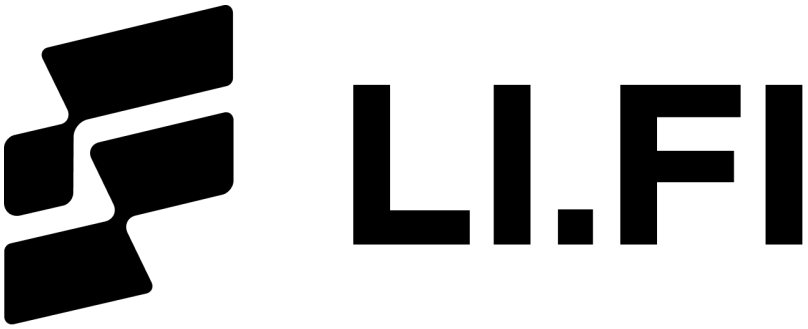

LI.FI Security Review

MegaETHBridgeFacet.sol(v1.0.0)

Security Researcher

Sujith Somraaj (somraajsujith@gmail.com)

Report prepared by: Sujith Somraaj

December 3, 2025

Contents

1 About Researcher **2**

2 Disclaimer **2**

3 Scope **2**

4 Risk classification **2**
4.1 Impact . . . 2
4.2 Likelihood . . . 3
4.3 Action required for severity levels . . . 3

5 Executive Summary **3**

6 Findings **4**
6.1 Gas Optimization . . . 4
6.1.1 Unnecessary intermediate variable in bridge selection . . . 4
6.2 Informational . . . 4
6.2.1 Use named constant for empty bytes parameter . . . 4
6.2.2 Inconsistent Zero Address Handling . . . 4
6.2.3 Missing `assetIdOnL2` Validation . . . 4

1 About Researcher

Sujith Somraaj is a distinguished security researcher and protocol engineer with over eight years of comprehensive experience in the Web3 ecosystem.

In addition to working as a Lead Security Researcher at Spearbit, Sujith is also the security researcher and advisor for leading bridge protocol LI.FI and also is a former founding engineer and current security advisor at Superform, a yield aggregator with over $170M in TVL.

Sujith has experience working with protocols / funds including Coinbase, Uniswap, Layerzero, Edge Capital, Berachain, Optimism, Ondo, Sonic, Monad, Blast, ZkSync, Decent, Drips, SuperSushi Samurai, DistrictOne, Omni-X, Centrifuge, Superform-V2, Tea.xyz, Paintswap, Bitcorn, Sweep n' Flip, Byzantine Finance, Variational Finance, Satsbridge, Rova, Horizen, Earthfast and Angles

Learn more about Sujith on sujithsomraaj.xyz or on cantina.xyz

2 Disclaimer

Note that this security audit is not designed to replace functional tests required before any software release, and does not give any warranties on finding all possible security issues of that given smart contract(s) or blockchain software. i.e., the evaluation result does not guarantee against a hack (or) the non existence of any further findings of security issues. As one audit-based assessment cannot be considered comprehensive, I always recommend proceeding with several audits and a public bug bounty program to ensure the security of smart contract(s). Lastly, the security audit is not an investment advice.

This review is done independently by the reviewer and is not entitled to any of the security agencies the researcher worked / may work with.

3 Scope

- src/Facets/MegaETHBridgeFacet.sol(v1.0.0)

4 Risk classification

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

4.1 Impact

High leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.

Medium global losses <10% or losses to only a subset of users, but still unacceptable.

Low losses will be annoying but bearable — applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.

4.2 Likelihood

High almost certain to happen, easy to perform, or not easy but highly incentivized

Medium only conditionally possible or incentivized, but still relatively likely

Low requires stars to align, or little-to-no incentive

4.3 Action required for severity levels

Critical Must fix as soon as possible (if already deployed)

High Must fix (before deployment if not already deployed)

Medium Should fix

Low Could fix

5 Executive Summary

Over the course of 8 hours in total, LI.FI engaged with the researcher to audit the contracts described in section 3 of this document ("scope").

In this period of time a total of 4 issues were found.

Project Summary	
Project Name	LI.FI
Repository	lifinance/contracts
Commit	721afd24
Audit Timeline	December 2, 2025
Methods	Manual Review
Documentation	Medium-High
Test Coverage	Medium-High

Issues Found	
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	0
Gas Optimizations	1
Informational	3
Total Issues	**4**

6 Findings

6.1 Gas Optimization

6.1.1 Unnecessary intermediate variable in bridge selection

Context: MegaETHBridgeFacet.sol#L168-L175

Description: The current implementation uses an unnecessary intermediate variable, nonStandardBridge when selecting which bridge to use. This creates additional memory allocation and slightly increases gas costs.

Recommendation: Simplify the bridge selection logic by directly assigning to the bridge variable and using an if-statement:

```
IL1StandardBridge bridge = s.bridges[_bridgeData.sendingAssetId];

if (LibUtil.isZeroAddress(address(bridge))) {
   bridge = s.standardBridge;
}
```

LI.FI: Fixed in 32905de

Researcher: Verified fix.

6.2 Informational

6.2.1 Use named constant for empty bytes parameter

Context: MegaETHBridgeFacet.sol#L181, MegaETHBridgeFacet.sol#L201

Description: The contract uses hardcoded "" (empty bytes) as the _data parameter when calling bridge functions.

According to the IL1StandardBridge interface, this data parameter is "provided solely as a convenience for external contracts" and is optional. However, using a magic value ("") instead of a named constant reduces code readability and maintainability.

Recommendation: Define a constant for empty bridge data and use it consistently

LI.FI: Fixed in 84d3ee4

Researcher: Verified fix

6.2.2 Inconsistent Zero Address Handling

Context: MegaETHBridgeFacet.sol#L68, MegaETHBridgeFacet.sol#L76 and MegaETHBridgeFacet.sol#L100

Description: The contract uses both `LibUtil.isZeroAddress()` and direct `address(0)` comparisons. For consistency, standardize on one approach.

LI.FI: Fixed in 94f5cda

Researcher: Verified fix.

6.2.3 Missing `assetIdOnL2` Validation

Context: MegaETHBridgeFacet.sol#L40

Description: The **assetIdOnL2** parameter is passed directly to `depositERC20To()` without validation when **requiresDepositTo** is false. If a user accidentally (or maliciously) passes `address(0)` as the L2 token address, the bridge call will proceed with a zero address.

Recommendation: Add validation before the `depositERC20To()` call:

```
+ if(LibUtils.isZeroAddress(_megaETHData.assetIdOnL2) revert InvalidAssetIdOnL2();

bridge.depositERC20To(
  _bridgeData.sendingAssetId,
  _megaETHData.assetIdOnL2,
  _bridgeData.receiver,
  _bridgeData.minAmount,
  _megaETHData.l2Gas,
  ""
);
```

LI.FI: Fixed in 740b0d9

Researcher: Verified fix.