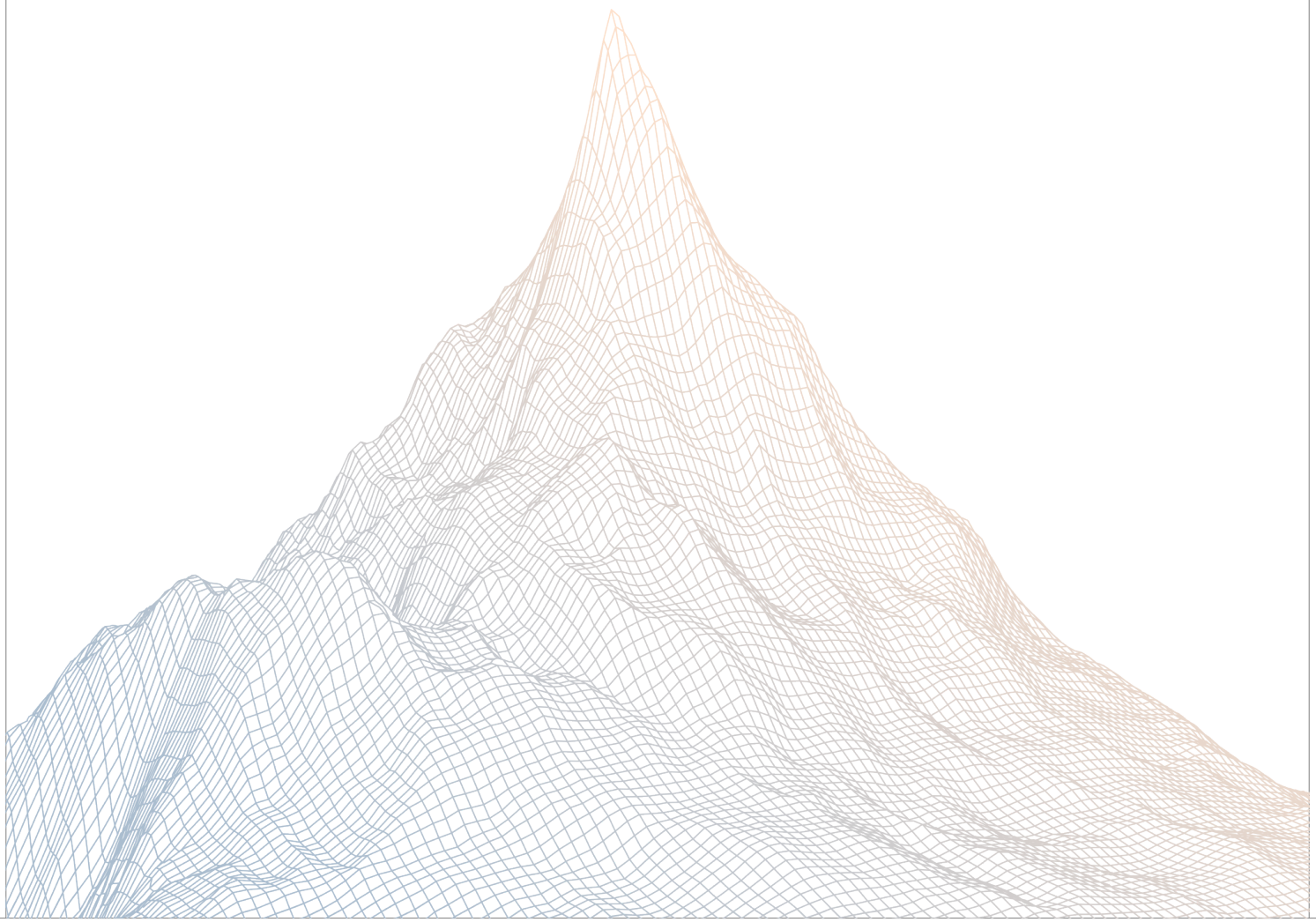
Zenith

# Bridge

## Smart Contract Security Assessment

VERSION 1.1



AUDIT DATES: May 12th to May 13th, 2025

AUDITED BY: Matte
Peakbolt

# Contents

# 1 Introduction

## 1.1 About Zenith

Zenith assembles auditors with proven track records: finding critical vulnerabilities in public audit competitions.

Our audits are carried out by a curated team of the industry's top-performing security researchers, selected for your specific codebase, security needs, and budget.

Learn more about us at https://zenith.security

## 1.2 Disclaimer

This report reflects an analysis conducted within a defined scope and time frame, based on provided materials and documentation. It does not encompass all possible vulnerabilities and should not be considered exhaustive.

The review and accompanying report are presented on an "as-is" and "as-available" basis, without any express or implied warranties.

Furthermore, this report neither endorses any specific project or team nor assures the complete security of the project.

## 1.3 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 |

# 2

# Executive Summary

## 2.1 About Bridge

Bridge gives everyone access to world-class financial services. We believe stablecoins will transform and improve global money movement. Bridge creates the infrastructure necessary for builders to take full advantage of this new medium.

Since launching 18 months ago, we've provided millions with faster and cheaper access to cross-border payments, enabled governments and aid agencies to more efficiently distribute funds to thousands, and given millions more true economic choice, enabling them to easily save and spend in USD or EUR.

## 2.2 Scope

The engagement involved a review of the following targets:

| | |
|---|---|
| **Target** | deterministic-proxy-factory |
| **Repository** | https://github.com/emo-eth/deterministic-proxy-factory |
| **Commit Hash** | `c73e706441a9ea0c0d5a9892ff5eb39b5fe8beaa` |
| **Files** | `src/*` |

## 2.3 Audit Timeline

| | |
|---|---|
| **May 12, 2025** | Audit start |
| **May 13, 2025** | Audit end |
| **May 14, 2025** | Report published |

## 2.4 Issues Found

| SEVERITY | COUNT |
|---|---|
| Critical Risk | 0 |
| High Risk | 0 |
| Medium Risk | 0 |
| Low Risk | 0 |
| Informational | 1 |
| **Total Issues** | **1** |

# 3

## Findings Summary

| ID | Description | Status |
|---|---|---|
| I-1 | Un-used ProxyDeploymentFailed error | Resolved |

# 4

Findings

## 4.1 Informational

A total of 1 informational findings were identified.

### [I-1] Un-used `ProxyDeploymentFailed` error

| SEVERITY: Informational | IMPACT: Informational |
| --- | --- |
| STATUS: Resolved | LIKELIHOOD: Low |

#### Target

- DeterministicProxyFactory.sol#L21

#### Description:

The `ProxyDeploymentFailed` error is currently un-used and can be removed as the solady `LibClone` functions rely on its own `DeploymentFailed` error for failed deployments.

```
contract DeterministicProxyFactory {

    /// @notice Reverts if the caller is not encoded into the top 160 bits of
    the salt.
    error InvalidDeployer();
    /// @notice Reverts if the proxy deployment fails.
    error ProxyDeploymentFailed();
```

#### Recommendations:

```
contract DeterministicProxyFactory {

    /// @notice Reverts if the caller is not encoded into the top 160 bits of
    the salt.
    error InvalidDeployer();
    /// @notice Reverts if the proxy deployment fails.
    error ProxyDeploymentFailed();
```

**Bridge:** Resolved with @781eca21f2...

**Zenith:** Verified.