
Security Review Report

NM-1013 - Assetera



NETHERMIND
SECURITY

(September 08, 2026)

Contents

1	Executive Summary	2
2	Audited Files	3
3	Summary of Issues	3
4	System Overview	4
5	Risk Rating Methodology	5
6	Issues	6
7	Documentation Evaluation	7
8	Test Suite Evaluation	8
8.1	Tests Output	8
9	About Nethermind	13

1 Executive Summary

This document presents the results of the security review conducted by [Nethermind Security](#) for Assetera. The focus of the engagement has been on the primary-market settlement router that executes a buyer's first acquisition of a regulated real-world asset.

The protocol settles one signed intent per purchase through an authorized settlement operator, cross-signed by the buyer over the same digest, alongside independent compliance and fee attestations. The router pulls an exact, single-transaction amount of settlement currency from the buyer, approves an address named in the signed intent (a third-party venue or an issuer-specific sale contract) and hands it opaque calldata, judging the result purely on the balance deltas it measures: refunding what the venue did not consume, forwarding to the buyer any asset that instead landed on the router itself, and asserting its own settlement-currency and asset-token balances both return to their pre-call values.

There is no on-chain allowlist of venues, assets, or currencies; a per-transaction cap on the full authorized debit stands in its place, and the compliance gate is fail-closed by construction. The system therefore combines buyer-signed consent with privileged, unallowlisted execution, measured-delta accounting in place of a trusted venue interface.

The audit comprises 382 lines of Solidity code. The audit was performed using (a) manual analysis of the codebase, and (b) automated analysis tools. **Along this document, we report** zero points of attention, as shown in Fig. 1.

This document is organized as follows. Section 2 presents the files in the scope. Section 3 presents the summary of issues. Section 4 presents the System Overview. Section 5 discusses the risk rating methodology. Section 6 details the issues. Section 7 discusses the documentation provided by the client for this audit. Section 8 presents the test suite evaluation and automated tools used. Section 9 concludes the document.

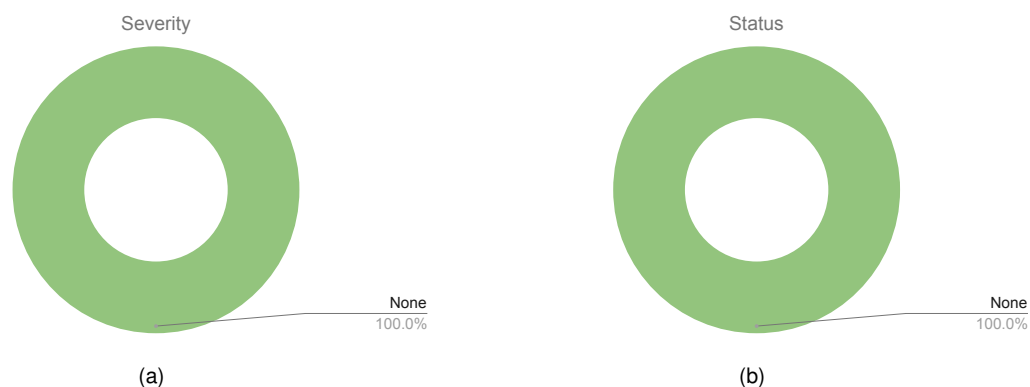


Fig. 1: Distribution of issues: Critical (0), High (0), Medium (0), Low (0), Undetermined (0), Informational (0), Best Practices (0).
Distribution of status: Fixed (0), Acknowledged (0), Mitigated (0), Unresolved (0)

Summary of the Audit

Audit Type	Security Review
Final Report	September 08, 2026
Initial Commit	653a8a6
Final Commit	653a8a6
Documentation Assessment	High
Test Suite Assessment	High

2 Audited Files

	Contract	LoC	Comments	Ratio	Blank	Total
1	src/primary/IntentGate.sol	54	160	296%	14	228
2	src/primary/AsseteraPrimarySales.sol	106	321	302%	30	457
3	src/primary/types/PrimaryTypes.sol	37	127	343%	8	169
4	src/primary/admin/SettlementLimits.sol	52	169	325%	18	239
5	src/primary/storage/PrimaryStorage.sol	24	90	375%	9	123
6	src/primary/settle/VenueSettler.sol	60	332	553%	23	415
7	src/primary/interfaces/ISettler.sol	20	110	550%	6	136
8	src/primary/interfaces/ISettlementLimits.sol	10	90	900%	5	105
9	src/primary/interfaces/IntentGate.sol	19	80	421%	5	104
	Total	382	1479	387.2%	118	1976

3 Summary of Issues

No issues were found.

4 System Overview

AsseteraPrimarySales is a UUPS proxy that executes a buyer's first (primary-market) acquisition of a real-world asset. It implements exactly one settlement family: VenueSettler pulls the buyer's settlement currency, hands opaque calldata to an address named in a signed intent (intent.venue), and judges the outcome exclusively on balance deltas it measures itself before and after that call—never on anything the venue reports back.

Entry point and guard rails. The only state-changing, fund-moving entry point is settlePrimary(venueCalldata, intent, intentSignature, buyerSignature, kyc, fee), guarded by whenNotPaused and nonReentrant. It is reachable directly or through permitAndCall (src/core/PermitRelay.sol, shared bytecode with the exchange), which runs an ERC-2612 permit for the caller and then self-delegates calls into the same frozen settlePrimary selector, re-appending the ERC-2771 sender suffix so both paths resolve the identical _msgSender().

Four signatures, three nonce namespaces. Every settlement requires four independent signatures, all verified before a single nonce is burned:

- the **settlement intent**, signed by SETTLEMENT_OPERATOR_ROLE and recovered via ECDSA.recover (IntentGate._verifyIntent)—the only Assetera-side key whose holder can cause a transfer;
- the **buyer's own consent**, over the identical EIP-712 digest, verified via SignatureChecker.isValidSignatureNow (EOA or ERC-1271) rather than plain recovery, so Safe and smart-account buyers are supported;
- the **KYC attestation**, signed by KYC_OPERATOR_ROLE and verified by the shared KycGate._verifyKyc;
- the **fee attestation**, signed by FEE_OPERATOR_ROLE and verified by the shared FeeGate._verifyFee, cross-checked so feeAtt.feeToken can only equal intent.settlementToken and feeAtt.paramsHash/kycAtt.paramsHash are transitively equal through the intent's own struct hash.

Nonces (usedIntentNonce, the KYC gate's usedNonce, usedFeeNonce) burn only after all four signatures verify. The compliance gate is inverted on this proxy specifically: complianceRequired reads an *exemption* (complianceExempt) out of the router's own namespace rather than the shared, fail-open mapping the exchange reads directly—so every action ordinal, declared or not, is gated by default rather than by enumeration.

The money path (VenueSettler._settleVenue). After SettlementLimits._authorizeSettlement binds both attestations to the intent's struct hash, burns all three nonces, and charges the per-transaction cap against the *full* authorized debit (venueQuoteIn + buyerFee, charged before the venue is ever called), _settleVenue runs a fixed sequence: snapshot the router's own currency and asset balances plus the buyer's asset balance; pull exactly the debit and measure the router's balance grew by exactly that amount (SettlementPullMismatch otherwise); approve the venue exactly venueQuoteIn—never the fee—and call it with hash-and-selector-bound opaque calldata; revoke the approval unconditionally; measure real consumption and refund the remainder; forward any asset that landed on the router to the buyer and assert measured delivery is at least intent.minAssetOut; pay the fee; and assert both token balances returned exactly to their pre-call values. Both legs are held to the same measured-delta discipline—a fix that closed a gap where an earlier revision measured only the asset side.

Asset accounting: two supported modes. intent.accountingMode (a uint8, since EIP-712 has no enum type) selects Erc20Balance (plain balanceOf/transfer) or RebasingShares (sharesOf/transferShares/getUnderlyingAmountByShares, for share-accounted tokens such as Backed/xStocks). The three dispatch helpers consistently gate on the same validated mode; an out-of-range or mismatched mode fails closed with a named revert rather than mis-measuring. The settlement currency has no such mode—it is always measured as a plain ERC-20, so a genuinely rebasing or fee-on-transfer currency is refused via the same measured-delta checks rather than an explicit type check.

Value caps, not an allowlist. There is no on-chain allowlist of venues, selectors, assets, or currencies. What bounds a compromised settlement-operator key is the buyer's own signature over the same intent, an allowance to the router that is always exact and single-transaction, and SettlementLimits—a per-settlement-token, per-transaction cap on the full authorized debit, where an unconfigured token (cap == 0) cannot settle at all.

Own issuance—AsseteraIssuanceVenue. The router holds no minting right on any path. Its own primary issuance routes through AsseteraIssuanceVenue, a per-offering contract called exactly as any third-party venue, immutably bound to one router address, that pulls and measures settlement currency *before* minting, and measures the mint before reporting success. This pull-before-mint ordering, combined with the router's ZeroVenueQuote guard and its exact allowance to the venue, is what the design leans on to argue that a compromised settlement-operator key cannot mint without genuine payment—traced through both contracts in this review and found to hold structurally. This contract is not among the files the current scope document lists as in-scope, and the document's own claim that it is “not yet built outside this repository” no longer matches this worktree.

One-transaction purchase. permitAndCall lets a buyer submit an ERC-2612 permit and the settlement in a single transaction. Permit failure is swallowed for three legitimate cases (no ERC-2612 support, a domain mismatch, an already-consumed permit) and falls back to a pre-existing allowance; a codeless token address is the one failure that is not swallowed. The relay itself carries no reentrancy guard—every function it can delegate into, settlePrimary included, carries its own—and is not payable, so there is no msg.value to double-spend across the delegated call.

5 Risk Rating Methodology

The risk rating methodology used by [Nethermind Security](#) follows the principles established by the [OWASP Foundation](#). The severity of each finding is determined by two factors: **Likelihood** and **Impact**.

Likelihood measures how likely the finding is to be uncovered and exploited by an attacker. This factor will be one of the following values:

- a) **High**: The issue is trivial to exploit and has no specific conditions that need to be met;
- b) **Medium**: The issue is moderately complex and may have some conditions that need to be met;
- c) **Low**: The issue is very complex and requires very specific conditions to be met.

When defining the likelihood of a finding, other factors are also considered. These can include but are not limited to motive, opportunity, exploit accessibility, ease of discovery, and ease of exploit.

Impact is a measure of the damage that may be caused if an attacker exploits the finding. This factor will be one of the following values:

- a) **High**: The issue can cause significant damage, such as loss of funds or the protocol entering an unrecoverable state;
- b) **Medium**: The issue can cause moderate damage, such as impacts that only affect a small group of users or only a particular part of the protocol;
- c) **Low**: The issue can cause little to no damage, such as bugs that are easily recoverable or cause unexpected interactions that cause minor inconveniences.

When defining the impact of a finding, other factors are also considered. These can include but are not limited to Data/state integrity, loss of availability, financial loss, and reputation damage. After defining the likelihood and impact of an issue, the severity can be determined according to the table below.

		Severity Risk		
Impact	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Info/Best Practices	Low	Medium
	Undetermined	Undetermined	Undetermined	Undetermined
		Low	Medium	High
		Likelihood		

To address issues that do not fit a High/Medium/Low severity, [Nethermind Security](#) also uses three more finding severities: **Informational**, **Best Practices**, and **Undetermined**.

- a) **Informational** findings do not pose any risk to the application, but they carry some information that the audit team intends to pass to the client formally;
- b) **Best Practice** findings are used when some piece of code does not conform with smart contract development best practices;
- c) **Undetermined** findings are used when we cannot predict the impact or likelihood of the issue.

6 Issues

No issues were found

7 Documentation Evaluation

Software documentation refers to the written or visual information that describes the functionality, architecture, design, and implementation of software. It provides a comprehensive overview of the software system and helps users, developers, and stakeholders understand how the software works, how to use it, and how to maintain it. Software documentation can take different forms, such as user manuals, system manuals, technical specifications, requirements documents, design documents, and code comments. Software documentation is critical in software development, enabling effective communication between developers, testers, users, and other stakeholders. It helps to ensure that everyone involved in the development process has a shared understanding of the software system and its functionality. Moreover, software documentation can improve software maintenance by providing a clear and complete understanding of the software system, making it easier for developers to maintain, modify, and update the software over time. Smart contracts can use various types of software documentation. Some of the most common types include:

- Technical whitepaper: A technical whitepaper is a comprehensive document describing the smart contract's design and technical details. It includes information about the purpose of the contract, its architecture, its components, and how they interact with each other;
- User manual: A user manual is a document that provides information about how to use the smart contract. It includes step-by-step instructions on how to perform various tasks and explains the different features and functionalities of the contract;
- Code documentation: Code documentation is a document that provides details about the code of the smart contract. It includes information about the functions, variables, and classes used in the code, as well as explanations of how they work;
- API documentation: API documentation is a document that provides information about the API (Application Programming Interface) of the smart contract. It includes details about the methods, parameters, and responses that can be used to interact with the contract;
- Testing documentation: Testing documentation is a document that provides information about how the smart contract was tested. It includes details about the test cases that were used, the results of the tests, and any issues that were identified during testing;
- Audit documentation: Audit documentation includes reports, notes, and other materials related to the security audit of the smart contract. This type of documentation is critical in ensuring that the smart contract is secure and free from vulnerabilities.

These types of documentation are essential for smart contract development and maintenance. They help ensure that the contract is properly designed, implemented, and tested, and they provide a reference for developers who need to modify or maintain the contract in the future.

Remarks about Assetera Primary Markets's documentation

The documentation for this audit consisted primarily of inline NatSpec comments within the contracts. Assetera development team addressed all questions and concerns raised by the Nethermind Security team, providing valuable insights and an in-depth understanding of the project's technical aspects.

8 Test Suite Evaluation

8.1 Tests Output

```

No files changed, compilation skipped

Ran 1 test for test/primary/AaplMainnetFork.t.sol:AaplMainnetForkTest
[SKIP: skipped] setUp() (gas: 0)
Suite result: ok. 0 passed; 0 failed; 1 skipped; finished in 7.31ms (0.00ns CPU time)

Ran 5 tests for test/primary/SettlementLimits.t.sol:SettlementCapEnforcementTest
[PASS] test_Consume_AcceptsADebitExactlyAtTheCap() (gas: 13505)
[PASS] test_Consume_RejectsADebitOneAboveTheCap() (gas: 16779)
[PASS] test_Consume_RejectsATokenWhoseCapWasSetBackToZero() (gas: 24265)
[PASS] test_Consume_RejectsAZeroDebitOnAnUnconfiguredToken() (gas: 16866)
[PASS] test_Consume_RejectsAnUnconfiguredToken() (gas: 16578)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 7.91ms (1.21ms CPU time)

Ran 4 tests for test/primary/PrimaryIntentVectors.t.sol:PrimaryIntentVectorsTest
[PASS] test_IntentTypehash_IsPinned() (gas: 13723)
[PASS] test_ParamsHash_ChangesWithEveryField() (gas: 48858)
[PASS] test_ParamsHash_EqualsTheCanonicalEip712StructHash() (gas: 20232)
[PASS] test_PrimarySettledEventSignature_IsPinned() (gas: 360)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 8.26ms (1.64ms CPU time)

Ran 2 tests for test/primary/SettlementLimits.t.sol:SettlementCapStorageTest
[PASS] test_SettingACapDoesNotTouchTheLowLinearSlots() (gas: 39975)
[PASS] test_TheCapStateLandsAtItsAppendedOffsetsInsideTheNamespace() (gas: 36127)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 1.54ms (470.04µs CPU time)

Ran 4 tests for test/primary/AsseteraPrimarySales.t.sol:PrimarySalesDomainTest
[PASS] test_Domain_DiffersFromTheExchanges() (gas: 5344423)
[PASS] test_Domain_IsAsseteraPrimarySalesVersionOne() (gas: 26690)
[PASS] test_ExchangeAttestation_CannotBeReplayedHere() (gas: 5400833)
[PASS] test_TheSameAttestationUnderThisDomainIsAccepted() (gas: 171704)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 11.68ms (5.14ms CPU time)

Ran 2 tests for test/primary/VenueSettler.t.sol:VenueSettlerRefundTest
[PASS] test_SettleVenue_RefundsTheWholeQuoteWhenTheVenueConsumesNothing() (gas: 465251)
[PASS] test_SettleVenue_RefundsWhatTheVenueDidNotConsume() (gas: 513902)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 12.69ms (6.05ms CPU time)

Ran 14 tests for test/primary/AsseteraPrimarySales.t.sol:PrimarySalesAdminTest
[PASS] test_ABareValueTransferToTheRouterStillReverts() (gas: 23425)
[PASS] test_DisablingTheKycGateDoesNotDisableTheIntentGate() (gas: 97503)
[PASS] test_Pause_OnlyAdmin() (gas: 20692)
[PASS] test_SetAllowedCollector_OnlyAdmin() (gas: 21028)
[PASS] test_SetAllowedCollector_TogglesAndEmits() (gas: 39975)
[PASS] test_SetComplianceRequired_OnlyAdmin() (gas: 21164)
[PASS] test_SetComplianceRequired_TogglesAndEmits() (gas: 46142)
[PASS] test_Unpause_RestoresSettlement() (gas: 183053)
[PASS] test_Upgrade_OnlyAdmin() (gas: 3459115)
[PASS] test_WhitelistHandshake_ForwardsTheFullValueToTheDestination() (gas: 66256)
[PASS] test_WhitelistHandshake_LeavesTheRouterWithNoNativeBalance() (gas: 63464)
[PASS] test_WhitelistHandshake_OnlyAdmin() (gas: 29694)
[PASS] test_WhitelistHandshake_RejectsTheZeroDestination() (gas: 26177)
[PASS] test_WhitelistHandshake_RevertsWhenTheDestinationRejectsTheTransfer() (gas: 73790)
Suite result: ok. 14 passed; 0 failed; 0 skipped; finished in 13.02ms (6.72ms CPU time)

Ran 4 tests for test/primary/SettlementLimits.t.sol:SettlementCapThroughTheEntryPointTest
[PASS] test_Settlement_BurnsNoNonceWhenTheCapRefusesIt() (gas: 194757)
[PASS] test_Settlement_IsChargedByTheSharedPreambleNotByTheFamily() (gas: 192783)
[PASS] test_Settlement_RefusesACurrencyWithNoCap() (gas: 177738)
[PASS] test_Settlement_SucceedsWithinTheCap() (gas: 173716)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 5.32ms (4.24ms CPU time)

Ran 3 tests for test/primary/PrimarySalesAdversarial.t.sol:PrimarySalesBuyerConsentReplayTest
[PASS] test_BuyerConsent_AMalleatedSignatureIsRefused() (gas: 104184)
[PASS] test_BuyerConsent_ASignatureCapturedFromOneSettlementDoesNotAuthoriseAnother() (gas: 110843)
[PASS] test_BuyerConsent_ASignatureForOneProxyDoesNotAuthoriseTheSameTermsOnAnother() (gas: 3817708)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 4.91ms (3.63ms CPU time)

```

```
Ran 7 tests for test/primary/VenueSettler.t.sol:VenueSettlerFeeVectorsTest
[PASS] test_AssertBuyerFee_AcceptsTheDerivedFee() (gas: 25496)
[PASS] test_AssertBuyerFee_RejectsAFeeOneWeiBelowTheDerivedOne() (gas: 28852)
[PASS] test_AssertBuyerFee_RejectsAFeeTheSettlementSignerInvented() (gas: 29028)
[PASS] test_ExpectedBuyerFee_FloorsExactlyAsEveryOtherFeeInTheSystemDoes() (gas: 12943)
[PASS] test_ExpectedBuyerFee_IsZeroWhenNoFeeIsAttested() (gas: 12279)
[PASS] test_ExpectedBuyerFee_IsZeroWhenTheFeeCannotReachOneWei() (gas: 11043)
[PASS] test_ExpectedBuyerFee_MatchesTheExactVector() (gas: 10897)
Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 1.22ms (104.04µs CPU time)

Ran 5 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerCapBoundaryTest
[PASS] test_Cap_ADebitExactlyAtTheCapSettles() (gas: 381382)
[PASS] test_Cap_ADebitOneWeiOverTheCapIsRefusedAndNothingMoves() (gas: 273784)
[PASS] test_Cap_ADecimalsChangeSilentlyRepricesTheStoredCap() (gas: 393925)
[PASS] test_Cap_IsChargedOnTheFullAuthorisedDebitNotOnTheNetOne() (gas: 260446)
[PASS] test_Cap_ResettingTheCapAfterADecimalsChangeRestoresTheIntendedBound() (gas: 277490)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 16.36ms (9.55ms CPU time)

Ran 3 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerAssetResidueTest
[PASS] test_AssetResidue_AForwardThatMovedLessThanItWasAskedToIsRefused() (gas: 498587)
[PASS] test_AssetResidue_ASingleWeiLeftBehindIsEnoughToRefuseTheSettlement() (gas: 486177)
[PASS] test_AssetResidue_TheSameSettlementGoesThroughWhenTheForwardMovesInFull() (gas: 404629)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 2.97ms (1.77ms CPU time)

Ran 10 tests for test/primary/AsseteraPrimarySales.t.sol:PrimarySalesBuyerConsentTest
[PASS] test_BuyerConsent_AcceptsAnErc1271ContractWallet() (gas: 372765)
[PASS] test_BuyerConsent_AcceptsTheBuyersSignatureOverTheSameIntent() (gas: 176733)
[PASS] test_BuyerConsent_RejectsABuyerContractThatIsNotAWallet() (gas: 82416)
[PASS] test_BuyerConsent_RejectsAMalformedSignature() (gas: 67786)
[PASS] test_BuyerConsent_RejectsARevokedErc1271Signature() (gas: 287912)
[PASS] test_BuyerConsent_RejectsASignatureFromTheWrongKey() (gas: 72232)
[PASS] test_BuyerConsent_RejectsASignatureOverADifferentIntent() (gas: 74780)
[PASS] test_BuyerConsent_RejectsAnEmptySignature() (gas: 64362)
[PASS] test_BuyerConsent_RejectsTheBuyersSignatureInTheOperatorSlot() (gas: 66712)
[PASS] test_BuyerConsent_RejectsTheOperatorsSignatureInTheBuyerSlot() (gas: 72167)
Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 16.88ms (9.47ms CPU time)

Ran 6 tests for test/primary/SettlementLimits.t.sol:SettlementCapAdminTest
[PASS] test_Cap_ReadsZeroForAnUnconfiguredToken() (gas: 16953)
[PASS] test_SetSettlementCap_ClosesATokenWithoutReadingItsDecimals() (gas: 34724)
[PASS] test_SetSettlementCap_RejectsANonAdminCaller() (gas: 21337)
[PASS] test_SetSettlementCap_RejectsTheSettlementOperator() (gas: 21284)
[PASS] test_SetSettlementCap_RejectsTheZeroToken() (gas: 19224)
[PASS] test_SetSettlementCap_StoresRawAndWholeAndEmits() (gas: 40539)
Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 2.02ms (308.17µs CPU time)

Ran 4 tests for test/primary/AsseteraPrimarySales.t.sol:PrimarySalesSettledTest
[PASS] test_SettlePrimary_BurnsAllThreeNoncesInSeparateNamespaces() (gas: 192622)
[PASS] test_SettlePrimary_EmitsPrimarySettledFromTheMeasuredResult() (gas: 183202)
[PASS] test_SettlePrimary_RejectsAReplayedIntentNonce() (gas: 197700)
[PASS] test_SettlePrimary_ResolverTheBuyerThroughTheTrustedForwarder() (gas: 225591)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 3.84ms (3.02ms CPU time)

Ran 7 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerReentrancyTest
[PASS] test_Reentrancy_ASecondSettlementFromInsideTheVenueCallIsRefused() (gas: 492290)
[PASS] test_Reentrancy_AStandingBuyerAllowanceIsNotReachableFromAVenueCallback() (gas: 488732)
[PASS] test_Reentrancy_AVenueCannotAllowlistItselfAsAFeeCollector() (gas: 493540)
[PASS] test_Reentrancy_AVenueCannotPauseOrUnpauseTheRouter() (gas: 493392)
[PASS] test_Reentrancy_AVenueCannotRaiseTheSettlementCap() (gas: 493984)
[PASS] test_Reentrancy_AVenueCannotTriggerTheWhitelistHandshake() (gas: 491288)
[PASS] test_Reentrancy_TheIntentNonceIsAlreadyBurnedWhenTheVenueRuns() (gas: 465324)
Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 6.99ms (5.77ms CPU time)

Ran 3 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerSilentTransferTest
[PASS] test_SilentTransfer_AFeeTransferThatMovedNothingIsCaught() (gas: 450439)
[PASS] test_SilentTransfer_ARefundThatMovedNothingIsCaught() (gas: 451647)
[PASS] test_SilentTransfer_TheSameSettlementGoesThroughWhenTransfersActuallyMove() (gas: 404021)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 2.90ms (1.73ms CPU time)

Ran 12 tests for test/primary/PrimarySalesAdversarial.t.sol:PrimarySalesReplayTest
[PASS] test_Replay_AFeeAttestationCannotBeLiftedOntoADifferentIntent() (gas: 114070)
[PASS] test_Replay_AFeeNonceCannotBeSpentTwiceEvenForADifferentSettlement() (gas: 581290)
[PASS] test_Replay_AFreshBuyerSignatureDoesNotReviveASpentIntent() (gas: 553613)
[PASS] test_Replay_AKycAttestationCannotBeLiftedOntoADifferentIntent() (gas: 114284)
[PASS] test_Replay_AKycNonceCannotBeSpentTwiceEvenForADifferentSettlement() (gas: 571886)
```

```
[PASS] test_Replay_APrimaryAttestationIsRefusedByTheExchange() (gas: 5379200)
[PASS] test_Replay_ASecondSettlementWithFreshNoncesSucceeds() (gas: 675325)
[PASS] test_Replay_ASettlementAssembledForOneProxyIsRefusedByAnother() (gas: 3815894)
[PASS] test_Replay_ASettlementSignedOnOneChainIsRefusedOnAnother() (gas: 105386)
[PASS] test_Replay_TheSameAttestationUnderTheExchangeDomainIsAccepted() (gas: 5722630)
[PASS] test_Replay_TheSamePayloadsOnTheSigningChainSettle() (gas: 506551)
[PASS] test_Replay_TheSameSettlementTwiceIsRefusedAndDebitsTheBuyerOnce() (gas: 559157)
Suite result: ok. 12 passed; 0 failed; 0 skipped; finished in 10.23ms (8.41ms CPU time)

Ran 5 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerFeeOnTransferTest
[PASS] test_FeeOnTransfer_CannotCreditTheCollectorLessThanTheAttestedFee() (gas: 329862)
[PASS] test_FeeOnTransfer_CannotReportTheBurnAsVenueConsumption() (gas: 343085)
[PASS] test_FeeOnTransfer_LeavesTheBuyerAllowanceIntact() (gas: 331350)
[PASS] test_FeeOnTransfer_RefusesEvenWhenTheVenueWouldTakeLessThanTheRouterHolds() (gas: 354992)
[PASS] test_FeeOnTransfer_RefusesTheSettlementAtThePull() (gas: 336896)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 4.02ms (2.79ms CPU time)

Ran 2 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerSenderSurchargeTest
[PASS] test_SenderSurcharge_CannotSettleAtAllBecauseEveryOutflowCostsMoreThanItMoves() (gas: 454948)
[PASS] test_SenderSurcharge_TheLowerBoundCatchesARouterLeftBelowItsOwnFee() (gas: 444387)
Suite result: ok. 2 passed; 0 failed; 0 skipped; finished in 2.35ms (1.17ms CPU time)

Ran 8 tests for test/primary/VenueSettler.t.sol:VenueSettlerRevertTest
[PASS] test_SettleVenue_ApprovesExactlyTheQuoteSoAGreedierVenueFails() (gas: 457489)
[PASS] test_SettleVenue_DoesNotApproveTheFeeToTheVenue() (gas: 457599)
[PASS] test_SettleVenue_RefusesABuyerFeeInventedForAnUnlistedCollector() (gas: 274222)
[PASS] test_SettleVenue_RevertsWhenTheVenueDeliversLessThanMinAssetOut() (gas: 534760)
[PASS] test_SettleVenue_RevertsWhenTheVenueDeliversNothing() (gas: 469735)
[PASS] test_SettleVenue_RevertsWhenTheVenueIsTheAssetToken() (gas: 275467)
[PASS] test_SettleVenue_RevertsWhenTheVenueIsTheSettlementToken() (gas: 275303)
[PASS] test_SettleVenue_RevertsWhenTheVenueRejectsTheOrder() (gas: 366982)
Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 8.68ms (6.39ms CPU time)

Ran 3 tests for test/primary/SettlementLimits.t.sol:SettlementCapAppliesToEveryFamilyTest
[PASS] test_AFamilyThatChargesNothingIsStillCharged() (gas: 169104)
[PASS] test_TheFamilyIsChargedTheFullAuthorisedDebit() (gas: 182386)
[PASS] test_TheSameFamilyReachesItsOwnBodyOnceTheCapIsOpen() (gas: 161159)
Suite result: ok. 3 passed; 0 failed; 0 skipped; finished in 4.49ms (2.78ms CPU time)

Ran 5 tests for test/primary/PrimaryStorageNamespace.t.sol:PrimaryStorageNamespaceTest
[PASS] test_NamespaceConstants_AreTheErc7201DerivationsOfTheirStrings() (gas: 1702)
[PASS] test_PrimaryNamespace_DoesNotCollideWithTheGateNamespace() (gas: 685)
[PASS] test_SettlementDoesNotTouchTheLowLinearSlots() (gas: 177305)
[PASS] test_TheDerivedMappingSlotsDoNotOverlap() (gas: 2031)
[PASS] test_UsedIntentNonce_WritesInsideTheNamespace() (gas: 171213)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 1.88ms (1.04ms CPU time)

Ran 6 tests for test/primary/PrimarySalesAdversarial.t.sol:PrimarySalesHostileWalletTest
[PASS] test_HostileWallet_APermissiveWalletIsAcceptedAndThatIsTheTrustBoundary() (gas: 603759)
[PASS] test_HostileWallet_ARevertingWalletIsRefused() (gas: 219058)
[PASS] test_HostileWallet_AShortReplyIsRefused() (gas: 190527)
[PASS] test_HostileWallet_ATruthyNonMagicAnswerIsRefused() (gas: 215253)
[PASS] test_HostileWallet_AWrongMagicValueIsRefused() (gas: 216728)
[PASS] test_HostileWallet_TheHonestWalletStillSettles() (gas: 731115)
Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 9.23ms (6.90ms CPU time)

Ran 7 tests for test/primary/SettlementLimits.t.sol:SettlementCapDecimalsTest
[PASS] test_ADecimalsBugIsCaughtEvenByAVeryGenerousCap() (gas: 39280)
[PASS] test_ATokenThatDoesNotReportDecimalsCannotBeCapped() (gas: 20684)
[PASS] test_AnAddressWithNoCodeCannotBeCapped() (gas: 24022)
[PASS] test_ImplausibleDecimalsAreRefused() (gas: 22331)
[PASS] test_TheBoundaryIsExactOnASixDecimalCurrency() (gas: 39275)
[PASS] test_TheBoundaryIsExactOnAnEighteenDecimalCurrency() (gas: 73265)
[PASS] test_TheSameWholeCapMeansDifferentRawCapsPerCurrency() (gas: 93622)
Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 1.19ms (192.13µs CPU time)

Ran 8 tests for test/primary/VenueSettler.t.sol:VenueSettlerHappyPathTest
[PASS] test_SettleVenue_DebitsTheBuyerDeliversTheAssetAndPaysTheFee() (gas: 512436)
[PASS] test_SettleVenue_EmitsPrimarySettledWithTheMeasuredNumbers() (gas: 512834)
[PASS] test_SettleVenue_HandlesAZeroFeeSettlement() (gas: 478998)
[PASS] test_SettleVenue_LeavesNoStandingApprovalToTheVenue() (gas: 506429)
[PASS] test_SettleVenue_LeavesThePreExistingRouterBalanceExactlyWhereItWas() (gas: 534136)
[PASS] test_SettleVenue_RevertsWhenTheBuyerAllowanceIsOneWeiShort() (gas: 287748)
[PASS] test_SettleVenue_RevokesTheApprovalEvenWhenTheVenueConsumedLessThanTheQuote() (gas: 509398)
[PASS] test_SettleVenue_SpendsTheBuyersExactAllowanceAndLeavesNoneStanding() (gas: 506297)
```

Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 24.29ms (17.61ms CPU time)

Ran 4 tests for test/primary/VenueSettler.t.sol:VenueSettlerLimitsTest

[PASS] test_SettleVenue_ChargesTheCapsOnceWithTheFullDebit() (gas: 509395)
 [PASS] test_SettleVenue_FailsClosedWhenTheSettlementCurrencyHasNoCap() (gas: 222321)
 [PASS] test_SettleVenue_RevertsAndMovesNothingWhenThePerTxCapIsExceeded() (gas: 304209)
 [PASS] test_SettleVenue_SucceedsOnceTheAdminSizesTheCurrency() (gas: 493139)
 Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 3.46ms (2.21ms CPU time)

Ran 8 tests for test/primary/PrimarySalesAdversarial.t.sol:PrimarySalesMetaTxTest

[PASS] test_MetaTx_ARelayedCallCannotSpoofTheBuyerThroughTheRelayedData() (gas: 150744)
 [PASS] test_MetaTx_ARelayedSettlementDebitsTheBuyerAndNotTheRelayer() (gas: 558694)
 [PASS] test_MetaTx_AppendedSenderBytesOnADirectCallAreIgnored() (gas: 103486)
 [PASS] test_MetaTx_TheCalldataOverrideDoesNotUnderflowOnACallShorterThanASuffix() (gas: 18631)
 [PASS] test_MetaTx_TheCalldataOverrideLeavesAnUntrustedCallersBytesAlone() (gas: 20712)
 [PASS] test_MetaTx_TheCalldataOverrideStripsTheForwarderSuffix() (gas: 20836)
 [PASS] test_MetaTx_TheForwarderRefusesARequestSignedByAnyoneButTheClaimedSender() (gas: 114618)
 [PASS] test_MetaTx_TheSamePayloadFromTheBuyerDirectlySettles() (gas: 505814)
 Suite result: ok. 8 passed; 0 failed; 0 skipped; finished in 5.00ms (4.54ms CPU time)

Ran 4 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerVenueIdentityTest

[PASS] test_VenueIdentity_AVenueCannotSpendATokenTheRouterNeverApproved() (gas: 393212)
 [PASS] test_VenueIdentity_NoAllowanceOnAnyTokenSurvivesASettlement() (gas: 416350)
 [PASS] test_VenueIdentity_TheRouterAsItsOwnVenueCannotReachItsOwnAdminSurface() (gas: 349259)
 [PASS] test_VenueIdentity_TheRouterAsItsOwnVenueCannotSettleAgain() (gas: 362616)
 Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 7.04ms (5.32ms CPU time)

Ran 24 tests for test/primary/AsseteraPrimarySales.t.sol:PrimarySalesEntryPointTest

[PASS] test_SettlePrimary_ReachesTheMoneyPathWithEveryGatePassed() (gas: 170933)
 [PASS] test_SettlePrimary_RejectsACallerWhoIsNotTheBuyer() (gas: 54874)
 [PASS] test_SettlePrimary_RejectsACollectorThatDisagreesWithTheIntent() (gas: 105824)
 [PASS] test_SettlePrimary_RejectsAFeeDenominatedInTheAssetToken() (gas: 74048)
 [PASS] test_SettlePrimary_RejectsAFeeParamsHashThatIsNotTheIntentHash() (gas: 73420)
 [PASS] test_SettlePrimary_RejectsAKycParamsHashThatIsNotTheIntentHash() (gas: 73527)
 [PASS] test_SettlePrimary_RejectsANonZeroMakerFee() (gas: 73099)
 [PASS] test_SettlePrimary_RejectsANonZeroMakerFeeBeforeTheCollectorChecks() (gas: 74968)
 [PASS] test_SettlePrimary_RejectsANonZeroOrderIdOnTheKycAttestation() (gas: 79014)
 [PASS] test_SettlePrimary_RejectsASelectorThatDoesNotMatchTheCalldata() (gas: 73755)
 [PASS] test_SettlePrimary_RejectsAZeroMinAssetOut() (gas: 53245)
 [PASS] test_SettlePrimary_RejectsAZeroVenue() (gas: 52988)
 [PASS] test_SettlePrimary_RejectsAZeroVenueQuote() (gas: 53656)
 [PASS] test_SettlePrimary_RejectsAZeroVenueQuoteEvenWithANonZeroFee() (gas: 53091)
 [PASS] test_SettlePrimary_RejectsAnAssetLegEqualToTheSettlementLeg() (gas: 53147)
 [PASS] test_SettlePrimary_RejectsAnAttestationBoundToADifferentIntent() (gas: 71496)
 [PASS] test_SettlePrimary_RejectsAnExpiredIntent() (gas: 54512)
 [PASS] test_SettlePrimary_RejectsAnIntentSignedByTheFeeOperator() (gas: 66858)
 [PASS] test_SettlePrimary_RejectsAnIntentSignedByTheKycOperator() (gas: 66241)
 [PASS] test_SettlePrimary_RejectsAnOverLongIntentTtl() (gas: 54623)
 [PASS] test_SettlePrimary_RejectsAnUnlistedCollector() (gas: 79277)
 [PASS] test_SettlePrimary_RejectsCalldataThatDoesNotHashToTheSignedValue() (gas: 72930)
 [PASS] test_SettlePrimary_RejectsMaxSettlementBelowQuotePlusFee() (gas: 53695)
 [PASS] test_SettlePrimary_RejectsWhenPaused() (gas: 75055)
 Suite result: ok. 24 passed; 0 failed; 0 skipped; finished in 27.46ms (21.08ms CPU time)

Ran 13 tests for test/primary/AsseteraPrimarySales.t.sol:PrimarySalesInitTest

[PASS] test_ComplianceGate_CanStillBeOpenedAndClosedDeliberately() (gas: 38966)
 [PASS] test_ComplianceGate_IsClosedForAnActionNobodyHasDeclaredYet() (gas: 13169)
 [PASS] test_ComplianceGate_IsClosedForEveryOrdinal() (gas: 932433)
 [PASS] test_Initialize_CannotReinitialize() (gas: 19730)
 [PASS] test_Initialize_GatesEveryDeclaredAction() (gas: 16582)
 [PASS] test_Initialize_GrantsAllFourRoles() (gas: 39203)
 [PASS] test_Initialize_RevertsOnAnyZeroSigner() (gas: 3761094)
 [PASS] test_ParamsHashAllowed_IsFalseForTheZeroAndUndeclaredActions() (gas: 13412)
 [PASS] test_ParamsHashAllowed_IsTrueForEveryDeclaredAction() (gas: 12083)
 [PASS] test_SettlementOperatorRole_IsDistinctFromKycAndFee() (gas: 15752)
 [PASS] test_SettlementSigner_HoldsNeitherOfTheOtherTwoRoles() (gas: 36563)
 [PASS] test_TrustedForwarderIsTheConstructorArgument() (gas: 13088)
 [PASS] test_Version() (gas: 14740)
 Suite result: ok. 13 passed; 0 failed; 0 skipped; finished in 2.98ms (1.10ms CPU time)

Ran 4 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerRebasingAssetTest

[PASS] test_Rebasing_ADownwardRebaseIsClampedToZeroRatherThanPanicking() (gas: 472455)
 [PASS] test_Rebasing_ARebaseBeforeTheCallIsOutsideTheMeasurement() (gas: 408672)
 [PASS] test_Rebasing_ARebaseDuringTheCallIsCountedAsDelivery() (gas: 400983)
 [PASS] test_Rebasing_TheSameVenueIsRefusedWhenTheBuyerHoldsNoPosition() (gas: 381785)

Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 3.97ms (4.97ms CPU time)

Ran 10 tests for test/primary/VenueSettlerHostile.t.sol:VenueSettlerLyingVenueTest
[PASS] test_LyingVenue_APreExistingAssetDonationIsNotHandedToTheBuyer() (gas: 422022)
[PASS] test_LyingVenue_APushBackBeyondTheWholeDebitIsRefused() (gas: 427992)
[PASS] test_LyingVenue_ASmallPushBackLeavesWithTheRefundRatherThanBeingStranded() (gas: 411787)
[PASS] test_LyingVenue_ASplitDeliveryCountsInFullOnceTheRouterHalfIsForwarded() (gas: 427353)
[PASS] test_LyingVenue_DeliveringExactlyTheFloorIsAccepted() (gas: 393721)
[PASS] test_LyingVenue_DeliveringOneWeiBelowTheFloorIsRefused() (gas: 423446)
[PASS] test_LyingVenue_DeliveringToTheRouterIsForwardedToTheBuyer() (gas: 404131)
[PASS] test_LyingVenue_DoingNothingAtAllIsRefused() (gas: 355731)
[PASS] test_LyingVenue_OverDeliveryIsMeasuredInFullAndAllofItReachesTheBuyer() (gas: 399208)
[PASS] test_LyingVenue_OverDeliveryLeavesNothingForAnyoneElseToClaim() (gas: 398599)
Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 7.87ms (11.02ms CPU time)

Ran 14 tests for test/primary/PermitAndSettle.t.sol:PermitAndSettleTest
[PASS] test_PermitAndSettle_AStrangerCannotCarryTheBuyersPermit() (gas: 117272)
[PASS] test_PermitAndSettle_BubblesTheInnerRevert() (gas: 125155)
[PASS] test_PermitAndSettle_CannotReachAdminFunctionsWithoutTheRole() (gas: 89433)
[PASS] test_PermitAndSettle_CodelessTokenAddressReverts() (gas: 60909)
[PASS] test_PermitAndSettle_DivergentDomain_FallsBackToAllowance() (gas: 1312333)
[PASS] test_PermitAndSettle_DivergentDomain_SigningAgainstTheRealDomainWorks() (gas: 1309660)
[PASS] test_PermitAndSettle_EmitsPrimarySettledFromTheRouter() (gas: 547538)
[PASS] test_PermitAndSettle_IsStoppedByThePauseLever() (gas: 147769)
[PASS] test_PermitAndSettle_LeavesNoStandingAllowance() (gas: 540775)
[PASS] test_PermitAndSettle_ReentrantTokenCannotReenterSettlePrimary() (gas: 2369075)
[PASS] test_PermitAndSettle_Relayed_IdentityIsTheBuyerNotTheRelayer() (gas: 593715)
[PASS] test_PermitAndSettle_SettlesWithNoPriorAllowance() (gas: 549300)
[PASS] test_PermitAndSettle_SignatureThatDoesNotRecover_DoesNotRevertTheSettlement() (gas: 540935)
[PASS] test_PermitAndSettle_TokenWithoutErc2612_FallsBackToAllowance() (gas: 1251574)
Suite result: ok. 14 passed; 0 failed; 0 skipped; finished in 9.91ms (9.08ms CPU time)

Ran 11 tests for test/primary/VenueSettlerShares.t.sol:VenueSettlerSharesTest
[PASS] test_Shares_ADonationIsNeitherForwardedNorCounted() (gas: 528220)
[PASS] test_Shares_AMultiplierChangeDuringTheCallStillReturnsToBaseline() (gas: 556660)
[PASS] test_Shares_ARebaseDuringTheCallIsNotCountedAsDelivery() (gas: 524671)
[PASS] test_Shares_DeliveryIsMeasuredInSharesButJudgedInUnderlying() (gas: 548842)
[PASS] test_Shares_ForwardsTheExactShareDeltaAndReturnsToBaseline() (gas: 503241)
[PASS] test_Shares_RebasingModeAgainstAPlainErc20Reverts() (gas: 276423)
[PASS] test_Shares_RejectsAFalseTransferSharesReturn() (gas: 544738)
[PASS] test_Shares_RejectsAnUnknownAccountingMode() (gas: 273551)
[PASS] test_Shares_SigningTheProvidersRawNominalAmountReverts() (gas: 548820)
[PASS] test_Shares_TheMoneyLegIsUnchanged() (gas: 505446)
[PASS] test_Shares_TheSameRouteUnderErc20ModeStillReverts() (gas: 573348)
Suite result: ok. 11 passed; 0 failed; 0 skipped; finished in 5.50ms (9.40ms CPU time)

Ran 35 test suites in 165.72ms (265.36ms CPU time): 231 tests passed, 0 failed, 1 skipped (232 total tests)

9 About Nethermind

Nethermind is a Blockchain Research and Software Engineering company. Our work touches every part of the web3 ecosystem - from layer 1 and layer 2 engineering, cryptography research, and security to application-layer protocol development. We offer strategic support to our institutional and enterprise partners across the blockchain, digital assets, and DeFi sectors, guiding them through all stages of the research and development process, from initial concepts to successful implementation.

We offer security audits of projects built on EVM-compatible chains and Starknet. We are active builders of the Starknet ecosystem, delivering a node implementation, a block explorer, a Solidity-to-Cairo transpiler, and formal verification tooling. Nethermind also provides strategic support to our institutional and enterprise partners in blockchain, digital assets, and decentralized finance (DeFi). In the next paragraphs, we introduce the company in more detail.

Blockchain Security: At Nethermind, we believe security is vital to the health and longevity of the entire Web3 ecosystem. We provide security services related to Smart Contract Audits, Formal Verification, and Real-Time Monitoring. Our Security Team comprises blockchain security experts in each field, often collaborating to produce comprehensive and robust security solutions. The team has a strong academic background, can apply state-of-the-art techniques, and is experienced in analyzing cutting-edge Solidity and Cairo smart contracts, such as ArgentX and StarkGate (the bridge connecting Ethereum and StarkNet). Most team members hold a Ph.D. degree and actively participate in the research community, accounting for 240+ articles published and 1,450+ citations in Google Scholar. The security team adopts customer-oriented and interactive processes where clients are involved in all stages of the work.

Blockchain Core Development: Our core engineering team, consisting of over 20 developers, maintains, improves, and upgrades our flagship product - the Nethermind Ethereum Execution Client. The client has been successfully operating for several years, supporting both the Ethereum Mainnet and its testnets, and now accounts for nearly a quarter of all synced Mainnet nodes. Our unwavering commitment to Ethereum's growth and stability extends to sidechains and layer 2 solutions. Notably, we were the sole execution layer client to facilitate Gnosis Chain's Merge, transitioning from Aura to Proof of Stake (PoS), and we are actively developing a full-node client to bolster Starknet's decentralization efforts. Our core team equips partners with tools for seamless node set-up, using generated docker-compose scripts tailored to their chosen execution client and preferred configurations for various network types.

DevOps and Infrastructure Management: Our infrastructure team ensures our partners' systems operate securely, reliably, and efficiently. We provide infrastructure design, deployment, monitoring, maintenance, and troubleshooting support, allowing you to focus on your core business operations. Boasting extensive expertise in Blockchain as a Service, private blockchain implementations, and node management, our infrastructure and DevOps engineers are proficient with major cloud solution providers and can host applications in-house or on clients' premises. Our global in-house SRE teams offer 24/7 monitoring and alerts for both infrastructure and application levels. We manage over 5,000 public and private validators and maintain nodes on major public blockchains such as Polygon, Gnosis, Solana, Cosmos, Near, Avalanche, Polkadot, Aptos, and StarkWare L2. Sedge is an open-source tool developed by our infrastructure experts, designed to simplify the complex process of setting up a proof-of-stake (PoS) network or chain validator. Sedge generates docker-compose scripts for the entire validator set-up based on the chosen client, making the process easier and quicker while following best practices to avoid downtime and being slashed.

Cryptography Research: At Nethermind, our cryptography Research team conducts cutting-edge internal research and collaborates closely with external partners on cryptographic protocols, consensus design, succinct arguments and folding schemes, elliptic curve-based STARK protocols, post-quantum security and zero-knowledge proofs (ZKPs). Our research has led to influential contributions, including Zinc (Crypto '25), Mova, FLI (Asiacrypt '24), and foundational results in Fiat-Shamir security and STARK proof batching. Complementing this theoretical work, our engineering expertise is demonstrated through implementations such as the Latticefold aggregation scheme, the Labrador proof system, zkvm-benchmarks, and Plonk Verifier in Cairo. This combined strength in theory and engineering enables us to deliver cutting-edge cryptographic solutions to partners and clients.

Smart Contract Development & DeFi Research: Our smart contract development and DeFi research team comprises 40+ world-class engineers who collaborate closely with partners to identify needs and work on value-adding projects. The team specializes in Solidity and Cairo development, architecture design, and DeFi solutions, including DEXs, AMMs, structured products, derivatives, and money market protocols, as well as ERC20, 721, and 1155 token design. Our research and data analytics focuses on three key areas: technical due diligence, market research, and DeFi research. Utilizing a data-driven approach, we offer in-depth insights and outlooks on various industry themes.

Our suite of L2 tooling: Warp is Starknet's approach to EVM compatibility. It allows developers to take their Solidity smart contracts and transpile them to Cairo, Starknet's smart contract language. In the short time since its inception, the project has accomplished many achievements, including successfully transpiling Uniswap v3 onto Starknet using Warp.

- **Voyager** is a user-friendly Starknet block explorer that offers comprehensive insights into the Starknet network. With its intuitive interface and powerful features, Voyager allows users to easily search for and examine transactions, addresses, and contract details. As an essential tool for navigating the Starknet ecosystem, Voyager is the go-to solution for users seeking in-depth information and analysis;
- **Horus** is an open-source formal verification tool for StarkNet smart contracts. It simplifies the process of formally verifying Starknet smart contracts, allowing developers to express various assertions about the behavior of their code using a simple assertion language;
- **Juno** is a full-node client implementation for Starknet, drawing on the expertise gained from developing the Nethermind Client. Written in Golang and open-sourced from the outset, Juno verifies the validity of the data received from Starknet by comparing it to proofs retrieved from Ethereum, thus maintaining the integrity and security of the entire ecosystem.

General Advisory to Clients

As auditors, we recommend that any changes or updates made to the audited codebase undergo a re-audit or security review to address potential vulnerabilities or risks introduced by the modifications. By conducting a re-audit or security review of the modified codebase, you can significantly enhance the overall security of your system and reduce the likelihood of exploitation. However, we do not possess the authority or right to impose obligations or restrictions on our clients regarding codebase updates, modifications, or subsequent audits. Accordingly, the decision to seek a re-audit or security review lies solely with you.

Disclaimer

This report is based on the scope of materials and documentation provided by you to [Nethermind](#) in order that [Nethermind](#) could conduct the security review outlined in **1. Executive Summary** and **2. Audited Files**. The results set out in this report may not be complete nor inclusive of all vulnerabilities. [Nethermind](#) has provided the review and this report on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. This report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on this report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, [Nethermind](#) disclaims any liability in connection with this report, its content, and any related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. [Nethermind](#) does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and [Nethermind](#) will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.