MixBytes()

Resolv ER Coordinator Security Audit Report

JANUARY 15, 2026

Table of Contents

1. Introduction

1.1 Disclaimer

The audit makes no statements or warranties regarding the utility, safety, or security of the code, the suitability of the business model, investment advice, endorsement of the platform or its products, the regulatory regime for the business model, or any other claims about the fitness of the contracts for a particular purpose or their bug-free status.

1.2 Executive Summary

Resolv is a DeFi protocol that issues the delta-neutral stablecoin **USR**, backed by crypto collateral; **RLP** represents liquidity provider shares and risk exposure; **wstUSR** is the yield-accruing wrapped form of USR; and **RESOLV** is the governance and incentive token.

The `ExternalRequestsCoordinator` contract provides a unified interface for completing external mint and burn requests for both USR and RLP tokens.

This audit was conducted using manual code review, static analysis tools, and security best practices. We went through our detailed checklist, covering other aspects such as business logic, common ERC20 issues, interactions with external contracts, integer overflows, reentrancy attacks, access control, typecast pitfalls, rounding errors and other potential issues.

The scope of this audit was limited to the `ExternalRequestsCoordinator.sol` contract. Other contracts in the codebase, including the ExternalRequestsManager contracts, treasury contracts, oracles, and token contracts, were not included in this audit scope.

No vulnerabilities were found.

Key notes and recommendations:

- **Token configuration restrictions**
  The protocol enforces restrictions on token conversions through manager configuration. The USR Manager allows minting for USDC, USDT, and RLP tokens, and not for USR itself. Similarly, the RLP Manager allows minting for USDC, USDT, and USR tokens, and not for RLP itself. This configuration prevents the creation of mint requests for USR in exchange for USR or RLP in exchange for RLP, eliminating the risk of circular conversions and potential inflation vectors.

- **Oracle price consistency between mint and burn**
  Exchange rate computation for `completeMint()` and `completeBurn()` is performed off-chain by the SERVICE_ROLE oracle. According to the client, the same pricing logic is used for both operations to ensure consistency and prevent arbitrage opportunities between the two paths.

- **Burn request processing and delta-neutral position management**
  Currently, the protocol processes burn requests with a 24-hour window for `completeBurn()`; the trading team manages position closures and asset purchases using optimal strategies for each request; and mint/burn operations are limited to USDC and USDT assets only. For direct USR ⇔ RLP conversions, closing delta-neutral positions is not required. Delta-neutral positions for USR and RLP are managed within a single pool. Mint/burn operations for USR ⇔ RLP are executed at fundamental prices.

- **Use idempotency keys for protocol token mint/burn**
  When `isProtocolToken` is true, `completeMint()` and `completeBurn()` call `mint(address,uint256)` and `burn(address,uint256)`, even though `ISimpleToken` exposes idempotent versions: `mint(bytes32,address,uint256)` and `burn(bytes32,address,uint256)`. The idempotent variants should be used so protocol-level mint and burn operations are properly tracked and cannot be replayed.

- **Access control and economic security**
  All addresses authorized for `requestMint()` and `requestBurn()` are whitelisted trusted providers with established agreements. Potential attacks involving large-scale mint or burn operations to destabilize the protocol or exploit pricing discrepancies are not economically viable due to reputational damage, legal consequences, and the requirement for capital comparable to market liquidity. The protocol operates with `completeMint()` in automatic mode with confirmation lag, while `completeBurn()` operates in manual mode with service within 24 hours.

1.3 Project Overview

Summary

Title	Description
Client	Resolv
Category	Basis Trading
Project	ExternalRequestsCoordinator
Type	Solidity
Platform	EVM
Timeline	30.12.2025 - 14.01.2026

Scope of Audit

File	Link
contracts/ ExternalRequestsCoordinator.sol	ExternalRequestsCoordinator.sol

Versions Log

Date	Commit Hash	Note
30.12.2025	dbee6f413138bf88e4900f29ace5d4ee7a3f8247	Initial Commit

Mainnet Deployments

File	Address	Blockchain
ExternalRequestsCoordinator.sol	0x36C9b5...c076c043	Ethereum

1.4 Security Assessment Methodology

Project Flow

Stage	Scope of Work
Interim audit	**Project Architecture Review:** · Review project documentation · Conduct a general code review · Perform reverse engineering to analyze the project's architecture based solely on the source code · Develop an independent perspective on the project's architecture · Identify any logical flaws in the design OBJECTIVE: UNDERSTAND THE OVERALL STRUCTURE OF THE PROJECT AND IDENTIFY POTENTIAL SECURITY RISKS.
	Code Review with a Hacker Mindset: · Each team member independently conducts a manual code review, focusing on identifying unique vulnerabilities. · Perform collaborative audits (pair auditing) of the most complex code sections, supervised by the Team Lead. · Develop Proof-of-Concepts (PoCs) and conduct fuzzing tests using tools like Foundry, Hardhat, and BOA to uncover intricate logical flaws. · Review test cases and in-code comments to identify potential weaknesses. OBJECTIVE: IDENTIFY AND ELIMINATE THE MAJORITY OF VULNERABILITIES, INCLUDING THOSE UNIQUE TO THE INDUSTRY.
	Code Review with a Nerd Mindset: · Conduct a manual code review using an internally maintained checklist, regularly updated with insights from past hacks, research, and client audits. · Utilize static analysis tools (e.g., Slither, Mythril) and vulnerability databases (e.g., Solodit) to uncover potential undetected attack vectors. OBJECTIVE: ENSURE COMPREHENSIVE COVERAGE OF ALL KNOWN ATTACK VECTORS DURING THE REVIEW PROCESS.

Stage	Scope of Work
	Consolidation of Auditors' Reports: · Cross-check findings among auditors · Discuss identified issues · Issue an interim audit report for client review OBJECTIVE: COMBINE INTERIM REPORTS FROM ALL AUDITORS INTO A SINGLE COMPREHENSIVE DOCUMENT.
Re-audit	**Bug Fixing & Re-Audit:** · The client addresses the identified issues and provides feedback · Auditors verify the fixes and update their statuses with supporting evidence · A re-audit report is generated and shared with the client OBJECTIVE: VALIDATE THE FIXES AND REASSESS THE CODE TO ENSURE ALL VULNERABILITIES ARE RESOLVED AND NO NEW VULNERABILITIES ARE ADDED.
Final audit	**Final Code Verification & Public Audit Report:** · Verify the final code version against recommendations and their statuses · Check deployed contracts for correct initialization parameters · Confirm that the deployed code matches the audited version · Issue a public audit report, published on our official GitHub repository · Announce the successful audit on our official X account OBJECTIVE: PERFORM A FINAL REVIEW AND ISSUE A PUBLIC REPORT DOCUMENTING THE AUDIT.

1.5 Risk Classification

Severity Level Matrix

Severity	Impact: High	Impact: Medium	Impact: Low
Likehood: High	Critical	High	Medium
Likehood: Medium	High	Medium	Low
Likehood: Low	Medium	Low	Low

Impact

- **High** – Theft from 0.5% OR partial/full blocking of funds (>0.5%) on the contract without the possibility of withdrawal OR loss of user funds (>1%) who interacted with the protocol.
- **Medium** – Contract lock that can only be fixed through a contract upgrade OR one-time theft of rewards or an amount up to 0.5% of the protocol's TVL OR funds lock with the possibility of withdrawal by an admin.
- **Low** – One-time contract lock that can be fixed by the administrator without a contract upgrade.

Likelihood

- **High** – The event has a 50–60% probability of occurring within a year and can be triggered by any actor (e.g., due to a likely market condition that the actor cannot influence).
- **Medium** – An unlikely event (10–20% probability of occurring) that can be triggered by a trusted actor.
- **Low** – A highly unlikely event that can only be triggered by the owner.

Action Required

- **Critical** – Must be fixed as soon as possible.
- **High** – Strongly advised to be fixed to minimize potential risks.
- **Medium** – Recommended to be fixed to enhance security and stability.
- **Low** – Recommended to be fixed to improve overall robustness and effectiveness.

Finding Status

- **Fixed** – The recommended fixes have been implemented in the project code and no longer impact its security.
- **Partially Fixed** – The recommended fixes have been partially implemented, reducing the impact of the finding, but it has not been fully resolved.
- **Acknowledged** – The recommended fixes have not yet been implemented, and the finding remains unresolved or does not require code changes.

1.6 Summary of Findings

Findings Count

Severity	Count
Critical	0
High	0
Medium	0
Low	0

2. Findings Report

2.1 Critical

Not Found

2.2 High

Not Found

2.3 Medium

Not Found

2.4 Low

Not Found

3. About MixBytes

MixBytes is a leading provider of smart contract audit and research services, helping blockchain projects enhance security and reliability. Since its inception, MixBytes has been committed to safeguarding the Web3 ecosystem by delivering rigorous security assessments and cutting-edge research tailored to DeFi projects.

Our team comprises highly skilled engineers, security experts, and blockchain researchers with deep expertise in formal verification, smart contract auditing, and protocol research. With proven experience in Web3, MixBytes combines in-depth technical knowledge with a proactive security-first approach.

Why MixBytes

- **Proven Track Record:** Trusted by top-tier blockchain projects like Lido, Aave, Curve, and others, MixBytes has successfully audited and secured billions in digital assets.
- **Technical Expertise:** Our auditors and researchers hold advanced degrees in cryptography, cybersecurity, and distributed systems.
- **Innovative Research:** Our team actively contributes to blockchain security research, sharing knowledge with the community.

Our Services

- **Smart Contract Audits:** A meticulous security assessment of DeFi protocols to prevent vulnerabilities before deployment.
- **Blockchain Research:** In-depth technical research and security modeling for Web3 projects.
- **Custom Security Solutions:** Tailored security frameworks for complex decentralized applications and blockchain ecosystems.

MixBytes is dedicated to securing the future of blockchain technology by delivering unparalleled security expertise and research-driven solutions. Whether you are launching a DeFi protocol or developing an innovative dApp, we are your trusted security partner.

Contact Information

https://mixbytes.io/

https://github.com/mixbytes/audits_public

hello@mixbytes.io

https://x.com/mixbytes