NETHERMIND SECURITY

Smart Contract Audit & Security Services

Offering

About Nethermind Security

Nethermind Security provides auditing services for Ethereum and Starknet protocols, and collaborates on large infrastructure security projects.

Our work also includes formal verification and building threat detection bots on the Forta Network. The team has a strong academic background; most 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.

Nethermind Security has audited leading web3 protocols, including Gnosis Chain, StarkWare, Lido, PolygonID, AAVE, and Brevan Howard. Our suite of open-source tools features Clear, a recently launched open-source formal verification tool for Solidity smart contracts.

Content

This document describes the Nethermind Security's Smart Contracts Audit offering and standard processes. Moreover, it defines other complementary security services that we offer. The document is organized into the following sections:

SECTION 1

Secure Auditing Solutions

Our team of experienced smart contract auditors will conduct a thorough review of your smart contracts to identify vulnerabilities and weaknesses in the code that could compromise the security or functionality of your project.

Our smart contract audit service includes the following:

- **A comprehensive review of your smart contract code**
- **Identification of vulnerabilities and weaknesses in the code**
- **Recommendations for improving the security and functionality of the smart contracts**
- **A detailed report outlining our findings and recommendations**

Moreover, we can also provide the following:

- **Performance reviews and recommendations for gas efficiency improvement**
- **In-depth analysis of the test suite**

We also have a strong background in:

- **Implementing fuzzing tests, white-box tests, and black-box tests**
- **Formal verification of Starknet smart contracts, zero-knowledge circuit verification, and EVM-based smart contract verification.**
- **Developing solutions for monitoring smart contracts in real-time using the Forta network**

Academic Background and Security Experience

More than 50% of our security team holds a Ph.D. in Computer Science. Together, they have published 150+ scientific articles in the most important academic conferences, holding 1,500+ citations in Google Scholar. We have extensive experience in smart contract auditing and have worked with numerous blockchain projects in Solidity and Cairo to ensure the security and reliability of their smart contracts.

Nethermind Security utilizes the experience and knowledge of other teams within Nethermind; the research team, smart contract development team, and protocol engineering team, among others. As of now, Nethermind is comprised of 220+ professionals.

Customer-Oriented Approach

The Nethermind Security team will work closely with your project team via multiple sync calls and communication channels to ensure the audit is completed efficiently and effectively. Your team will have full visibility of the audit process, and findings will be discussed on the go, so your team can start working on the fixes as soon as we find a problem. We are confident in our ability

to provide high-quality auditing services and assure you that the audit will be conducted with the utmost professionalism and attention to every detail. We are taking an agile approach to our auditing process, enabling our team to deliver value to our clients much faster and with increased transparency. Clients are invited to participate actively throughout the audit process.

SECTION 2

Methodology for Smart Contract Audits

Before the audit starts, the Nethermind Security team will ask for access to the repository and the definition of a commit hash to be used during the audit. The client must also decide on the products composing the audit report. Additional products composing the audit report are listed below:

- **Check for security and code-matching technical specifications (non-optional)**
- **Gas optimization and code performance studies (optional)**
- **Abstract formal specification of listed functionalities of the protocol (optional)**
- **In-depth evaluation of the test suite (optional)**
- **Proposal of bots for monitoring the smart contract (optional)**
- **Implementation of the bots for monitoring the smart contract (optional)**
- **Formal Verification of the smart contract (optional)**

The audit begins in a joint meeting where the client walks us through the codebase and provides all the documentation available for the audit, including the test suite. We recommend that tests provide code coverage above 90% (100% is desirable). A joint communication channel is established between both teams, and the meeting dates are set.

The audit process is represented in Fig. 1.

Before kick off	Kick off	Audit	Reaudit	Final
Nethermind scopes the audit The client decides on the services composing the audit report The client defines the commit hash and provides access to the repository	Joint meeting for code walk through Creation of a joint communication channel Definition of dates for the sync meetings Client provides technical specifications	The Nethermind team analyzes the protocol and use cases Nethermind performs the technical audit and makes notes of all the issues and points of attention Frequent communication between both teams	Nethermind delivers the initial audit report The client reads the report, evaluates each finding, and perform the fixes when necessary Nethermind receives the reaudit notes from the client, and reaudits the code	Nethermind creates the final report and send to the client Client validates the report, and reports any inconsistencies At the end of the audit, the teams can perform joint marketing campaigns to promote the protocol

Fig 1. Audit Process is divided into five main stages: a) activities that must be developed before the project starts; b) activities that must be developed when the project is starting; c) activities that must be accomplished during the technical phase; d) activities that take place during the reaudit phase. e) activities that must be performed when finishing the audit.

- **During the audit**, the Nethermind team dedicates an initial time to learn the functionalities and use cases of the protocol. When the team is comfortable with the protocol, the technical audit starts, and our team will search for issues, exploits, and attack vectors. The code will be matched against the technical specification. The client's team and Nethermind auditors will have many meetings to discuss the audit progress.

- **When the technical audit is complete,** the client will receive an initial document listing issues and points of attention raised by Nethermind. The client will evaluate the list of findings. At the same time, the Nethermind team will work on the audit report (the audit report also evaluates the documentation provided by the client and the quality of the test suite). Once the initial audit report is ready, it is shipped to the client.

- **During the reaudit**, the client evaluates the report and performs the fixes in the code. The client may request more information and meetings with the Nethermind team. When the client finishes evaluating the findings and fixing the code, the Nethermind team rechecks the code. This process is reiterated until both teams reach a final decision (the client will comment on every issue, and that comment will be part of the audit report). Then, the final report is sent to the client for final approval.

SECTION 3

Going Beyond Traditional Smart Contract Reports

At Nethermind Security, we generate comprehensive audit reports. A high-quality audit report for a smart contract is critical to ensuring the contract's safety, security, and overall functionality. An audit report provides an independent, third-party review of the contract's code and identifies any vulnerabilities, weaknesses, or areas of improvement. This helps ensure that the contract will perform as intended and not expose users to unnecessary risks.

There are several reasons why having a detailed audit report is important. The audit report helps to prevent costly and potentially devastating errors. Smart contracts are often used for financial transactions or to manage sensitive data, meaning that any errors or vulnerabilities in the code could result in significant financial losses or data breaches. An audit report can identify and address these issues before they become major problems.

Moreover, having a good audit report can increase the confidence of users and stakeholders in the contract's reliability and safety. Smart contracts are often used in decentralized applications, where trust is a critical factor. A complete audit report following the best auditing practices can help to build trust in the project and its underlying technology.

Our audit process can also help to identify areas of improvement in the contract's design and code. This can help ensure the contract is scalable, maintainable, and efficient. By identifying potential issues early on, developers can address them before they become major problems, improving the overall quality of the code and reducing the risk of future vulnerabilities.

Nethermind Security audit reports comprise the following sections:

- **Executive Summary**: a high-level summary of the audit
- **List of Audited Files**: delimits the scope of the audit
- **Assumptions**: describes the assumptions assumed for the audit
- **Summary of Issues**: a table summarizing the identified issues
- **System Overview**: a description of the protocol and its core functionalities
- **Formal Specification of Selected Contracts (optional)**: a mathematical description using formal methods
- **Risk Rating Methodology**: a strategy used to rank the issues identified
- **List of Issues**: an in-depth description of the issues defined along the audit
- **Gas Optimization (optional)**: a section describing actions to reduce gas consumption
- **Complementary Validations Performed by Nethermind (optional)**: when agreed upon, the Nethermind team can run fuzzy tests, white-box testing, and black-box testing, ensure that all branches are covered by test cases, and other testing related activities for ensuring the highest quality of the audit
- **Technical Specification of Bots for Monitoring the Smart Contracts (optional)**: a high-level description of the set of proposed Forta bots for monitoring the smart contracts
- **Documentation Evaluation**: assessment of the documentation provided for the audit
- **Client's Test Suite Output**: execution and assessment of the test suite provided by the client
- **Analysis from Automated Tools**: findings detected by automated testing tools

SECTION 4

Architecture Assessment for Smart Contracts

The architecture assessment comprehensively analyzes smart contracts' underlying design and structure. This involves evaluating how various components interact, the flow of data, and the integration of external dependencies within the system. By carefully reviewing the contract's architecture, we can identify potential weaknesses or inefficiencies that might not be immediately obvious during a code-level audit.

One of the primary benefits of an architecture assessment is the early detection of security vulnerabilities. Flaws in the design of smart contracts can lead to serious security risks that might only become apparent once the system is live. Identifying these issues at the architectural level allows for proactive mitigation, reducing the likelihood of costly security incidents in the future.
It provides the opportunity to optimize the performance of smart contracts. Poorly designed structures can lead to inefficiencies, such as excessive gas consumption, negatively impacting performance and costs. Refining the contract's architecture can improve operational efficiency and streamline processes.

A robust architectural design supports the system's future growth and evolution, allowing for the integration of new features or components without compromising security or performance. This makes the contract more adaptable to changing business requirements and technological advancements.

SECTION 5

Security Assistance During Development

Our Security Assistance During Development service provides clients with ongoing access to our experienced security researchers throughout development. Our clients can ensure that their smart contracts undergo continuous security reviews, addressing potential vulnerabilities before they become problematic. This service enables developers to receive expert real-time guidance, creating a more secure and resilient product.

This is a proactive approach to security. Rather than waiting for a final audit, our security researchers engage at critical stages of development, providing insights on design decisions, code implementation, and integrating new features. This ongoing support minimizes the chances of introducing security flaws, allowing for quicker identification and resolution of potential issues. This service helps optimize the development process's efficiency. By maintaining regular security reviews, developers can avoid costly delays caused by major overhauls or fixes that might arise from discovering issues late in the cycle. With continuous feedback, teams can ensure their code is secure without sacrificing development speed.

This service offers clients the flexibility to adapt to changing needs during development. Whether reviewing new modules, advising on third-party integrations, or helping with compliance with best practices, our security experts are available to provide tailored assistance whenever needed. This results in a more secure, streamlined development process and ensures that the final product meets the highest security standards.

SECTION 6

Real-Time Smart Contract Monitoring

Smart contracts are not infallible and can contain errors or vulnerabilities that attackers can exploit. In some cases, these vulnerabilities led to the loss of millions of dollars. Therefore, it is recommended to monitor smart contracts in real-time to detect any unusual behavior or security issues as early as possible. Adopting the Forta bots for monitoring your smart contracts adds another layer of protection. The bots can monitor almost every action, inconsistency, unexpected transaction, broken invariants, and hack attempt; thus can be promptly identified, reported, and countered.

The Forta network provides a platform for the real-time monitoring of smart contracts. It allows developers and security professionals to set up custom alerts that trigger when specific events occur within a smart contract. These events could include the movement of funds, changes in contract ownership, or the execution of a particular function. Real-time monitoring of smart contracts using the Forta network can help detect potential security issues before they become significant problems.

Furthermore, the Forta network's real-time monitoring can be particularly valuable in DeFi applications, where the value of assets held within smart contracts can be significant. By monitoring these contracts in real time, developers and security professionals can quickly respond to any

security issues and prevent the loss of funds. **As of July 2024, we have built 200+ detection bots for 20 protocols across 7 Blockchains, including Yearn, dYdX, Flexa, and TraderJoe.** The total value monitored by these bots is 11 billion USD.

Our expertise and involvement span multiple areas:

- **Custom bot development**
- **Forta Detection Bot Wizard**
- **Generalized bot maintenance work**
- **Forta Agent Tools library**
- **Smart contract development for Forta**
- **Scanner node operator for Forta Network**
- **Forta Governance Council member**
- **Forta Arbiter committee member**

Forta continuously expands into the emerging landscape of new L1s and L2s, bringing runtime monitoring and anomaly detection to Ethereum, Avalanche, Polygon, BNB Chain, Fantom, Arbitrum, and Optimism.

SECTION 7

Formal Verification of Smart Contracts

Formally verifying smart contracts significantly increases the assurance of their correctness and security by rigorously analyzing their code and mathematical properties. This involves using formal methods, such as theorem proving and model checking, to verify that the contract behaves as intended and contains no critical flaws. The benefits of formally verifying smart contracts are numerous. It can significantly reduce the risk of errors and vulnerabilities, resulting in financial losses or other negative consequences. Formal verification can also increase confidence in the contract's functionality and ensure it will perform as expected.

Formal verification can help to identify potential weaknesses or flaws in the protocol allowing developers to address these issues before they become major problems. This can save time and resources by avoiding costly rework or contract failure. Moreover, formal verification can provide a solid basis for compliance. It can demonstrate that a contract has been rigorously analyzed, which can help mitigate legal risks and attract institutional interest. As blockchain technology becomes increasingly prevalent, the need for secure and reliable smart contracts will only grow, making formal verification an essential part of the development and compliance process.

SECTION 8

Research and Analytics

Our research solutions serve as the nexus where our theoretical knowledge meets real-world

applications. Catering to a wide range of clients, from institutional players to Web3 organizations and DAOs, we specialize in solving industry problems via innovative applications of our research.

Cryptography Research

We believe that diversity of expertise is key to tackling complex challenges and that cross-disciplinary collaboration is essential to driving meaningful innovation. Our Cryptography Research team works closely with our external partners and continuously conducts internal research across several domains.

Our team has expertise in the design and analysis of cryptographic protocols of all sorts of nature. In particular, we mostly focus on a wide range of ZKP topics, spanning from theory-focus problems (e.g. analysing the security of a protocol) to designing new cutting-edge schemes. We have expertise in all sorts of ZKP-related schemes and notions: IOPs, Polynomial Commitment Schemes, Folding, Recursion, IVC and PCD, the FRI protocol, Lookup arguments, arithmetization, etc. In addition to ZKP, we work on emerging powerful cryptographic primitives such as FHE, Witness Encryption, time-lock puzzles, etc.

Protocol Research

Our team possesses a rigorous understanding of all layers of the distributed ecosystem production line, from the consensus mechanisms that keep our systems ticking, to the applications that drive user adoption, and all of the crypto-economic infrastructure in between.

We distinguish ourselves through our understanding of how these layers interact. This allows us to mould protocols where they can be moulded, and identify incompatibilities when they arise. Our protocol research team's expertise includes:

- **Consensus design, with a particular focus on incentives**
- **Maximal Extractable Value (MEV)**
- **L2s, including decentralization, incentivization, coordination of roles, value capture and fees**
- **Shared Sequencers:**
  - **Value Capture & Value Redistribution to Sequenced Chains**
  - **Implications for Sequenced Chains in general**
- **Preconfirmations**
- **Decentralized Exchange Protocols (DEXs)**

Research Solutions

Our researchers are also able to provide comprehensive technical due diligence services that extend beyond conventional DeFi or economic analyses. By assembling teams proficient in both cryptography and software engineering, we meticulously scrutinize the low-level mechanics of projects that require a multidisciplinary approach to assess their viability for potential investment. We have collaborated with innovative funds such as 1RT/10T, offering expert advising and delving deep into the technical details of such protocols.

SECTION 9

Quality Assurance in our Smart Contract Audits

QA is a critical part of audits as it helps ensure the process and the resulting audit report are accurate. At Nethermind Security, the QA processes are rigorous and designed to ensure that the audit work meets the highest standards, ensuring consistency in the process and the resulting comprehensive audit report. Furthermore, QA processes also help to identify areas where improvements can be made in the audit process, increasing the efficiency and effectiveness of the audit process and leading to a better quality of the audit reports. Finally, by focusing on Quality Assurance, we can ensure that the audit work meets stakeholders' expectations. This is important as stakeholders have high expectations of the audit process and the resulting audit report. Failure to meet these expectations can result in losing trust and confidence. Stakeholders often rely on audit reports to decide whether to invest in a project.

SECTION 10

Preparation for a Smart Contract Audit

Preparing your smart contracts for an audit is an essential step in ensuring their security and reliability. Nethermind Security suggests our clients take the following actions before sending the smart contract for an audit.

- **Review the code yourself.** Before submitting your smart contract for an audit, you should review the code yourself to identify any obvious vulnerabilities or errors.
- **Use industry-standard libraries and frameworks.** Using industry-standard libraries and frameworks can help ensure your code is reliable and secure.
- **Comment on your code.** Commenting on your code can make it easier for auditors to understand how your smart contract works and identify potential issues. Add NatSpec documentation to all functions (describing what the function should do, arguments, and return variables).
- **Perform automated testing.** Automated testing can help you identify potential vulnerabilities in your smart contract before submitting it for an audit.
- **Check that all tests are passing.** Try to achieve a test code coverage of 80% or higher.
- Compile a list of resources that help understand the intended behavior and specifications of the project.
- **Prepare technical documentation.** Technical documentation should cover the majority of the codebase. If full documentation is incomplete, a more concise document explaining each contract and how they interact can suffice.
- **Provide instructions on how to execute the test suite.** Ensure the README or some other document accurately describes how to compile the code and run tests.
- **Use a Modular test suite.** A modular test suite allows auditors to create proofs of concepts and experiment exploit scenarios easily.
- **List properties and invariants.** A list of properties and invariants that must always be satisfied can improve audit quality.

By following these steps, you ensure that your smart contract is well-prepared for an audit and is as secure and reliable as possible.

SECTION 11

Testimonials

“

We worked with the Nethermind team on two audits of the Gyroscope Solidity codebase and have a third scheduled. The Nethermind team is a pleasure to work with and has the expertise needed to audit a complex codebase involving many interacting contracts as well as fixed point numerics, which are especially important for AMM applications. We have worked with a number of top auditors, and our work with Nethermind has been on the same or higher level.

Ariah Klages-Mundt
Co-Founder, Gyroscope

“

Collaborating with the Nethermind team on building the formal verification of our Cairo smart contracts was a real pleasure. The Nethermind team displayed excellent skills, and throughout the working process demonstrated their expertise and professional standards. Their knowledge of Cairo is profound and a valuable asset for StarkNet. We are happy, as always, to have them take part in building StarkNet.

Eli Ben-Sasson
Co-Founder & President, StarkWare

“

Working with Nethermind on the formal verification of our Cairo smart contracts has been a fantastic experience. Not only is the team at Nethermind extremely knowledgeable and professional, but they are also incredibly thorough and responsive. We are impressed by their mastery in the Cairo language as well as their expertise in StarkNet. They have been an invaluable partner to help ensure the security and correctness of our contracts, and we look forward to strengthening this partnership through future collaboration.

Jonathan Lei
Co-founder and CTO, zkLend

SECTION 12

Trusted by the best

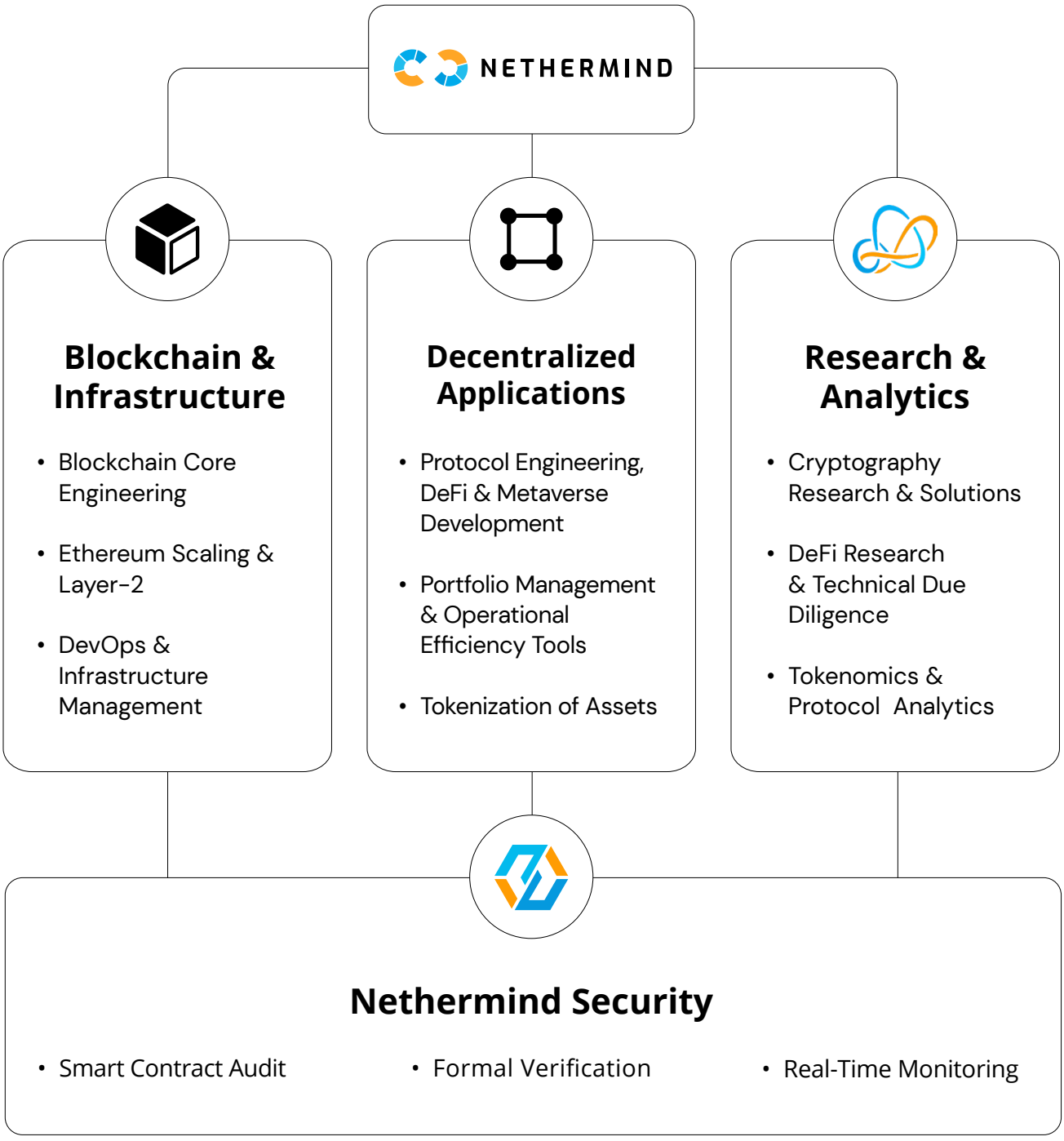
Nethermind Security Clients

AAVE
Argent
Atlendis
Avalanche
AVNU
Blockframe
Braavos
Briq
Carmine Finance
Cartridge
Chi Protocol
ClayStack
Cryptix
Decentraland
Dinero
Dojo
Dyve
Ekubo
Ethereum Foundation
EtherFi
Etherspot
EveryRealm
Fibrous
Fileverse
Fleek
Flexa
Forta
Game7
Gnosis Chain
Gyroscope
Haiko
Hopper Labs
HyperNest
HyperPlay
Influence
Jasmine Energy
JediSwap
Layer Akira
Libre
Lido
Lighter
Mangrove
Midnight Society
Monarch
Munchables
mySwap
Nodle
Numbers Game
NUNW
Ondo Finance
OpenZeppelin
Optimism
Ora
Pimlico
Pocket Network
Polygon ID
Polygon Village
Pontis
Pragma
PropHouse
Puffer
PWN
Pyth Network
RavenDAO
Rook
SafeStake
SithSwap
SkateFi
SMG
Sphere Finance
StarkGate
Starknet Foundation
Starknet ID
Starkware
Summon
Swell
TokenTable
Trader Joe
TxFusion
WebN/TruFin
Wilderworld
Worldcoin
zkLend
zkSync
zkVeggies
+ more

SECTION 13

About Nethermind

Nethermind is a leading infrastructure builder, researcher and core development team in the blockchain space. We empower enterprises and developers worldwide to access and build on the decentralized web. 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 collaborate with a global developer community and partners to advance Ethereum, Starknet, and the broader blockchain industry.



Nethermind Security

Nethermind Security provides auditing services for Ethereum and Starknet protocols, and collaborates on large infrastructure security projects. Our work also includes formal verification and building threat detection bots on the Forta Network. The team has a strong academic background; most 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. Nethermind Security has audited leading web3 protocols, including Gnosis Chain, StarkWare, Lido, PolygonID, AAVE, and Brevan Howard. Our suite of open-source tools features Clear, a recently launched open-source formal verification tool for Solidity smart contracts.

Nethermind Research

Nethermind Research is the research arm of Nethermind, creating synergies between the fields of cryptography, decentralized finance, and protocol research. With over 30 published articles and comprising 5 PhD holders and 2 CFA Charterholders, this team crafts research solutions that bridge the gap between theoretical knowledge and real-world applications. Collaborations include cutting-edge protocols like Lido, a leading decentralized liquid staking solution, and Obol Labs, an R&D team focusing on proof-of-stake infrastructure for blockchain.

Blockchain Core Development

We are active builders of the Ethereum and Starknet protocols, closely aligning our development and research work with their roadmaps. The core engineering team, consisting of over 20 developers, maintains, improves, and upgrades our flagship product - the Nethermind Ethereum Execution Client, launched in 2017. The Client supports Gnosis Chain, the OP- stack and Energy Web. Our ecosystem contributions for Starknet include Juno, an open-source full-node implementation, and Voyager, a block explorer and API platform.

Enterprise Engineering

The Enterprise Engineering team works as a bridge between blockchain and the financial sector and institutions. We develop DeFi solutions, provide engineering and due diligence research for VC firms, and work with leading digital asset investment firms. The team recently collaborated with leading ecosystem contributors as the engineering backbone of Libre Capital, an onchain tokenized asset issuance and distribution protocol.

L2 and Starknet tooling

Our L2 tooling team is forging tools and products for ZK scaling solutions. Nethermind is a strategic partner to the Starknet Foundation, and as such, we're dedicated to developing tools and infrastructure that advance Starknet's decentralization and accessibility. Key contributions feature Juno, an open-source full-node implementation, and Voyager, a block explorer and API platform. Other Layer-2 projects include Starknet RPC service, support for the OP-Stack in the Nethermind client and the zkSync Era Plugin for the Remix IDE.

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.

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.

Nethermind's Starknet RPC service offers developers, protocols, and businesses fast, reliable on-chain data access with dedicated support from the RPC builders for expert problem-solving.

Artificial Intelligence & ZK Engineering

Focused on AI and zero-knowledge research advancement, our projects range from core protocol development of zero-knowledge components for Starknet to design and development of Hardware and Software for Worldcoin.

DevOps and Infrastructure Management

Building processes, infrastructure, security, cloud, and administration are all within this team's scope. They specialize in setting up private and hybrid blockchain networks, emphasizing stored and processed data's security, privacy, and accuracy. 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 Ethereum, Polygon, Gnosis, Solana, Cosmos, Near, Avalanche, Polkadot, Aptos, and StarkWare L2.

NETHERMIND
SECURITY

nethermind.io
NethermindSec
Nethermind
NethermindETH
hello@nethermind.io
Nethermind
NethermindStark
NethermindEth
Nethermindeth