report
Language	Not recorded in the original report
Repository	Not recorded in the original report
Commit	Not recorded in the original report
Methods	Not recorded in the original report
Assessment date	20 October 2024

Document history

VERSION	DATE	CHANGE
1.0	20 October 2024	Original report issued View original report
1.1	October 2026	Reformatted into the Safe Edges standard template. Findings unchanged

This report was reformatted into the Safe Edges standard template in October 2026. Findings, severities, recommendations and code references are unchanged from the original report, which is preserved unchanged in the [Safe Edges Audit Portfolio on GitHub](#).

CONTENTS

Contents

01	Executive summary	3
02	Scope	4
03	Methodology	5
04	Severity and status	6
05	Findings summary	7
06	Detailed findings	8
	C-01 Reward Snapshot Misalignment	8
	C-02 State Update Failure on Redemption Cancellation	8
	H-01 Incorrect Oracle Price Structure	8
	H-02 Incorrect Trove Retrieval Logic	9
	H-03 Timestamp Synchronization Flaw	9
	M-01 Trove Redemption Order Violation	9
	M-02 Missing Validation Checks	9
	M-03 Imprecise Reward Distribution	10
	M-04 Validation Logic Inconsistency	10
07	Security recommendations	11
08	Conclusion	11
	Disclaimer	13

Executive summary

About Fluid Protocol

Fluid Protocol is a decentralized liquidity platform for collateralized borrowing.

Summary of the review

Safe Edges audited the Fluid Protocol in October 2024 across reward distribution, liquidation logic, oracle integration and collateral management. This report details nine findings: two Critical, three High and four Medium, all resolved. The Critical issues concerned reward snapshots after partial liquidations and trove state after cancelled redemptions.

Findings by severity



Findings by status

STATUS	ISSUES
Resolved	9

Scope

Repository	Not recorded in the original report
------------	-------------------------------------

Commit	Not recorded in the original report
--------	-------------------------------------

Platform	Not recorded in the original report
----------	-------------------------------------

Language	Not recorded in the original report
----------	-------------------------------------

FOCUS AREAS

- Reward distribution
- Liquidation logic
- Oracle integration
- Collateral management

Only the code at the commit above was reviewed. Changes made after this commit are not covered by this report.

Methodology

Methods recorded for this engagement: Not recorded in the original report.

Manual review

Auditors read the in-scope code line by line, tracing control flow, state changes and trust boundaries against the intended behaviour described by the client.

Threat modelling

We map privileged roles, external dependencies and value flows, then test the assumptions the code makes about each of them.

Functional and automated testing

Where tooling exists for the language and chain, we run static analysis and targeted tests to confirm suspected issues and check edge cases.

Fix review

After the client addresses the findings, we review the updated code and record the final status of each issue.

Severity and status

Severity levels

SEVERITY	DEFINITION
Critical	Direct loss or permanent lock of funds, or full compromise of the protocol, with few or no preconditions. Must be fixed before deployment.
High	Significant loss of funds, broken core functionality or manipulation of critical state under realistic conditions. Should be fixed before deployment.
Medium	Limited loss or degraded functionality, or issues that need specific conditions or privileged access to exploit. Should be fixed.
Low	Minor impact, low likelihood, or deviations from best practice. Fix when practical.
Informational	Code quality, documentation, gas and style observations with no direct security impact.

Status definitions

STATUS	DEFINITION
Resolved	The issue was fixed and the fix was verified.
Partially resolved	The risk was reduced but not fully removed.
Acknowledged	The client accepted the issue without changing the code.
Open	The issue was not addressed at the time of the report.
Not recorded	The original report did not record a resolution status for this issue.

Findings summary

ID	TITLE	SEVERITY	STATUS
C-01	Reward Snapshot Misalignment	Critical	● Resolved
C-02	State Update Failure on Redemption Cancellation	Critical	● Resolved
H-01	Incorrect Oracle Price Structure	High	● Resolved
H-02	Incorrect Trove Retrieval Logic	High	● Resolved
H-03	Timestamp Synchronization Flaw	High	● Resolved
M-01	Trove Redemption Order Violation	Medium	● Resolved
M-02	Missing Validation Checks	Medium	● Resolved
M-03	Imprecise Reward Distribution	Medium	● Resolved
M-04	Validation Logic Inconsistency	Medium	● Resolved

Detailed findings

C-01 Reward Snapshot Misalignment

Severity **Critical** Status ● Resolved

DESCRIPTION

Partial liquidations did not update the borrower's reward snapshot correctly.

IMPACT

This led to reward miscalculations and the possibility of double-counted rewards.

RESOLUTION

Snapshots are now updated immediately post-liquidation; batch logic improved.

C-02 State Update Failure on Redemption Cancellation

Severity **Critical** Status ● Resolved

DESCRIPTION

Cancelling partial redemptions didn't update trove state or reintegrate them into the sorted structure.

IMPACT

Caused inconsistencies in system state.

RESOLUTION

Proper reinsertion and state syncing logic implemented.

H-01 Incorrect Oracle Price Structure

Severity **High** Status ● Resolved

ISSUE

The oracle integration missed critical fields like confidence intervals and exponents.

IMPACT

Risk of inaccurate price feeds affecting operations.

RESOLUTION

Updated to use full Pyth structure with validations.

H-02 Incorrect Trove Retrieval Logic

Severity **High** Status ● Resolved

ISSUE

Troves with the highest collateral ratio were used instead of the lowest.

IMPACT

Allowed undercollateralized troves to persist.

RESOLUTION

Retrieval logic corrected to identify lowest-ratio positions first.

H-03 Timestamp Synchronization Flaw

Severity **High** Status ● Resolved

ISSUE

Time drift between on-chain timestamp and oracle could cause false reverts.

IMPACT

Oracle data was rejected despite being valid.

RESOLUTION

Enhanced staleness logic to account for drift.

M-01 Trove Redemption Order Violation

Severity **Medium** Status ● Resolved

DESCRIPTION

Reward calculations during redemption altered trove order unexpectedly.

M-02 Missing Validation Checks

Severity **Medium** Status ● Resolved

DESCRIPTION

Batch liquidation lacked validations for sorting and duplication.

M-03 Imprecise Reward Distribution

Severity **Medium** Status ● Resolved

DESCRIPTION

Pending rewards were excluded from stake-based reward calculation.

M-04 Validation Logic Inconsistency

Severity **Medium** Status ● Resolved

DESCRIPTION

Insert position logic referenced incorrect node pointers.

Security recommendations

Oracle integration

- Validate oracle confidence intervals before using price data
- Account for time drift between off-chain and on-chain timestamps
- Follow oracle safety best practices

Liquidation system

- Validate trove states before processing
- Enforce risk-based ordering in liquidation logic
- Add edge-case checks to batch functions

Reward mechanism

- Always update snapshots after state changes
- Reflect pending rewards in calculations
- Avoid state mismatches during batched logic

Code quality

- Remove unnecessary checks
- Add zero-amount transfer validations
- Standardize access control and ownership patterns

Low priority improvements

- Code optimization: eliminate redundant logic and dead code, improve gas efficiency, enhance readability and modularity
- Input validation: validate all external input parameters, check asset IDs and token references properly, add detailed error messages for failed conditions
- Time-based operations: use precise time constants, optimize oracle timeouts, handle outdated prices more gracefully

Conclusion

The Fluid Protocol has addressed all critical and high-severity vulnerabilities discovered during the audit. Improvements were implemented across multiple layers including redemption logic, oracle integrations, liquidation handling, and reward mechanisms. These changes enhance the system's resilience, accuracy, and operational integrity.

While a few medium and low-priority issues remain as potential optimization areas, none pose direct threats to protocol safety. The Fluid Protocol is now well-prepared to operate securely under normal and edge-case conditions.

Disclaimer

This report is based on the code at the commit listed in the Scope section and on information the client provided during the review. It reflects Safe Edges' assessment at the time of the review and is not a guarantee that the code is free of vulnerabilities.

Changes made after the reviewed commit, deployment configuration, off-chain components and third-party dependencies are outside the scope of this report unless stated otherwise.

Security is ongoing. We recommend continued testing, monitoring and bug bounty coverage alongside any audit.

This report is not investment advice and is not an endorsement of the project, its team or any token.



Safe Edges is a Web3 security firm and official security partner for the Fuel, ENI and XI EcoChain ecosystems. We work across smart contract audits, Rust and WASM reviews, ZK circuit review, AI and MCP security, and penetration testing.

200+

PROJECTS AUDITED

\$4B+

TVL SECURED

0

POST-AUDIT HACKS