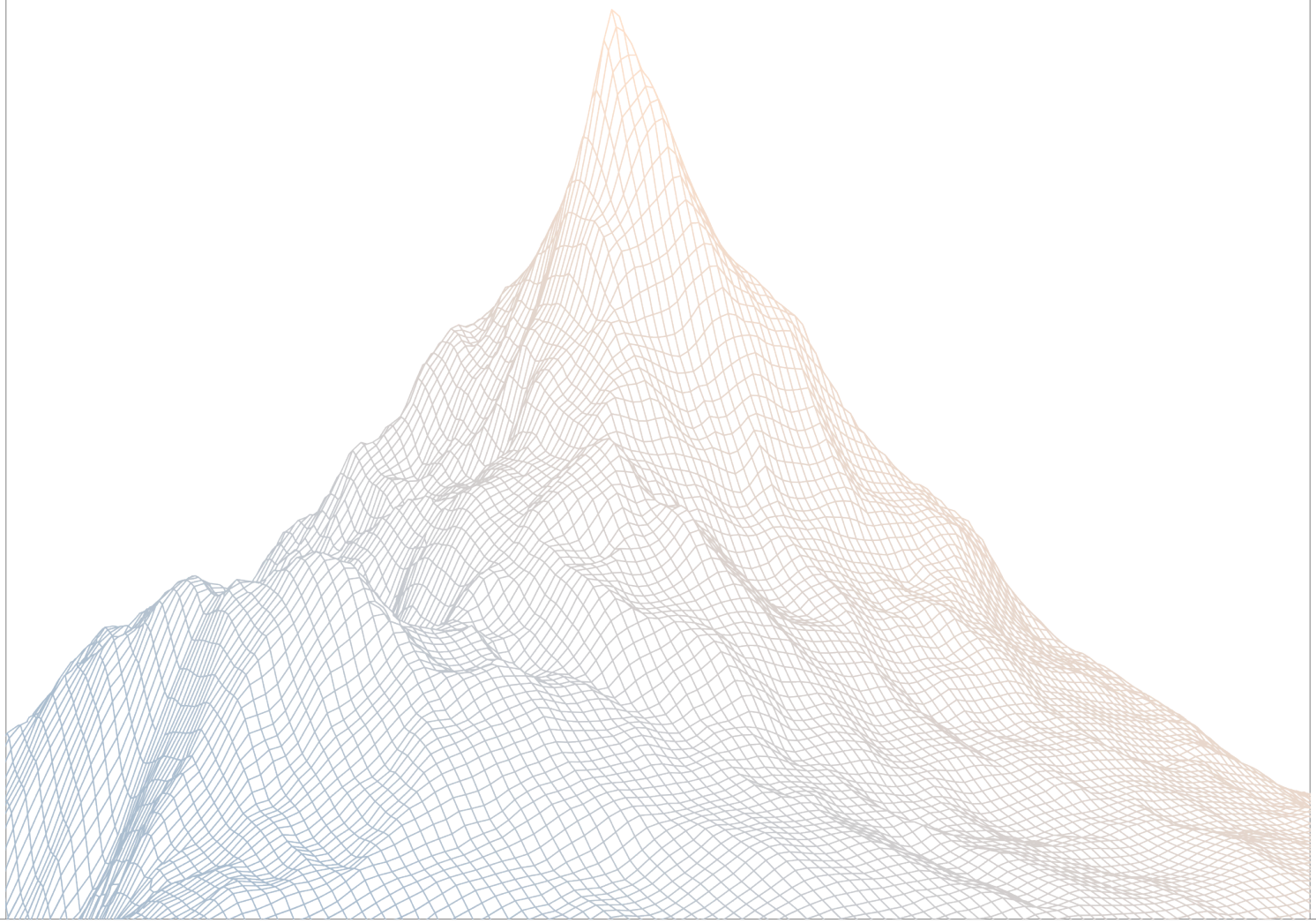
Zenith

Meteora

Smart Contract Security Assessment

VERSION 1.1



AUDIT DATES: December 31st, 2025 to January 2nd, 2026

AUDITED BY: 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 Meteora

Our mission is to build the most secure, sustainable and composable liquidity layer for all of Solana and DeFi.

By using Meteora's DLMM and Dynamic AMM Pools, liquidity providers can earn the best fees and yield on their capital.

This would help transform Solana into the ultimate trading hub for mainstream users in crypto by driving sustainable, long-term liquidity to the platform. Join us at Meteora to shape Solana's future as the go-to destination for all crypto participants.

2.2 Scope

The engagement involved a review of the following targets:

Target	dynamic-bonding-curve
Repository	https://github.com/MeteoraAg/dynamic-bonding-curve
Commit Hash	`e62727a9284acf800fe56c5e994912f960641997`
Files	`Changes in PR #151`

2.3 Audit Timeline

December 31, 2025	Audit start
January 2, 2026	Audit end
January 5, 2026	Report published

2.4 Issues Found

SEVERITY	COUNT
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	0
Informational	2
Total Issues	**2**

3

Findings Summary

ID	Description	Status
I-1	Wrong error message for PoolError::InvalidTokenDecimals	Acknowledged
I-2	claim_partner_pool_creation_fee event omits the fee_receiver address	Acknowledged

4

Findings

4.1 Informational

A total of 2 informational findings were identified.

[I-1] Wrong error message for `PoolError::InvalidTokenDecimals`

SEVERITY: Informational	IMPACT: Informational
STATUS: Acknowledged	LIKELIHOOD: Low

Target

- error.rs#L56-L57

Description:

`create_config()` validates `token_decimal` is within the supported range (6—9).

However, `PoolError::InvalidTokenDecimals` is defined with an incorrect error message (`"Invalid activation type"`), causing the program to emit a misleading failure reason.

```
#[msg("Invalid activation type")]
InvalidTokenDecimals,
```

Recommendations:

Update `PoolError::InvalidTokenDecimals` to a correct message (e.g., `"Invalid token decimals"`).

Meteora: Acknowledged, and will fix in next release.

[I-2] `claim_partner_pool_creation_fee` event omits the `fee_receiver` address

SEVERITY: Informational	IMPACT: Informational
STATUS: Acknowledged	LIKELIHOOD: Low

Target

- ix_claim_partner_pool_creation_fee.rs#L44-L48

Description:

`claim_partner_pool_creation_fee()` lets the partner fee claimer withdraw the partner pool creation fee to a specified `fee_receiver`.

However, the emitted event does not include the `fee_receiver` pubkey even though funds are transferred to that account.

```
emit_cpi!(EvtPartnerClaimPoolCreationFee {
    pool: ctx.accounts.pool.key(),
    partner: ctx.accounts.fee_claimer.key(),
    creation_fee: partner_fee,
});
```

Recommendations:

Include `ctx.accounts.fee_receiver.key()` in the emitted event `EvtPartnerClaimPoolCreationFee`.

Meteora: Acknowledged, and will fix in next release.