# The Reddio Blockchain

# Parallel EVM for Autonomous AI

March 2025
V1.0

**Abstract**

The blockchain industry faces critical hurdles—scalability, performance, and computational efficiency—hindering mainstream adoption, especially in decentralized finance (DeFi), gaming, and artificial intelligence (AI). Reddio overcomes these with a transformative Layer 2 architecture, fusing parallel processing and optimized state management. Built on CUDA-compatible Parallel Execution Virtual Machines (EVMs), a Peer-to-Peer (P2P) GPU network, and a Decentralized Modular Sequencer SDK, Reddio delivers unmatched throughput and efficiency while preserving full Ethereum Virtual Machine (EVM) compatibility. Outpacing existing Layer 2 solutions like zkEVMs and Optimistic Rollups, Reddio sets a new standard for blockchain innovation, unlocking scalable, high-performance decentralized applications.

# 1. Introduction

Blockchain technology has redefined trust and interoperability in digital systems, with Ethereum serving as a cornerstone for decentralized applications. However, as adoption grows, Ethereum [2] faces persistent challenges: limited transaction throughput (15-30 transactions per second), escalating fees, and sequential processing that struggles to meet the computational demands of emerging use cases like AI-driven automation and real-time gaming. Existing Layer 2 solutions, such as Optimistic Rollups and zkEVMs, have made strides in addressing scalability but often compromise on decentralization, latency, or developer flexibility—shortcomings that hinder their suitability for high-performance applications.

Reddio's mission is to overcome these barriers by delivering a scalable, secure, and efficient Layer 2 platform that seamlessly integrates with Ethereum. Designed to empower developers and users alike, Reddio introduces a novel architecture that accelerates transaction processing, enhances computational capacity, and simplifies blockchain customization, while maintaining high levels of decentralization. Reddio aims to unlock new possibilities for web3, driving adoption across industries by providing a reliable and upgradeable infrastructure. This white paper details Reddio's technical framework, explores its core components, and articulates its vision for a decentralized future where blockchain technology rivals the ubiquity and efficiency of modern cloud systems.

# 2. Reddio's System Overview

Reddio is designed as a **cutting-edge Layer 2 solution** that enhances the Ethereum blockchain by addressing its inherent limitations and scaling challenges.

Specifically targeting sectors like **decentralized finance (DeFi)**, **AI-driven applications**, and **gaming**, Reddio's architecture is engineered to maximize **performance**, **security**, and **developer accessibility**. This section outlines the core components of Reddio's system, each designed to tackle specific challenges in the blockchain ecosystem.

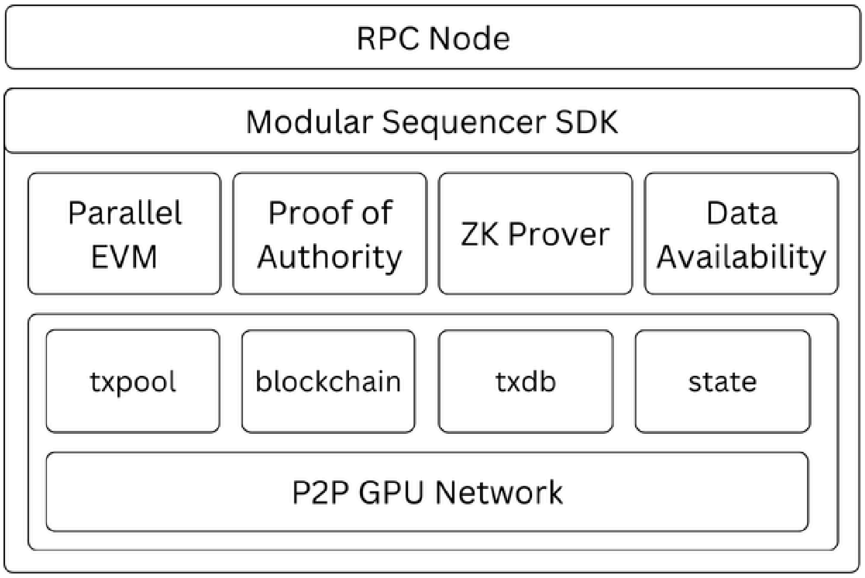


*Figure 1: Architecture Overview*

Reddio introduces a groundbreaking architecture that addresses critical bottlenecks in traditional blockchain systems. At its core, Reddio leverages four key innovations, which will be elaborated in details in the following sections too:

1. **Parallel Execution and Asynchronous Storage Optimizations:** Reddio introduces advanced state management techniques to minimize I/O overhead and maximize hardware efficiency. By enabling direct state reading and asynchronous state loading, Reddio reduces the latency associated with traditional Merkle Patricia Trie (MPT) traversals. Its pipelined workflow decouples transaction execution, state reading, and storage updates into overlapping phases, ensuring storage operations do not bottleneck performance—a critical advantage for real-time applications.
2. **The 1st ever CUDA-Compatible Parallel Execution Virtual Machine (EVM):** Reddio's Parallel EVM redefines smart contract execution by enabling parallel processing on GPUs. Unlike traditional EVMs that process transactions sequentially, Reddio's Parallel EVM executes multiple transactions simultaneously, leveraging NVIDIA's CUDA technology to boost processing speeds and computational efficiency. This capability significantly reduces latency and increases throughput—achieving over 10,000 transactions per second (TPS)—making it ideal for complex computations in AI-driven applications and high-transaction environments like gaming and DeFi. Crucially, Reddio maintains

full EVM compatibility, allowing developers to migrate existing Ethereum applications seamlessly while benefiting from substantial performance gains.

3. **Decentralized Modular Sequencer SDK:** The Modular Sequencer SDK serves as the backbone of Reddio's scalability and customization features. Built on the Yu framework [7] in Golang, this SDK powers Reddio's Layer 2 operations and provides developers with flexible tools to design and deploy custom Layer 2 or Layer 3 solutions. Supporting multiple virtual machines (e.g., CairoVM, EVM, and Parallel EVM) and customizable consensus mechanisms like Proof of Authority, the SDK simplifies blockchain development, enabling tailored platforms optimized for performance and functionality.
4. **Peer-to-Peer (P2P) GPU Network:** Reddio's P2P GPU Network harnesses distributed GPU resources to enhance blockchain processing power. By pooling computational capacity from nodes worldwide, this network enables intensive computations—such as those required by AI algorithms—to be performed directly on-chain, securely and efficiently. This democratizes access to high-performance computing, empowering developers to build sophisticated dApps without significant infrastructure investments, while scaling computational resources dynamically with network demand.

## 2.1 Integration with Ethereum

Reddio seamlessly integrates with Ethereum, utilizing it as the primary layer for **data immutability** and **final transaction settlement** while performing transaction processing and computation on **Layer 2**. This integration ensures that while Reddio can operate at a higher efficiency and lower cost, it still benefits from the robust security and wide adoption of the Ethereum network.

## 2.2 Security and Data Integrity

Security is a cornerstone of Reddio's design, particularly for sectors like finance and gaming where reliability is non-negotiable. Reddio employs zero-knowledge proofs (via the SP1 zkVM) to validate transactions swiftly and privately, ensuring computational integrity without exposing underlying data. Combined with a decentralized sequencer model, this approach mitigates risks associated with centralized systems (e.g., Optimistic Rollups' fraud delays), delivering robust security and trust without sacrificing execution speed.

## 2.3 Strategic Advantages

Reddio's architecture positions it as a leader in next-generation blockchain platforms. By addressing scalability, computational demands, and developer accessibility, Reddio enhances the Ethereum ecosystem and paves the way for broader blockchain adoption. Its ability to process over 10,000 TPS, integrate AI natively, and provide modular tools empowers developers to create responsive, secure, and innovative dApps—unlocking new possibilities across industries.

# 3. Parallel Execution and Asynchronous Storage Optimizations

This section delves into the technical backbone of Reddio, spotlighting the innovations that drive its high-throughput transaction processing and scalable architecture. These advancements build atop Ethereum's foundation, optimizing it for high-demand applications in decentralized finance, AI, and gaming. For exhaustive technical details, refer to [1].

Reddio revolutionizes blockchain performance by enabling parallel transaction execution and optimizing state storage, addressing Ethereum's sequential processing and I/O-heavy Merkle Patricia Trie (MPT) [2] bottlenecks. This subsection details Reddio's approach, which combines a batch-based execution model with an asynchronous, pipelined storage workflow to maximize concurrency and efficiency.

## 3.1 Batch-Based Parallel Execution

Reddio processes transactions in batches, leveraging multi-threaded parallelism to enhance throughput beyond Ethereum's single-threaded limits. Each block $B_l$ contains an ordered set of transactions $\langle T_1, T_2, \ldots, T_m \rangle$, which Reddio splits into batches executed concurrently across η threads, where η is the number of available threads.

- **Optimistic Execution:** Transactions within a batch execute optimistically on a consistent state snapshot, using an Optimistic Concurrency Control (OCC) strategy [3, 4]. Each thread processes its assigned transactions independently, reading from a shared state cache ($C_{state}$) initialized from the prior block's state. This eliminates the need for upfront read/write set declarations.
- **Conflict Detection and Resolution:** Post-execution, a coordinator identifies conflicts—defined as cases where a transaction $T_j$ (where $j > i$) reads a state item modified by an earlier transaction $T_i$. Conflicting transactions are aborted and deferred to the next batch for re-execution. This ensures deterministic serializability, where the final state matches the sequential order $\langle T_1, T_2, \ldots, T_m \rangle$(see correctness proof in [1]).
- **Performance Gains:** By executing independent transactions in parallel, Reddio achieves significant speedups—up to 3-5x in low-conflict workloads—compared to Ethereum's 15-30 TPS. Benchmarks show scalability with thread count, limited only by conflict rates and storage access efficiency [1].

This batch-based model reduces latency and boosts concurrency, making it ideal for high-volume scenarios like micro-payments or gaming interactions.

## 3.2 Direct State Access and Asynchronous Loading

Ethereum's MPT requires root-to-leaf traversals for state reads, incurring multiple I/O operations per access—accounting for ~70% of execution overhead [5]. Reddio optimizes this with direct state access and asynchronous node loading.

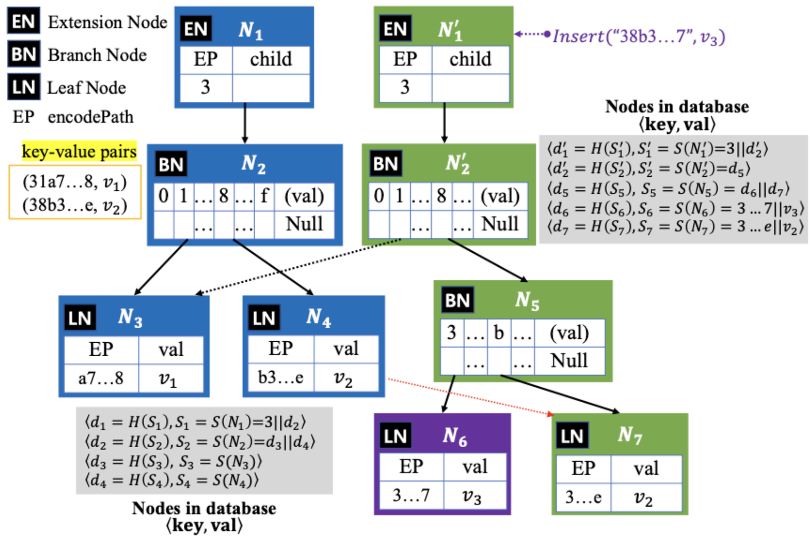


*Fig. 2: An example of Merkle Patrica Trie (MPT).*

- **Direct State Database:** Reddio maintains a parallel key-value store [4, 6] ($D_{direct}$) alongside MPT, recording state as ⟨A‖κ,v⟩ pairs, where (A) is the account address, κ is the storage key (or null for account state), and (v) is the value. During execution, the EVM fetches state directly from $D_{direct}$ via a (Get()) operation, bypassing MPT traversals and reducing read latency by orders of magnitude.
- **Asynchronous Node Preloading:** For writes, Reddio preloads MPT nodes concurrently with execution. Using ζ dedicated load threads (typically matching CPU core count), it fetches nodes into a memory cache ($C_{node}$) based on known account addresses or dynamically identified storage keys. Updates are buffered in $C_{state}$ and enqueued in $Q_{load}$, processed in parallel by LevelDB's [8] concurrent read capabilities. This ensures execution isn't stalled by I/O, with nodes ready for hashing when needed.
- **Consistency Mechanism:** To maintain MPT integrity, $D_{direct}$ updates include block height (I) recording state as (⟨A‖κ,v,I⟩), enabling recovery from crashes by discarding stale entries and rebuilding from MPT during replay [1].

This dual approach slashes I/O overhead, aligning storage speed with parallel execution demands.

### 3.3 Pipelined Storage Workflow

Reddio replaces Ethereum's sequential update-hash-store cycle with a pipelined workflow, overlapping these phases to maximize hardware utilization and minimize bottlenecks.

- **Decoupled Phases:** Transactions execute while state reads and writes occur concurrently. Updates are cached in $C_{state}$, with hashing and storage offloaded to separate threads via queues ($Q_n$ for hashing, $Q_p$ for persisting). This decoupling prevents storage from throttling execution, unlike Ethereum's synchronous model.
- **Dynamic Hashing:** Nodes modified during execution are pre-hashed asynchronously when they reach a commit point—determined by trie level and remaining transaction count (e.g., $L^*=4000$ for levels $r \geq 4$). If a node is updated again, rehashing is deferred to avoid wasted computation, balancing efficiency with accuracy [1].
- **Batch Storage:** Write sets from executed batches are cached in memory, then persisted in bulk to $D_{direct}$ and MPT. Overlapping writes are deduplicated, reducing I/O operations and leveraging larger, more efficient disk writes. Configurable cache sizes provide back-pressure, ensuring system stability under load.

This pipelined design delivers near-linear scaling with thread count, achieving over 13,000 TPS in optimized conditions, as detailed in [1].

## 4. CUDA-Compatible Parallel EVM

Building on the CPU-based parallel execution foundation, CuEVM [9] extends this capability to GPUs. Reddio's CUDA-Compatible Parallel EVM (the version based on CuEVM but optimized for blockchain execution) supercharges the optimized parallel execution framework by harnessing GPU parallelism, co-developed with the National University of Singapore. Building on the batch-based execution and asynchronous storage foundation, CuEVM leverages NVIDIA's CUDA technology to convert EVM opcodes into GPU-executable kernels, maintaining full Ethereum compatibility while unlocking the raw power of GPUs—hardware inherently designed for massive parallel computation. This subsection details CuEVM's opcode transformation, compatibility strategy, and GPU-driven performance enhancement.

### 4.1 Opcode Transformation into CUDA Kernels

CuEVM reimagines EVM execution by translating Solidity bytecode into CUDA kernels, enabling GPU-accelerated parallelism beyond CPU capabilities.

- **Opcode-to-Kernel Mapping:** Each EVM opcode (e.g., ADD, MUL, SHA3, SSTORE) is converted into a CUDA kernel—a function executed concurrently across thousands of GPU threads. For instance, an ADD opcode becomes a kernel that performs addition for multiple transactions simultaneously, while SHA3 computes hashes in parallel across a batch. This transformation is automated via a custom compiler that interprets EVM bytecode and generates CUDA equivalents, preserving semantic accuracy.
- **Execution Workflow:**
  - **Initialization:** Kernels are initialized with transaction-specific contexts—gas limits, stack values, and memory pointers—loaded into GPU global memory. This ensures each thread operates on isolated, transaction-specific data.
  - **Parallel Invocation:** A single kernel launch executes an opcode across all relevant transactions in a batch. For example, a batch of 1,000 SSTORE operations updates storage slots concurrently, leveraging CUDA's thread-level parallelism.
  - **Result Consolidation:** Post-execution, GPU threads synchronize to aggregate results (e.g., updated stack or storage values), which are then written back to Reddio's state database via the asynchronous storage pipeline.
- **Optimization:** Frequent opcodes are pre-optimized into lightweight kernels, minimizing overhead. Complex operations (e.g., cryptographic hashes) exploit GPU's parallel architecture, reducing per-transaction latency significantly compared to CPU execution.

This opcode transformation taps into GPUs' natural parallelism, amplifying the throughput of Reddio's batch-based execution model.

## 4.2 Ensuring EVM Compatibility

CuEVM maintains seamless compatibility with Ethereum's EVM, ensuring developers can deploy existing smart contracts without modification.

- **Semantic Preservation:** Each CUDA kernel replicates the exact behavior of its EVM opcode counterpart, adhering to Ethereum's gas semantics, stack operations, and state transition rules. For example, a CUDA-based CALL opcode executes contract invocations with identical gas costs and return values as the standard EVM.
- **Bytecode Interpretation:** CuEVM interprets Solidity-compiled bytecode directly, avoiding the need for source-level changes. The compiler maps EVM's 256-bit word operations to GPU-friendly 32-bit or 64-bit equivalents, handling overflows and underflows consistently with Ethereum's specification.
- **Fallback Mechanism:** For rare edge cases (e.g., unsupported opcodes or GPU-incompatible contracts), CuEVM seamlessly delegates execution to the

CPU-based parallel EVM, ensuring uninterrupted compatibility while maintaining performance for GPU-optimized workloads.

This compatibility layer bridges Ethereum's ecosystem to Reddio's GPU-driven architecture, offering a drop-in solution with enhanced performance.

## 4.3 Harnessing GPU Parallelism for Enhanced Execution

GPUs, designed for parallel computation, naturally extend Reddio's optimized parallel execution, pushing throughput and scalability to new heights.

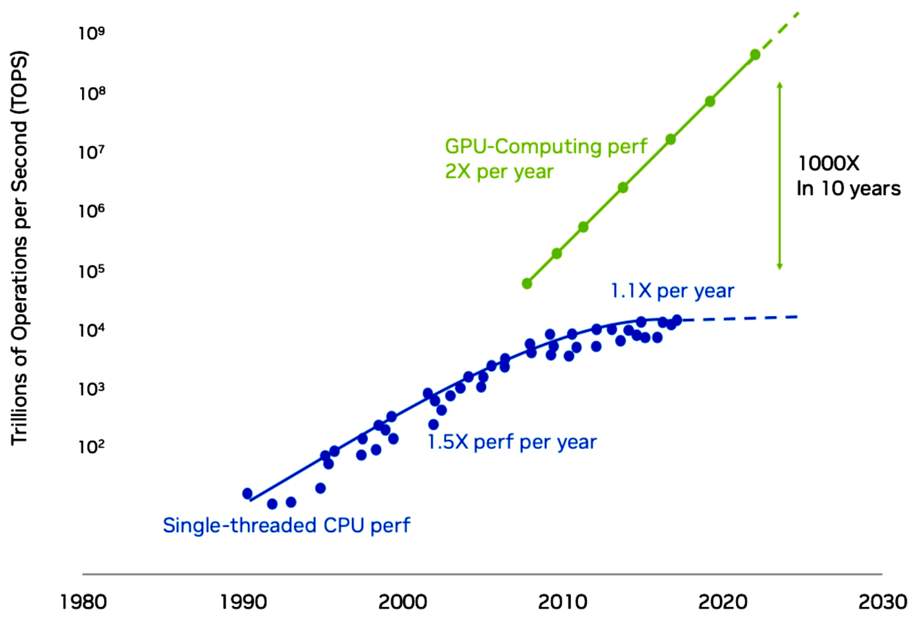


*Figure 3: GPU Acceleration: 1000X Performance Growth vs. CPU Limitations*

- **Massive Thread Capacity:** A single GPU supports thousands of threads (e.g., NVIDIA GPUs offer up to 65,536 threads per grid), dwarfing CPU core counts. CuEVM assigns each transaction—or sub-operation within a transaction—to a thread, executing entire batches in parallel. For instance, a 10,000-transaction batch can be processed in a single GPU pass, achieving over 10,000 TPS [1].
- **Thread Organization:** CUDA's hierarchy—threads grouped into blocks, blocks into grids—optimizes CuEVM's workload distribution. Blocks handle transaction subsets (e.g., 128 threads per block), while grids manage batch-level parallelism, reducing synchronization overhead and accelerating state updates.
- **Memory Efficiency:** CuEVM leverages CUDA's memory types—shared memory for fast intra-block data sharing (e.g., temporary stack values), global memory for transaction inputs, and registers for rapid arithmetic—minimizing

latency in compute-intensive tasks like cryptographic operations or AI-driven contract logic.

By harnessing GPU parallelism, CuEVM enhances the foundational parallel execution, delivering exponential performance gains for high-demand applications.

# 5. Decentralized Modular Sequencer SDK

Reddio's Decentralized Modular Sequencer SDK redefines Layer 2 and Layer 3 blockchain development, offering a flexible, scalable toolkit built on the robust Yu framework in Golang also designed by Reddio. This SDK power's Reddio own Layer also enables developers to craft custom blockchain solutions, integrating seamlessly with Reddio's high-performance architecture while broadening network participation. This subsection details its design and operational mechanics.

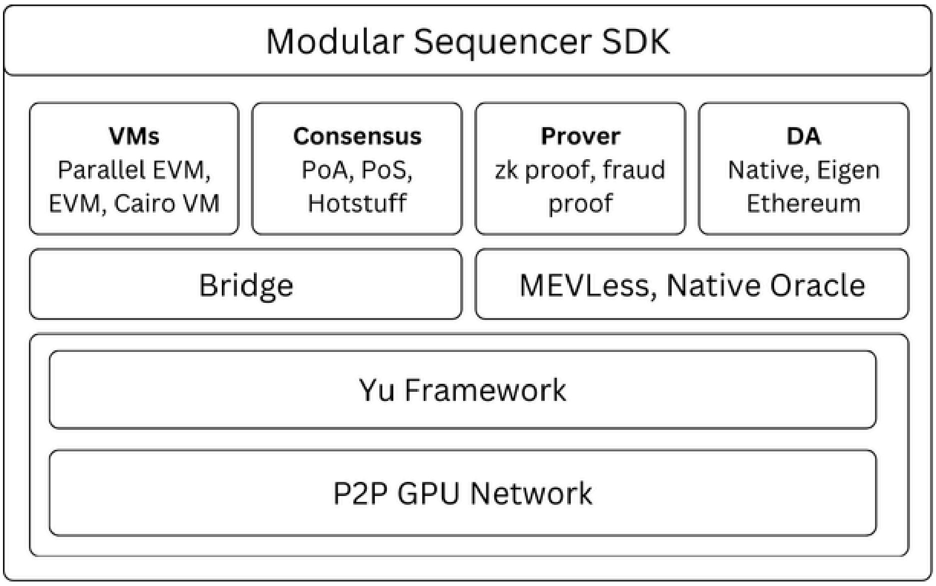


*Figure 4: Modular Sequencer SDK Framework*

### 5.1 Modular Architecture and Integration

The SDK's design prioritizes adaptability, enabling a wide range of applications through robust integration with virtual machines and Ethereum's Layer 1.

- **Virtual Machine Compatibility:** It supports CairoVM, Ethereum's EVM, and Reddio's Parallel EVM, blending Ethereum's foundational security with advanced parallel processing. Transactions processed via the Parallel EVM achieve over 10,000 TPS, leveraging optimizations detailed earlier [1].
- **Flexible Components:** The SDK offers modular building blocks—such as ZK Provers and customizable consensus options like Proof of Authority

(PoA)—allowing developers to craft solutions for diverse needs, from high-speed financial systems to intricate dApps.
- **Layer 1 Anchoring:** By syncing with Ethereum for final settlement via ZK or OP, the SDK ensures data immutability and trust, while delegating computation and sequencing to Reddio's Layer 2 for scalability.

This modular framework lets developers fine-tune performance and functionality without breaking Ethereum compatibility.

### 5.2 Operational Mechanics

The SDK streamlines the entire transaction processing workflow, for illustration purpose, we take Reddio network as an example:

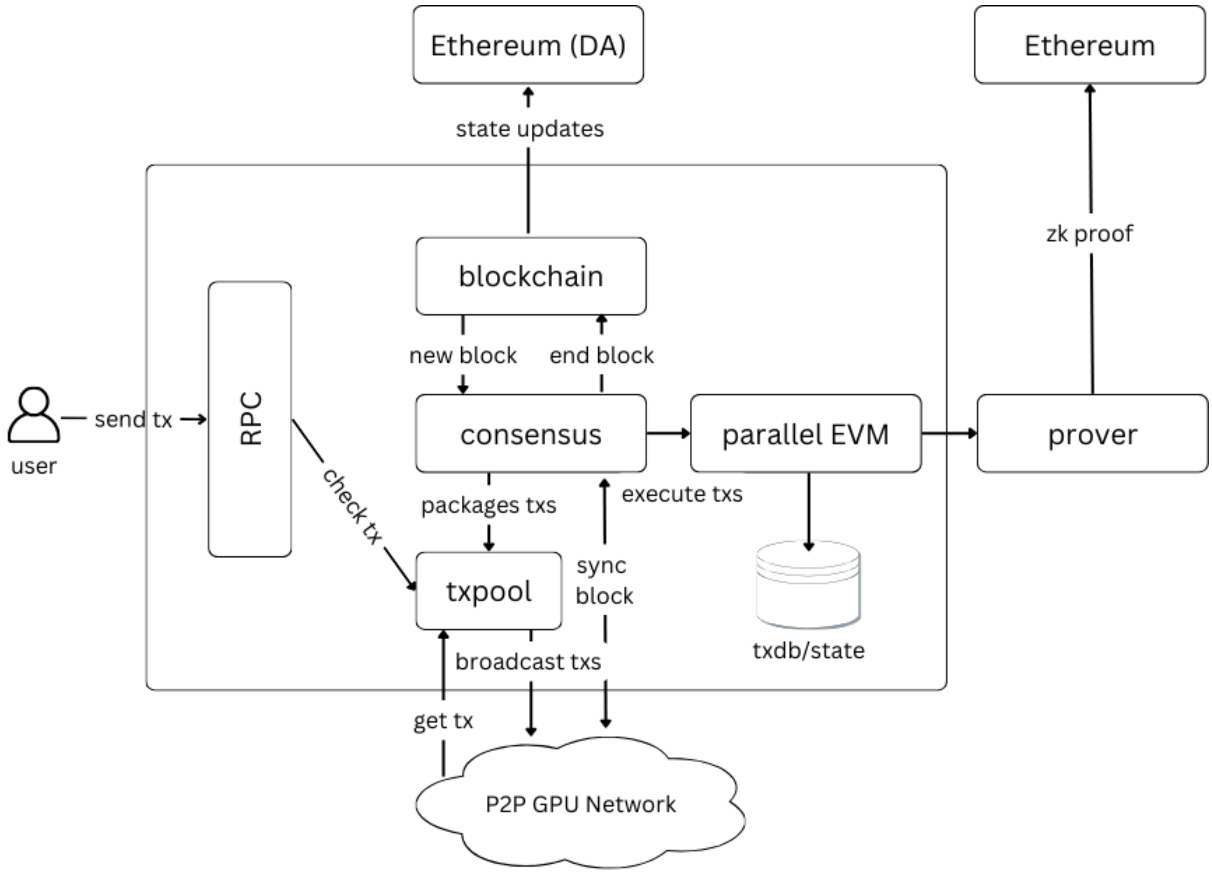


*Figure 5: Reddio workflow*

- **Transaction Submission**: Users submit transactions to the sequencer via RPC. Each transaction is rigorously checked for legality and integrity, ensuring compliance with network rules and security standards against vulnerabilities such as replay attacks.
- **Consensus and Block Production**: Transactions are pooled and periodically assembled into blocks by the SDK's consensus system. These blocks are

then processed in parallel, leveraging the Modular Sequencer's integration with Reddio's Parallel EVM, which optimizes execution state synchronization with Ethereum's Layer 1.
- **Data Availability and Proofs**: Post-execution, the SDK generates data availability proofs and employs SP1, a high-performance zkVM from Succinct Labs, for zero-knowledge proof creation. SP1's precompile-centric architecture and AVX-512 optimizations enable rapid, cost-efficient proofs—verified on Ethereum for ~300,000 gas—enhancing throughput and security without compromising decentralization [1].

This workflow ensures fast, trustworthy transaction processing, leveraging SP1's cutting-edge ZK capabilities.

# 6. P2P GPU Network

Leveraging the Modular Sequencer SDK and CUDA-accelerated execution, Reddio powers a decentralized Peer-to-Peer (P2P) GPU Network, enhancing computational capacity across its Layer 2 ecosystem. This network harnesses distributed GPU resources to support high-performance transaction processing and AI-driven applications, underpinned by a Proof of Authority (PoA) consensus model. This subsection details its architecture, integration, and ecosystem benefits.

## 6.1 Distributed Network Architecture

Reddio's P2P GPU Network aggregates CUDA-compatible GPU resources from nodes worldwide, forming a decentralized computing fabric optimized for efficiency and speed.

- **Node Participation:** Any individual or organization with CUDA-enabled GPUs can join as a node, contributing computational power for transaction validation and complex tasks. Participants earn transaction fees or network tokens, incentivizing broad engagement.
- **Dynamic Resource Allocation:** The network allocates GPU resources dynamically based on task complexity and demand—e.g., scaling thread assignments for a 10,000-transaction batch or an AI computation. This ensures optimal resource use and sustains high performance under varying loads.
- **PoA Foundation:** Building on the SDK's PoA consensus, the network designates GPU nodes as validators, streamlining block production while maintaining decentralization, a direct outcome of the modular sequencer's design.

This architecture delivers robust, scalable computing power, critical for Reddio's performance-driven goals.

## 6.2 Operational Benefits

The P2P GPU Network delivers transformative advantages, enhancing performance, economics, and innovation.

- **Scalability:** Its distributed design scales computational resources with demand, maintaining sub-second latency as transaction or AI workloads increase—crucial for real-time applications like gaming or DeFi trading.
- **Cost Efficiency:** By distributing loads across GPU nodes, Reddio reduces reliance on centralized infrastructure, lowering operational costs compared to traditional Layer 2s, with miners sharing gas revenue.
- **AI and Blockchain Fusion:** The network's robust GPU power enables on-chain AI functionalities—e.g., autonomous smart contracts or predictive analytics—paving the way for next-generation dApps.

These benefits position Reddio as a scalable, cost-effective platform for advanced blockchain use cases.

## 6.3 Decentralized Sequencing and Economic Model

Unlike centralized sequencer models, Reddio's P2P GPU Network employs a decentralized, PoA-based approach, driving participation and fairness.

- **Decentralized Sequencing:** Transaction ordering is distributed across GPU validators, eliminating single-point control and reducing manipulation risks. This enhances network security and trust, a direct extension of the SDK's PoA framework.
- **Revenue Sharing:** Validators contributing GPU resources earn a share of gas fees, proportional to their computational input. This incentivizes sustained participation, aligning miners' rewards with network health and transaction volume.
- **Ecosystem Growth:** By tying validator incentives to Reddio's success, the network fosters a community-driven model. Miners become stakeholders, potentially influencing governance and future enhancements, strengthening long-term resilience.

This economic structure differentiates Reddio, promoting both technical and participatory scalability.

## 7. Unique Use Cases Supported by Reddio

**AI Agent Network**: Reddio's robust computational and parallel processing capabilities enable AI agents to perform complex tasks like scheduling, leasing, and on-chain inference efficiently. This facilitates a dynamic network where AI agents can autonomously interact, make decisions, and manage digital assets with minimal latency.

**AI Inference On-Chain**: Reddio pioneers the integration of AI models directly on the blockchain, enabling on-chain data processing and real-time decision making, which is crucial for sectors like decentralized healthcare and financial services.

**On-Chain DEX & DeFi**: With the high performance to support on-chain execution of order books and matching engines, Reddio ensures that decentralized exchanges operate with unprecedented speed and decentralization, supporting the high-frequency trading requirements of modern decentralized financial ecosystems.

**Fully On-Chain Gaming**: By processing all game logic on the blockchain, Reddio creates a transparent and cheat-proof environment for online gaming, enhancing player trust and engagement.

## 8. Tokenomics and Network Participation

Reddio's ecosystem is powered by the RDO token, fostering a decentralized community of participants who collectively own and govern the network. The RDO token serves as the backbone for operations, security, and decision-making, driving engagement across validators, developers, and users.

The RDO token is essential for transaction and network fees, enabling all activities within Reddio's Layer 2—from processing high-throughput transactions to supporting AI-driven computations. Validators, operating within the Proof of Authority (PoA) consensus framework, must stake RDO tokens to join the GPU miner network, securing the system and earning rewards for their contributions. This staking requirement ensures a committed validator pool, enhancing network reliability without the resource demands of traditional mining.

Governance over the P2P GPU network is decentralized, with RDO token holders influencing critical aspects—such as validator selection, sequencing policies, and ecosystem upgrades. This empowers the community to shape Reddio's evolution, aligning technical advancements with collective goals. Validators also benefit from a revenue-sharing model, receiving a portion of transaction fees paid in RDO tokens, reinforcing their role in maintaining network performance and scalability.

Reddio's tokenomics integrate developers and users into this participatory framework. Developers leverage RDO token incentives to build innovative dApps, while users engage through fee payments and governance rights, fostering a cohesive ecosystem. This structure supports Reddio's high-performance ambitions, as detailed in prior sections [1], with a full token economics overview to follow in a future publication.

## 9. Community Ecosystem and Path Forward

Reddio's ecosystem unites developers, validators, and users in a shared mission to advance blockchain technology. The RDO token and Modular Sequencer SDK empower this community, enabling developers to build innovative Layer 2 solutions with robust tools and support, while validators and users shape the network through staking and governance. Open initiatives—like hackathons and developer incentives—foster a vibrant environment for creating next-generation dApps, from DeFi to AI-driven systems.

Looking ahead, Reddio plans to explore adaptive transaction scheduling and enhanced state caching mechanisms to further improve blockchain scalability. Reddio adapts to emerging technologies and market needs through continuous upgrades and strategic partnerships across fintech, gaming, and beyond. Its scalable, secure platform, rooted in GPU-accelerated execution and decentralized sequencing, accelerates global blockchain adoption, simplifying access and enhancing performance. We invite developers, validators, and enthusiasts to join this journey—building, securing, and shaping a platform that unlocks blockchain's full promise for all [1].

# 10. References


[1] Boosting Blockchain Throughput: Parallel EVM Execution with Asynchronous Storage for Reddio, https://arxiv.org/abs/2503.04595, 2024

[2] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014):1–32, 2014.

[3] Senthil Nathan, Chander Govindarajan, Adarsh Saraf, Manish Sethi, and Praveen Jayachandran. Blockchain meets database: Design and implementation of a blockchain relational database. Proceedings of the VLDB Endowment, 12(11):1539–1552, 2019.

[4] Patrick O’Neil, Edward Cheng, Dieter Gawlick, and Elizabeth O’Neil. The log-structured merge-tree (lsm-tree). Acta Informatica, 33(4):351–385, 1996.

[5] Chenxing Li, Sidi Mohamed Beillahi, Guang Yang, Ming Wu, Wei Xu, and Fan Long. LVMT: An efficient authenticated storage for blockchain. In 17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23), pages 135–153, 2023.

[6] Manas Minglani, Jim Diehl, Xiang Cao, Binghze Li, Dongchul Park, David J Lilja, and David HC Du. Kinetic action: Performance analysis of integrated key-value storage devices vs. leveldb servers. In 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), pages 501–510. IEEE, 2017.

[7] https://github.com/yu-org/yu, 2022

[8] Manas Minglani, Jim Diehl, Xiang Cao, Binghze Li, Dongchul Park, David J Lilja, and David HC Du. Kinetic action: Performance analysis of integrated key-value storage devices vs.leveldb servers. In 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), pages 501–510. IEEE, 2017.

[9] https://github.com/sbip-sg/CuEVM, 2023