yAudit Euler Protocol Deployment Review

Auditors:

- Adriro
- Invader-Tak

Table of Contents

Review Summary

Euler Protocol

Euler is a decentralized finance protocol that provides capital-efficient borrowing and lending services. This review assesses the feasibility and compatibility of deploying the core component of Euler protocol on two new blockchains: Polygon and Avalanche. It attempts to identify potential compatibility issues, deployment risks, and critical differences between the Ethereum mainnet and the new blockchains. The review is limited to a technical evaluation of contract deployment aspects and does not evaluate security practices or operational security. The review may not have identified all potential issues associated with deploying Euler protocol on a new chain.

yAudit and the auditors make no warranties regarding the security of the code and do not warrant that the code is free from defects. yAudit and the auditors do not represent nor imply to third parties that the code has been audited nor that the code is free from defects. By deploying or using the code, Euler and users of the contracts agree to use the code at their own risk.

Evaluation Matrix and Findings Explanation

The following spreadsheet gives a breakdown of what potential conflicts and issues were assessed during the review:

Euler new chain deployment checklist

Below, you can find an explanation of the assessed categories.

Category	Description
Opcode Support or Diff Semantics	Evaluates compatibility of EVM opcodes and differing EIP implementations on the target blockchains, ensuring that new chains support necessary opcodes (like PUSH0) and semantic differences.
Blocks	Assesses potential risks related to block processing, including reorg susceptibility, block time, and numbering dependencies, which could affect protocol behavior across different chains.
Mempool	Analyzes mempool structure and behavior on the target blockchains, focusing on whether they have public or private mempools and how this affects MEV (Miner Extractable Value) strategies.
Others	Covers various cross-chain and chain-specific considerations, such as signature replay protection, deterministic deployment methods, and the impact of chain-specific precompiles and consensus mechanisms.
Oracles and Price Feeds	Verifies the availability, stability, and consistency of external price feeds and data providers on the new blockchains, such as Chainlink, Pyth, and Redstone.
Euler Specific	Focuses on features or configurations specific to the Euler protocol, including hardcoded addresses, dispatch methods, and subaccount generation logic, ensuring these elements remain compatible with the new chains.

Opcode Support or Diff Semantics

1. PUSH0

Reference documentation

EIP-3855

Auditor Comments

As long as contracts are compiled with Solc version ≥0.8.20 and EVM version Shanghai, the contracts will use PUSH0. This may cause issues with chains that don't support PUSH0.

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

2. Changes introduced in the Cancun upgrade

Reference documentation

EIP-1153 EIP-5656 EIP-6780

Auditor Comments

Several new Opcodes and changes were introduced in Cancun, including Transient Storage, MCOPY, and changes in how self-destruct is handled.

Recommendation

Be aware that Avalanche currently doesn't support any of the listed upgrades.

Chain support

Polygon	Avalanche
✓	

Blocks

1. Block reorgs

Reference documentation

N/A

Auditor Comments

The reorg attack can happen in the GenericFactory.sol contract due to the way proxies for vaults are created. This issue has already been mentioned in the Spearbit audit (issue 5.5.3).

Recommendation

Avalanche does not have reorgs, meaning that any issues caused by reorgs aren’t applicable there.

Chain support

Polygon	Avalanche
✓	

2. Block time or Numbering dependency

Reference documentation

N/A

Auditor Comments

The protocol doesn't use `block.number` and `block.timestamp` should be OK for Avalanche and Polygon (though production is faster).

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

Mempool

1. Mempool discrepancies

Reference documentation

N/A

Auditor Comments

Avalanche does not have a public mempool, but this does not protect users from MEV, as it's still available for AVAX stakers and through some paid services.

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

Other Chain features

1. Signature replay across chains

Reference documentation

N/A

Auditor Comments

The protocol uses EIP712, so it should not be vulnerable to cross-chain signature replay attacks.

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

2. Permit2 support

Reference documentation

Uniswap deployments

Auditor Comments

The Permit2 contract is deployed to Polygon and Avalanche (at the same address as the mainnet).

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

3. Address derivation

Reference documentation

N/A

Auditor Comments

There are no differences in how the create and create2 Opcode generate addresses. `calculateDTokenAddress()` should be fine.

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

4. Deterministic deployment

Reference documentation

Deterministic deployer

Auditor Comments

Factories for deterministic deployment exist on both chains.

Recommendation

In case there are contracts that must be deployed at the same address across multiple chains,

Chain support

Polygon	Avalanche
✓	✓

5. Precompiles

Reference documentation

EIP-7212

Auditor Comments

Polygon has a precompile (secp256r1) available at an address greater than 0xFF, which may break some assumptions in the Euler code (see `isSignerValid()`). Note that this precompile will be included in more chains, including the Ethereum mainnet, as it allows for passkey/webauthn validation to occur on-chain.

Recommendation

Be aware that this may break some assumptions made in the code.

Chain support

Polygon	Avalanche
✓	

Oracles

1. Chainlink

Reference documentation

Polygon feeds Avalanche feeds

Auditor Comments

Some feeds in POL have 27s heartbeat, which is less than half the minimum stale timeout.

Recommendation

Be conscious that Polygon feeds staleness checks may have to be adjusted.

Chain support

Polygon	Avalanche
✓	✓

2. Uniswap

Reference documentation

Uniswap v3 TWAP Oracles in Proof of Stake

Auditor Comments

Faster block production could allow easier TWAP manipulation, although this is already mentioned in the contract's documentation.

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

3. Pyth

Reference documentation

Pyth Oracle addresses

Auditor Comments

Available on both Avalanche and Polygon.

Recommendation

None.

Chain support

Polygon	Avalanche
✓	✓

4. Chronicle

Reference documentation

Auditor Comments

There are currently no oracles running on Polygon PoS and Avalanche.

Recommendation

None.

Chain support

Polygon	Avalanche

5. Redstone

Reference documentation

- Redstone Supported Chains
- Redstone Data Services

Auditor Comments

Both Polygon and Avalanche are supported by Redstone. Take into account that in order to use a specific data service (such as the Avalanche data service) the adapter must inherit from the proper Redstone base contract. Currently, RedstoneCoreOracle.sol extends the PrimaryProdDataServiceConsumerBase data service.

Recommendation

The Redstone adapter might need some adjustments if planning on using another data service.

Chain support

Polygon	Avalanche
✓	✓

6. Lido

Reference documentation

N/A

Auditor Comments

The LidoOracle.sol and LidoFundamentalOracle.sol adapters rely on hardcoded addresses for the stETH and wstETH contracts. These hardcoded addresses don't exist on either chain, but they are not of any concern as LIDO integration isn't relevant on either chain.

Recommendation

None.

Chain support

Polygon	Avalanche

Final Remarks

The deployment of the Euler protocol on Polygon and Avalanche is feasible and should not require any adjustments. However, in case the differences highlighted in this review affect any external interaction or monitoring strategies, these should be adjusted accordingly.