Pharos Network: A Deep Parallel Layer-1 Network Illuminating Scalable Innovations

Pharos Team

1 Introduction

"We believe in: rough consensus and running code" - David Clark, 1992

Web3 represents the next generation of Internet protocols and services, highlighting innovations in ownership-centric, trustless technology. The first Web3 network, Bitcoin, has undergone a 15-year public-network test through global consensus and the Simplified Payment Verification (SPV) security mechanism, transforming the stateless Internet into a stateful design. Ethereum, the first Web3 decentralized application platform, enhanced asset programmability with its Ethereum Virtual Machine (EVM), and its ERC token standards have become the de facto digital asset standards. Recently, new Web3 networks such as Solana, Aptos, and Sui have further advanced Web3 technology, improving performance and asset security.

Starting from a payment network design and envisioning a new era of the value Internet, Web3 networks still lack infrastructure comparable to the traditional financial world. For example, in 2021, digital wallets supported 28.6% of global sales transactions [1], amounting to over 13.3 trillion dollars across billions of users. Secure, reliable, and stable real-time payments are essential to enhancing the Super-App user experience within these digital wallets.

Pharos is a new Web3 network designed for trustless innovations. Pharos aims to elevate the asset capabilities to unprecedented levels, with an inclusive infrastructure for both Crypto Assets and Real-World Assets (RWA). To achieve this, Pharos provides decentralized infrastructure for high-density asset value, leveraging the power of performance, scalability, security, and extensibility. By supporting real-time payments, decentralized applications (dApps), transactions, and liquidity, Pharos is paving the way for the next century of financial technology, offering exceptional user experiences and scalability to billions of accounts. Pharos also aims to provide a superior Layer 1 (L1) blockchain by integrating compute and storage of assets directly into the L1 framework. This integration ensures enhanced security and efficiency of digital assets management. Pharos envisions a thriving ecosystem powered by technology sharing and collaborative problem-solving within the community.

Standing at the forefront of Web3 innovation, Pharos is poised to bridge the gap between traditional financial infrastructure and the emerging decentralized economy of the future. It aims to make the benefits of blockchain technology accessible to all, while continually evolving through community-driven advancements.

2 Design Rationale

Blockchains are more than ledgers, they are computers. With the growing complexity of Web3 applications, the need for scalability and efficiency becomes paramount. The industry is witnessing a shift towards more powerful parallel processing capabilities to meet the demands of mass adoption. The evolution from sequential processing to highly parallelized architectures marks a significant milestone in blockchain's journey. Significant strides have been made in optimizing L1 network protocols, execution engines, scheduling algorithms, and storage management to enhance overall throughput to the next scale.

To provide a comprehensive view of the existing and future technologies in L1 parallelism, we define the power of network parallelization into distinct **D**egree of **P**arallelism (DP) and their expected capabilities in enhancing the performance (p) as follows:[1]

DP0: Sequential Model (p)

Linear processing without parallel execution.

DP1: Scalable Consensus (2p∼5p)

Improved consensus mechanisms with scalability, adaptivity, security, and high-performance block propagation ensuring immediate finality.

DP2: DP1 + Parallel Transaction (5p∼10p)

High-throughput transaction-level parallel processing by leveraging parallel speculation, scheduling, and multi-thread, multi-instruction power.

DP3: DP2 + Pipelining (15p∼50p)

Incorporates multi-stage pipelined processing, significantly increasing overall throughput and resource utilization while ensuring multi-level-finality and security.

DP4: DP3 + Parallel Merklization and Accelerated State Access (100p∼500p)

The pinnacle of parallelism, with a unified Merklization and indexing scheme to solve IO efficiency and state bloat issues, enables full parallelism of data access and minimizing storage costs.

DP5: DP4 + Parallel Heterogeneous Computation (500p+)

Fully utilize the asymmetric computing power of nodes and support the parallel capabilities of heterogeneous hardware, such as GPU, TEE, FHE accelerator, ZK accelerator, and smart network card, to enhance the security of the whole network and applications while comprehensively promoting network parallel capabilities. chain, using the hash of the accepted block as the previous hash.

The larger the scale of the scenario, the higher DP is required. With the development of Web3 applications, some platforms have recently entered DP3 from DP2. However, to thoroughly address the scalability challenges and ensure mass adoption, transitioning to DP4 is essential. Pharos is uniquely positioned to lead the advancement of blockchain technology into the next generation,

[1] We focus on the parallelism of L1 only in this white paper. L2/L3 can directly work together with L1 to further improve the overall scalability.

aiming to facilitate the widespread adoption of Web3. It is built on core design principles to embrace DP4 and even DP5 parallelism.

Scalability(DP4): Pharos supports Internet-scale users and transactions through deep parallel and ultra-high performance techniques. Pharos employs parallelism at all levels of the blockchain, boosts performance, reduces latency, and optimizes resource utilization for complex dApps:

1. **Scalable Consensus Protocol:** A high-throughput, low-latency BFT consensus protocol that fully utilizes the whole network resource.
2. **Dual VM Parallel Execution:** A parallel dual VM (EVM and WASM) execution layer with advanced compilation techniques.
3. **Full Life Cycle Asynchronous Pipelining:** Enables parallel and asynchronous processing throughout the full lifecycle of each transaction (execution, consensus, I/O) and among blocks.
4. **High-Performance Store with Authenticated Data Structure (ADS):** Provides exceptional throughput, low-latency I/O, and cost-efficient state storage, securely scalable to billions of accounts.

Extensibility(DP5): Pharos features a modular Special Processing Networks (SPN) for seamless integration of new software, hardware, and geographic decentralization, supporting diverse use cases and emerging technologies.

Meanwhile, Pharos prioritizes network, asset, and application security by employing fundamental security measures, including client SPV on petabyte-level data, Byzantine fault tolerance, auditability, analytical tools, and authenticated storage to ensure data integrity and security.

Pharos is committed to fostering a decentralized future with limitless scalability, unwavering security, and boundless extensibility. Technical details will be discussed in the next section.

3 Technical of Pharos

3.1 Scalable Network and Consensus

Pharos employs a consensus protocol that delivers both high throughput and low latency [2]. It is specifically designed for large-scale nodes in a wide-area network, enhancing system robustness and consensus security. There are two design goals of Pharos consensus: responsiveness and optimal resource utilization. Responsiveness refers to the property whereby the processing speed of the blockchain system is limited only by the actual network delay, without introducing any timeouts or waiting intervals. Regarding resource utilization, consensus nodes should exhibit a symmetric communication pattern to operate simultaneously.

Most existing blockchain consensus protocols, however, fail to achieve these two goals, leading to performance and scalability issues. Some blockchains, for instance, impose a time interval between two consecutive blocks, resulting in a fixed block generation rate. As a result, there is an upper bound of the system throughput, regardless of network or computational optimizations. Moreover, the most common communication pattern is *proposing-voting*. In this pattern, a single proposer is responsible for building and broadcasting the block content, while others merely sign and send relatively small voting message. Thus, the proposer becomes the bottleneck. As the number of validators increases, more network resource of the proposer is consumed, causing performance degradation. However, with more validators, the available bandwidth resource also increases. By fully utilizing the resources of all validators, a scalable consensus network can be achieved.

Figure 1: Left: blockchain consensus that generates blocks with fixed time slots and only one validator is allowed to propose within each slot.
Right: Pharos consensus allows all validators to propose freely and simultaneously.

To achieve this goal, Pharos consensus incorporates two innovative features. Firstly, it does not rely on fixed timing assumptions, allowing block generation to be responsive to actual network delays rather than being constrained by a predefined timeout. Secondly, Pharos consensus enables simultaneous proposals from all consensus nodes, eliminating the reliance on a single proposer. These features significantly enhance scalability and substantially increase the system throughput of Pharos. Furthermore, Pharos consensus introduces a flexible advancement paradigm, enabling nodes to propose and commit blocks more flexibly to accommodate varying workloads and network conditions. For example, a slow and far-away node can propose less frequently, without worrying about its proposal being suppressed. In our global testbed comprising 100 nodes, the prototype implementation achieves a throughput of over 130,000 transactions per second.

3.2 Parallel Transaction

Transactions in Pharos network are executed with exceptional performance, scalability, and efficiency. The architecture comprises two key components: the scheduler and the executor. The scheduler is a core component for parallel scheduling and executing transactions, utilizing optimized algorithms to maximize parallelism and minimize conflicts. The executor features dual virtual machine (EVM and WASM) engines for fast and flexible smart contract execution. The EVM engine ensures compatibility with Solidity contracts, while the WASM engine provides high-performance execution for a broader set of smart contract languages.

3.2.1 Scheduler

The Pharos scheduler forms near-optimal parallel transaction groups with high concurrency to ensure fast transaction execution with integrity and correctness, efficient transaction validation, deterministic finality, and conflict handling. To ensure parallel execution efficiency and preset deterministic finality, Pharos employs several effective strategies:

- **Parallel Hint Generation:** Leveraging static analysis and speculative execution [3] of smart contracts, Pharos achieves near-accurate transaction read-write set hints that minimize conflicts and increase parallelism. A comprehensive compilation framework and fine-grained analysis algorithm [4] provide robust static analysis, generating detailed control flow and data flow insights. This approach reduces runtime burden by making hints as accurate as possible.
- **Transaction Dependency Analysis:** Pharos analyzes transaction dependencies based on these read-write sets and generates optimal parallelizable transaction groups using the union-find algorithm. These groups are merged to align with the number of execution workers, with all state objects that each group may access preloaded before execution through a single batched I/O operation, significantly reducing execution time [5].

- **Optimistic Execution and Pipeline Finality:** Pharos employs optimistic execution [3] and a "Pipeline Finality" algorithm to quickly converge execution results and determine the final state [6].

Moreover, Pharos is committed to continuously improving parallelism: Validators are able to balance tasks such as speculative execution while fully leveraging multi-core CPU and I/O resources. Executors also handle scheduling work, optimizing overall resource utilization and eliminating single points of resource concentration. Performance is further enhanced through the parallel and storage optimization of global data with read-write set conflicts, such as global counters. Additionally, fine-grained algorithmic control facilitates faster conflict detection and minimal-cost re-execution, culminating in optimal parallel performance.

3.2.2 Executor

Pharos executor is implemented with a straightforward interface and embedded host, combining modern compilation optimization techniques with blockchain-specific enhancements to achieve high performance and low fees. For typical contracts, it can improve performance tenfold compared to existing EVM implementations. Pharos executor supports both EVM and WASM, allowing interoperability between the two. This dual support enables developers to benefit from the EVM ecosystem while gaining efficiency improvements and access to existing libraries in high-level languages like Rust, C++, Go, and Java. The main process of the execution layer is as follows:

3.2.3 Dialect Conversion

Dialect conversion [7] maps opcodes and system function calls between EVM and WASM, ensuring seamless contract translation between platforms. Using Pharos IR, the intermediate representation allows scalability and compatibility with other VMs.

3.2.4 Pharos IR

Pharos IR is designed based on the LLVM and MLIR frameworks as a register-based intermediate representation [8]–[10]. It performs various optimizations and converts EVM opcodes and WASM bytecode into a unified dialect representation. This scalability improves consistency, correctness, and development efficiency.

3.2.5 Symbolic Verifier

Pharos IR is designed to be verification-friendly, with strong type and maximum static determinism. Pharos Symbolic Verifier is a symbolic execution engine based on Pharos IR. It utilizes abstract interpretation and refinement execution techniques to efficiently explore execution paths and program states. It aims to achieve the following goals:

- **Compilation Integrity:** Ensures logic consistency between the EVM dialect and WASM code, as well as between different optimization passes.

- **Speculative Execution Integrity:** Mitigates security concerns arising from speculative execution complexity by performing formal verification on each speculative rewind to ensure integrity.

- **Verification on User Defined Specification:** Pharos Spec is a domain specification language allowing user-defined safety properties that can be verified on Symbolic Verifier such as type safety, memory safety, control flow, and data flow integrity. The verification system of Pharos Spec and Symbolic Verifier build an extra layer of security for smart contracts on the Pharos network.

3.2.6 Passes

Optimization passes applied to the dialect include dead code elimination [11], constant folding [12], function inlining, opcode fusion and register promotion [13]. These optimizations produce the best-optimized machine code for extreme performance. The VM optimizes high-frequency I/O operations as much as possible into memory operations and optimizes memory operations into register operations, such as certain global storage or contract cost counters, which can not only minimize instruction consumption but also reduce read-write conflicts and improve parallelism, thereby reducing costs.

3.3 Pipelining

Pharos employs a distinctive block-level multi-stage pipeline mechanism, similar to the pipeline in a superscalar processor in modern computers, to enhance throughput, performance, and resource utilization. This pipeline design enables Pharos to achieve significant speedup over transaction-level parallel execution by optimizing the entire block processing lifecycle. Figure. 3.3 visually explains how this pipeline functions in Pharos.

Pharos pipeline optimizes effectiveness by dividing the block lifecycle into six fine-grained stages: consensus ordering, parallel execution, database reading, Merklization, block building, and database flushing. This division allows for maximum parallelism across multiple blocks by different processors. The pipeline scheduling separates IO-intensive stages from CPU-intensive ones and prioritizes the execution result of the most recent block [14].

Pharos incorporates 3 types of finality within the pipeline stages:

1. **Ordering Finality:** Every transaction is permanently sequenced within a block, ensuring consistent execution order and transaction hashes.

2. **Transaction Finality:** Each transaction is executed deterministically, providing users with results and account state changes.

3. **Block Finality:** The block is finalized through building and flushing, allowing users or APIs to instantly access complete block information, eliminating delays in Merkle root generation and queries.

Pharos pipeline ensures integrity and security through its design. It guarantees no data loss at any finalized stage, even if all validators fail simultaneously, thus maintaining data integrity.

Validators can resynchronize and update to the latest state after a crash. The system ensures that the storage of a new block is deferred until the previous block is fully stored. Additionally, it strictly manages dirty reads and phantom reads at all stages to ensure data consistency.

By implementing this robust pipelining mechanism, Pharos not only enhances the performance and scalability of the network but also ensures a high level of security and data integrity, thereby supporting the seamless and efficient execution of complex Web3 applications.

3.4 Parallel Merklization and Accelerated State Access

When enabling parallel capabilities to drive mass adoption, IO performance and storage scalability are the next big problem because of the state bloat.

Traditional blockchain storage usually uses a two-layer architecture to index state data: a universal key-value storage system based on LSM trees is used as the storage backend, and an Authenticated Data Structure (ADS) is built on top. ADS is usually a Merkle tree, and each node in the tree is independently stored in a key-value storage system. For example, Ethereum uses MPT+LevelDB [15] as the state data storage engine. However, traditional storage has limitations:

1. **Long I/O Paths:** The I/O path in the two-tier architecture is long, and access to a single key-value pair involves querying multiple tree nodes, each requiring multiple disk I/Os in a database based on a multi-level LSM tree.

2. **Write and Space Amplification:** The encoding method of Merkle trees causes write amplification and space amplification. Updating a single object requires updating the hash values of each node on the vertical path in the tree. To maintain multi-version capabilities, nodes must be recoded and written to the storage backend even if only one field within the node is updated.

3. **Hash-Based Addressing:** This results in significant I/O bandwidth consumption. LSM-based key-value storage systems regularly perform compaction in the background, merging low-level SST files into high-level files in an orderly manner. The inherent randomness of Merkle hashing results in a random distribution of keys, causing overlap among multiple SST files, and consuming a large amount of I/O bandwidth.

To overcome the aforementioned limitations posed by the state bloat problem, we have developed Pharos Store. It is the first blockchain-native, verifiable storage solution, that provides high-performance read/write along with efficient Merklization. With Pharos store, we can achieve up to 15.8× throughput and 80.3% storage cost saving [16].

Innovations of Pharos Store include:

- **ADS Pushdown:** Integrates ADS into the storage engine, enabling fine-grained I/O optimization by eliminating the boundary between ADS and backend storage.

- **Delta Encoding:** Persists tree nodes via delta encoding, storing only modified data to reduce I/O and space amplification.

- **Version Addressing:** Replaces hash addressing with version addressing, eliminating file compaction overhead by ordering data by version number.

3.4.1 Single-Layer with ADS inside

To break through the limitations of traditional two-level architecture, Pharos Store adopts a new architecture that pushes ADS to the storage engine and separates storage indexes and user metadata to further reduce I/O amplification.

Pharos Store consists of three main parts:

1. **DMM-Tree (Delta-encoded Multi-version Merkle Tree):** A built-in multi-version Merkle tree that supports dictionary trees, B-trees, and other tree structures. Leaf nodes record the file number and offset of the user data.

2. **LSVPS (Log-Structured Versioned Page Store):** Provides a page index abstraction between memory and secondary storage for DMM-Tree.

3. **VDLS (Versioned Data Logging Stream):** Stores user metadata in an append-only manner.

Additionally, Pharos Store provides an API layer and an append-only persistence layer designed to flexibly support modern storage devices, including SSD/NVMe or cloud storage. It acts as an abstraction layer for backend storage systems (such as EXT4, NAS, HDFS, CEPH), supporting append writes and efficient random access reads.

3.4.2 Delta Encoding

In the LSVPS layer of Pharos Store, each page aggregates incrementally updated data into DeltaPages and persists them. To control the size of the DeltaPage, a threshold is set. When a DeltaPage accumulates a certain number of updates, it is frozen and a new DeltaPage is created to continue aggregating updates. To minimize the overhead of page loading, each page will generate a full checkpoint BasePage after a certain number of updates.

3.4.3 Version Addressing

Leveraging the monotonically increasing nature of blockchain data versions, Pharos Store introduces version addressing to replace hash-based indexing, improving disk bandwidth efficiency.

- **API Support:** Provides APIs using (V, k) as the index key for accessing any historical data based on version number.

- **Version Management:** Each tree node and page maintains a version number, allowing efficient data retrieval without the need for reordering or compaction.

- **Ordered Storage:** Stores pages in version order, enabling efficient range operations and batch pruning of historical pages.

Pharos Store represents a significant step forward in blockchain-native storage, providing cryptographic tamper evidence with excellent performance and resource efficiency. This innovation enables true scalability for billions of users and growing performance demands, offering a universal solution for other blockchains.

3.5 Parallel Heterogeneous Computation

To build a robust, scalable, and decentralized blockchain network, Pharos employs a carefully designed network topology and consensus mechanism that integrates three essential types of nodes: validator nodes, full nodes, and relayer nodes. This structure ensures the network's integrity, performance, and security.

Figure 2: The Pharos Network

Validator nodes form the backbone of the Pharos consensus mechanism, using a Byzantine fault-tolerant (BFT), proof-of-stake protocol. Hundreds of Validators collaborate to maintain a secure and reliable network, processing user transactions swiftly. Beyond traditional transaction fees and staking rewards, Validators can earn additional income through Restaking [17], allocating their token certificates to SPNs or DApps to enhance scalability, security, and liquidity, and to receive additional rewards.

Full Nodes store complete blocks and state transactions, providing critical support functions. Although they do not participate directly in consensus, they offer Fast State Sync to enable other nodes to quickly synchronize with the latest state and provide Parallel Hint to validators to enhance parallel execution efficiency.

Relayer Nodes are lightweight clients, that store the latest state and handle recent transactions. With lower machine configuration requirements, they are ideal for joining DePIN networks. Relayer Nodes provide essential Node Services, including transaction simulation and forwarding, and they earn additional rewards by efficiently forwarding SPN Messages within the SPN network.

By integrating a sophisticated node architecture with an advanced consensus protocol, Pharos ensures a scalable and efficient blockchain network capable of supporting a wide range of applications and maintaining high performance under varying conditions. On top of that, Pharos natively supports the deployment of lightweight Special Processing Networks (SPNs). By selecting a validator within the Pharos network, users can create a dedicated SPN. These SPNs can employ entirely different protocols, such as AIoT private networks or networks specifically designed for multi-party privacy-enhancing computations (MPT). Lightweight SPNs also support networks that access TEE hardware for transaction confidentiality/MEV and other networks that rely on specialized hardware.

3.5.1 Heterogeneous Computing Network

SPNs address the need for specialized data and computational resources, enhancing scheduling for heterogeneous computing resources. They facilitate seamless interaction between different blockchain ecosystems and computing environments, optimizing resource use, improving performance, and reducing costs. Additionally, SPNs support dynamic scaling, ensuring the network can handle increasing demands and complex applications.

Restaking allows validators to dynamically allocate their staked assets to SPNs or DApps, boosting security and liquidity while supporting flexible resource distribution. Validators restake assets to support SPNs, enhancing security and reliability. Restaking also redistributes staked assets, improving network liquidity and economic incentives, and enabling adaptive resource allocation based on network demands for optimal performance.

For security, SPNs incorporate privacy-enhanced computing with techniques such as secure multi-party computation, TEE, and homomorphic encryption to protect data during AI processes. Efficient and secure resource scheduling between SPNs and validators is achieved through Restaking, which enhances overall network security. Advanced verification techniques ensure computation integrity and prevent malicious activities.

Pharos SPNs advance blockchain technology by combining enhanced resource allocation, interoperability, and security, creating a robust infrastructure for digital innovation. Leveraging SPNs and Restaking, Pharos aims to build a flexible, secure, and efficient blockchain ecosystem adaptable to the rapidly changing digital world.

4 Conclusion

Pharos represents a significant advancement in blockchain technology, offering a scalable, efficient, and secure solution for the future of Web3. By embracing parallelism at all levels and integrating innovative storage and execution techniques, Pharos ensures exceptional performance and reliability. With a commitment to inclusivity, security, and extensibility, Pharos is poised to lead the next generation of decentralized applications and digital innovations. The roadmap ahead is filled with exciting milestones and developments that will continue to push the boundaries of what is possible in the blockchain space. Together, we can build a decentralized future that empowers users and fosters innovation across the globe.

References

[1] *The global e-wallets landscape: 2023 and beyond*, `https://www.hellozai.com/blog/the-global-e-wallets-landscape-2023-and-beyond`.

[2] S. Liu, W. Xu, C. Shan, *et al.*, "Flexible advancement in asynchronous bft consensus," in *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023, pp. 264–280.

[3] R. Gelashvili, A. Spiegelman, Z. Xiang, *et al.*, "Block-stm: Scaling blockchain execution by turning ordering curse to a performance blessing," in *Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, 2023, pp. 232–244.

[4] Y. Lu, C. Liu, M. Zhao, *et al.*, "Fsc: A fast smart contract transaction execution approach via read-write static analysis," *Authorea Preprints*, 2023.

[5] C. Gorenflo, L. Golab, and S. Keshav, "Xox fabric: A hybrid approach to blockchain transaction execution," in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 2020, pp. 1–9.

[6] *Road to high performance: Parallel evm for bnb chain*, `https://www.bnbchain.org/en/blog/road-to-high-performance-parallel-evm-for-bnb-chain`.

[7] Y. Shi, K. Casey, M. A. Ertl, and D. Gregg, "Virtual machine showdown: Stack versus registers," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 4, no. 4, pp. 1–36, 2008.

[8] *Dialect conversion*, `https://mlir.llvm.org/docs/DialectConversion/`.

[9] C. Lattner, M. Amini, U. Bondhugula, *et al.*, "Mlir: Scaling compiler infrastructure for domain specific computation," in *2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, IEEE, 2021, pp. 2–14.

[10] Y. Chen, Z. Guo, R. Li, *et al.*, "Forerunner: Constraint-based speculative transaction execution for ethereum," in *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*, 2021, pp. 570–587.

[11] J. Cocke, "Global common subexpression elimination," in *Proceedings of a symposium on Compiler optimization*, 1970, pp. 20–24.

[12] S. Muchnick, *Advanced compiler design implementation*. Morgan kaufmann, 1997.

[13] A. Sastry and R. D. Ju, "A new algorithm for scalar register promotion based on ssa form," in *Proceedings of the ACM SIGPLAN 1998 Conference on Programming Language Design and Implementation*, 1998, pp. 15–25.

[14] X. Qi, Z. Chen, H. Zhuo, *et al.*, "Schain: Scalable concurrency over flexible permissioned blockchain," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, IEEE, 2023, pp. 1901–1913.

[15] *Merkle patricia trie*, `https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/`.

[16] S. Tian, Z. Lu, H. Zhuo, *et al.*, "Letus: A log-structured efficient trusted universal blockchain storage," in *Companion of the 2024 International Conference on Management of Data*, 2024, pp. 161–174.

[17] E. Team, "Eigenlayer: The restaking collective," *URL: https://docs.eigenlayer.xyz/overview/whitepaper*, 2024.