PHAROS
RESEARCH

Parallelism and DAGs: Rethinking Public Chain Consensus

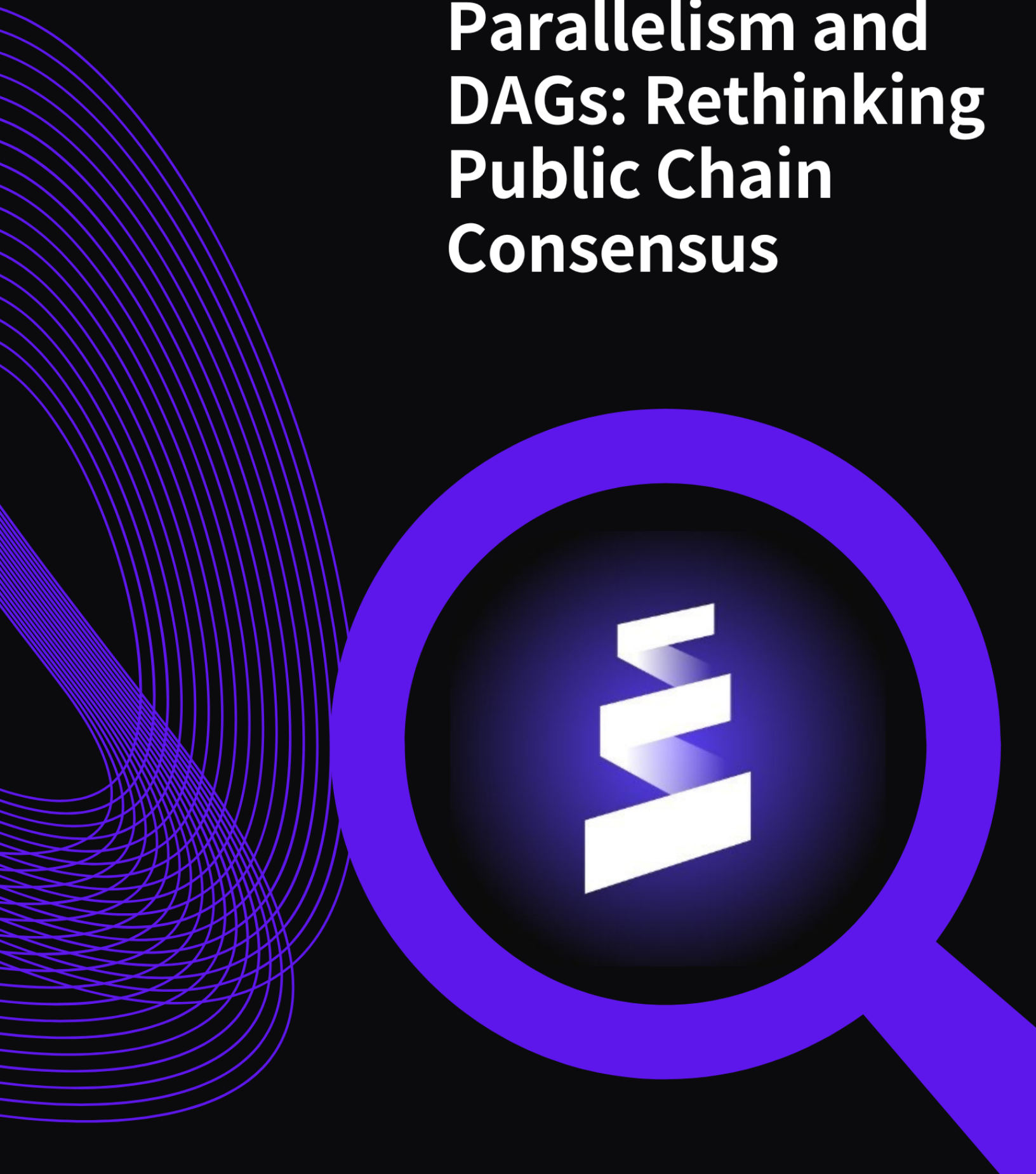


Table of Contents

Abstract

This paper aims to explore viable paths to resolving the blockchain trilemma of achieving performance, security, and decentralization simultaneously. The study first identifies the inherent on-chain serial execution model in traditional blockchain architectures as the fundamental bottleneck that limits transaction throughput (TPS), increases time to finality (TTF), and leaves multi-core computing resources underutilized, thereby hindering the large-scale adoption of public chains. To systematically evaluate emerging solutions, an analytical framework is constructed, encompassing three core dimensions: parallel execution, Directed Acyclic Graph (DAG) data structures, and modular design.

Within this framework, the paper conducts in-depth case studies of four representative systems. Pharos, through its combination of a layered DAG and hybrid consensus, establishes a closed loop of "parallel propagation, ordered persistence, and BFT finality" within a monolithic chain, striking a balance between strong consistency and low transaction finality time. Monad, a Layer 1 (L1) solution, parallelizes the Ethereum Virtual Machine (EVM) by decoupling ordering from execution and introducing a parallel pipeline, effectively shortening the critical path to state confirmation. megaETH, a Layer 2 (L2) scaling solution, employs a three-stage separation of "execution, commitment, and verification" alongside state-stream replication technology to offer users a millisecond-level interactive experience, while anchoring its economic finality to the Ethereum mainnet. Lastly, Celestia, as a specialized data availability (DA) layer, provides a scalable and secure data foundation for upper execution layers by leveraging key technologies such as Data Availability Sampling (DAS) and Namespaced Merkle Trees (NMTs).

A comparative analysis reveals that the performance metrics claimed by current solutions are often theoretical design values, and actual on-chain TPS data still lacks direct comparability. Time to Finality (TTF) and interaction latency are identified as more effective indicators for measuring user experience. It is noteworthy that aggressive parallelization designs and the pursuit of ultra-low latency often lead to significantly higher hardware requirements for nodes, creating a short-term tension where "performance supply outpaces the degree of decentralization." In contrast, the modular approach, through functional layering, successfully decouples the core issues of security and scalability, effectively mitigating inherent system-level trade-offs.

Based on these findings, this paper proposes a more promising integrated technical route: utilizing a modular DA layer as a shared security foundation while integrating a high-performance parallelization engine at the execution layer. This path not only enhances performance while maintaining compatibility with the developer ecosystem but also allows the system to progressively increase its level of decentralization along a smoother curve. Therefore, future architectural evaluations should move beyond an excessive focus on TPS metrics. Instead, they should adopt a comprehensive approach that considers a diverse set of indicators, including TTF, the maturity of development toolchains, the distribution of the validator network, and the Nakamoto Coefficient. The most suitable combination of parallelization and layered architecture should be selected based on the specific application scenario.

Keywords: Parallel Execution; Parallel EVM; DAG; Modularity; Data Availability (DA); Time-to-Finality (TTF)

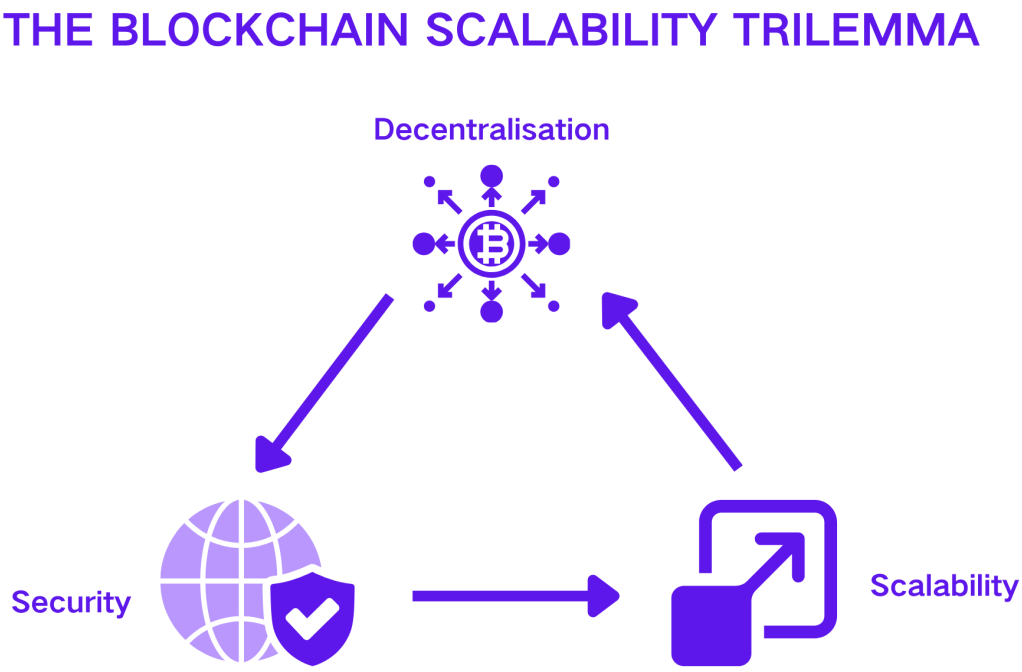
01 / The Endless Quest for the "Perfect" Blockchain

Since the genesis block of Bitcoin, blockchain technology has continuously sought a balance among performance, security, and decentralization. The first generation of consensus mechanisms, such as Proof of Work (PoW) and Proof of Stake (PoS), laid the foundation for a decentralized digital economy. However, as application scenarios have expanded, their inherent limitations have become increasingly apparent. The industry is now awakening from its fascination with single consensus mechanisms and is moving towards a profound architectural revolution aimed at breaking through traditional constraints to pave the way for mass adoption.

1.1 The Core Trade-off: The Blockchain Trilemma

The core challenge of blockchain technology can be summarized by the "Blockchain Trilemma" or the "Scalability Trilemma." Popularized by Ethereum co-founder Vitalik Buterin, this theory posits that no blockchain network can simultaneously optimize the following three core attributes; at best, it can achieve two out of the three.

Figure 1: The Blockchain Trilemma



Source: Crypto.com, Pharos Research

To delve deeper into new trends, it is essential to first precisely define these three dimensions[4]:

- Decentralization: Refers to the degree of distribution of control over the network. A highly decentralized network is not controlled by any single entity, making it resistant to censorship and single points of failure. Its measurement includes not only the number of nodes but also

their geographical distribution, the concentration of power among validators (or miners), and the openness of its governance.

- Security: Refers to the network's ability to resist attacks (especially 51% attacks) and ensure the immutability of transaction records. In a decentralized system, security relies on the robustness of cryptography and consensus mechanisms to prevent malicious activities such as double-spending.

- Scalability: Refers to the blockchain's ability to handle a growing transaction load without sacrificing performance. Key performance indicators (KPIs) include:

    - Throughput: Typically measured in Transactions Per Second (TPS), representing the number of transactions the network can process per unit of time.

    - Time-to-Finality (TTF): The time required for a transaction to be confirmed and become irreversible. This is a critical metric for user experience as it determines how long a user must wait to be certain their transaction is permanent.

Bitcoin and the early version of Ethereum are classic examples of this trade-off. Their design philosophies explicitly prioritized decentralization and security. Bitcoin, through its PoW consensus mechanism, achieved extremely high security, and its network has proven its robustness since 2009. However, the cost of this design is extremely low scalability—the Bitcoin network can only process about 7 transactions per second on average, with finality taking up to 60 minutes. Before its transition to PoS, Ethereum faced a similar dilemma, with TPS limited to 15-30. When network demand surged, transaction fees (Gas Fees) would soar to prohibitive levels, severely hindering its vision of becoming a "world computer." This inherent performance bottleneck is the fundamental driver for the entire industry's exploration of next-generation consensus mechanisms.

1.2 The Core Issue: Why Sequential Execution is a Performance Bottleneck

The Blockchain Trilemma highlights scalability as one of the core challenges in public chain design. The fundamental technical reason for the poor performance of first-generation blockchains in this dimension is their sequential processing execution model. In networks like Ethereum, transactions are like cars on a single-lane road; they must be processed and executed one by one in strict order. While this single-threaded design simplifies the maintenance of state consistency and ensures deterministic transaction outcomes, it also brings fatal flaws:

1. Throughput Ceiling: The overall TPS of the network is limited by the processing speed of a single node and cannot be linearly increased by adding more nodes.

2. Resource Waste: Modern computers are typically equipped with multi-core processors, but the sequential execution model cannot effectively utilize this parallel computing power, leading to a significant amount of idle hardware resources.

3. Network Congestion: During peak transaction periods, a large number of transactions queue up in the mempool, leading to confirmation delays and soaring transaction fees, which drastically degrades the user experience.

Therefore, breaking the shackles of sequential execution and achieving parallel execution of transactions has become the core focus of architectural design for next-generation high-performance public chains. The Blockchain Trilemma is not an absolute physical law but rather a design constraint within a specific technological paradigm. Emerging architectural innovations are challenging this inherent assumption through two mainstream paths:

- Evolutionary Path (Advanced Monolithic): Represented by new-generation high-performance public chains like Pharos, these projects do not completely deconstruct the architecture. Instead, within a single monolithic framework, they employ more advanced infrastructure such as layered DAGs and hybrid consensus to expand the boundaries of performance, security, and governance simultaneously, without sacrificing the integration of core functions.

- Deconstructive Path (Modularity): Represented by modular architectures like Celestia, this approach proposes that a single chain does not need to perform all three tasks. It outsources functions like data availability (DA) and consensus to a specialized underlying network, allowing the execution layer (Rollup) to focus on optimizing scalability. This specialization transforms the Blockchain Trilemma from an internal conflict within a single chain into a multi-layered ecosystem problem that can be synergistically optimized through system-level design.

The core challenge for both paths has evolved from "how to build an all-powerful chain" to "how to architect a system where each part is specialized and the overall efficiency is maximized." This is not just an iteration of solutions but a redefinition of the problem itself.

02 / Paradigm Shift: Key Trends Reshaping Blockchain Consensus

To break through the constraints of the Blockchain Trilemma, the industry is exploring several cutting-edge domains. This chapter will deeply analyze the three core trends of parallel execution, DAG architecture, and modular design, revealing their technical essence and strategic trade-offs through highly representative project case studies.

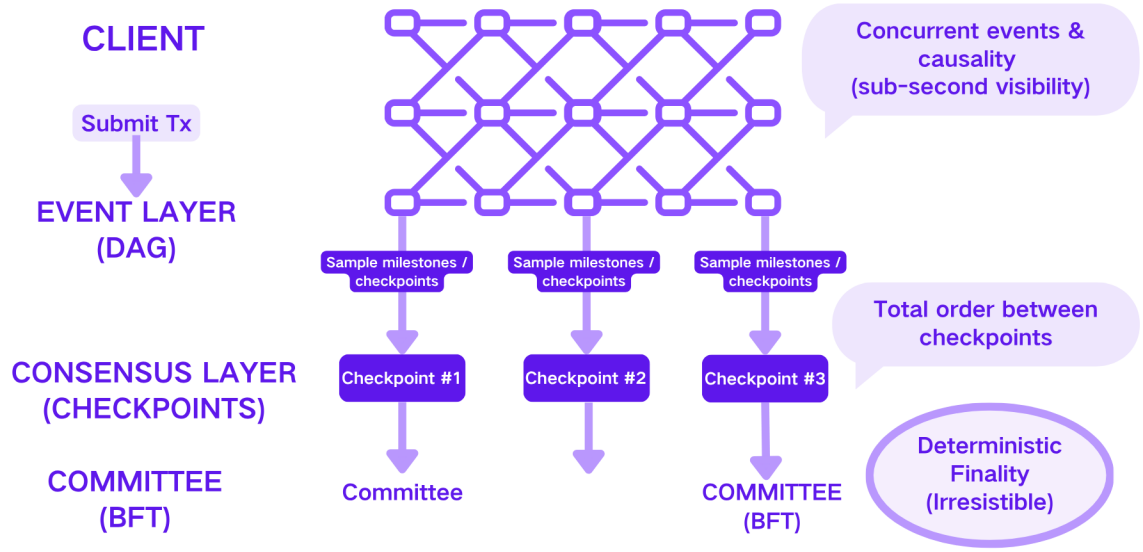
2.1 Trend One: DAG Architecture—Beyond Linear Chain Structures

In addition to innovations at the execution layer, some projects have chosen to revolutionize the more fundamental data structure layer by adopting a Directed Acyclic Graph (DAG) instead of the traditional linear blockchain structure. In a DAG, transactions (or event blocks) directly reference each other, forming a mesh-like structure rather than a single chain. This design allows for the asynchronous processing of transactions, theoretically enabling higher throughput and faster confirmation speeds.

Case Study 1: Pharos — A DAG Platform for Large-Scale Commercial Applications

Pharos, an emerging high-performance public chain project, has chosen DAG as its underlying architecture to provide infrastructure for commercial-grade applications that require high concurrency and deterministic finality.

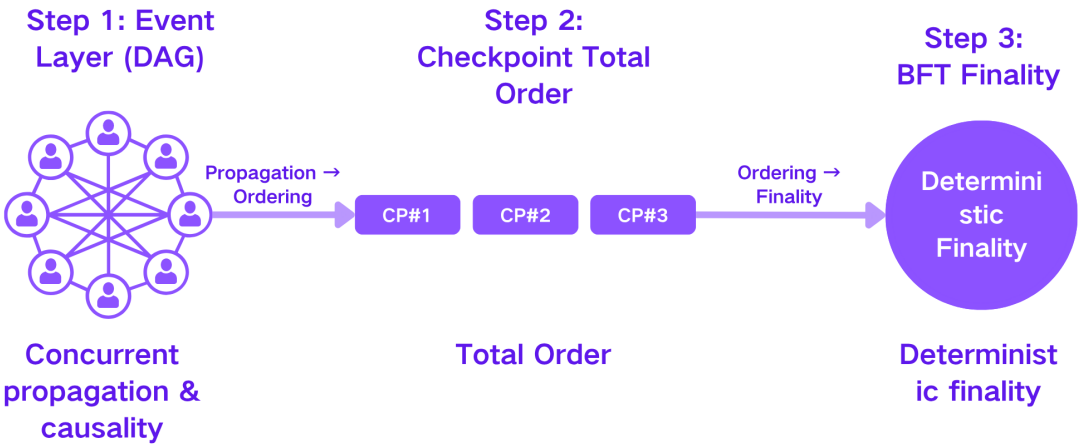
Figure 2A: A Layered DAG & BFT Checkpointing



Source: Pharos Research

Its architecture is first reflected in its layered responsibilities (see Figure 2A). In this structure, transactions initiated by a Client enter the Event Layer (DAG), where they propagate concurrently and form causal references, achieving sub-second visibility. The Consensus Layer (Checkpoints) periodically extracts "milestones/checkpoints" from the event layer and imposes a global order on the set of events between two checkpoints. Subsequently, the Committee (BFT) provides deterministic finality for this ordered interval, thus guaranteeing irreversibility and strong consistency. Based on this division of labor, the system's operational rhythm can be summarized in a three-stage process (Figure 2B): first achieving rapid visibility in the Event Layer (DAG) (Propagation), then enforcing a total order with Checkpoints (Ordering), and finally reaching deterministic finality through the BFT committee (Finality). Through this hierarchical organization of "concurrent event layer—total order checkpoints—BFT finality," Pharos significantly enhances throughput and confirmation efficiency while maintaining strong consistency.

Figure 2B: Three Stages



Source: Pharos Research

- Core Architecture:
    - Layered DAG: Unlike the relatively flat graph structures in Fantom or Hedera, Pharos is logically divided into multiple layers. The underlying Event Layer optimizes transaction ingestion and network throughput, forming a causal relationship graph. The upper Consensus Layer periodically extracts Milestones/Checkpoints, which serve as anchors to totally order the set of transactions within an interval, providing a verifiable sequential basis for finality. This layering decouples propagation from ordering, allowing the system to scale its concurrent capabilities without sacrificing consistency.
    - Hybrid Consensus Mechanism: The consensus process combines the asynchronous advantages of a DAG with the determinism of traditional BFT. The initial confirmation of a transaction depends on it being referenced by a sufficient number of subsequent events in the DAG, thus achieving rapid visibility. The final, irreversible confirmation is completed by a committee-driven BFT protocol, which makes a deterministic ruling

on the order of transactions between adjacent checkpoints. This design establishes a clear engineering division of labor and a trustworthy closed loop between speed and security.

2.2 Trend Two: Parallel Execution—Breaking the Sequential Barrier

Parallel execution is the most direct response to the bottleneck of sequential processing. Its core idea is to identify and simultaneously process non-conflicting transactions, thereby fully utilizing multi-core computing resources to significantly increase network throughput. Currently, parallel execution is mainly divided into the following schools of thought:

1. Deterministic Parallelism: This model requires transactions to declare in advance the state they will access (i.e., which accounts or smart contracts). The system can then build a dependency graph beforehand and schedule transactions with no state conflicts to be processed in parallel on different cores. Solana is a typical representative of this model.

2. Optimistic Parallelism: This model assumes that most transactions within a block will not conflict. Therefore, it first "optimistically" executes all transactions in parallel. After execution, the system detects any actual state conflicts and only re-orders and sequentially executes the conflicting transactions. This method is more developer-friendly as it does not require pre-declaration of state dependencies. Projects like Aptos have adopted this model.

3. Parallel EVM (Parallelization under EVM Semantics): Represented by Monad and megaETH, this approach attempts to "transplant" parallel execution capabilities into an EVM-compatible environment without changing EVM semantics. It introduces mechanisms such as parallel/pipelined execution, conflict re-execution, and the separation of execute-commit-verify to reduce ecosystem migration costs and enhance overall throughput.

Table 1: Comparison of Sequential vs. Parallel Execution Models		
Characteristic	**Sequential Execution**	**Parallel Execution**
Processing Model	Single-threaded, one transaction at a time.	Multi-threaded, simultaneously processes multiple non-conflicting transactions.
Resource Utilization	Cannot fully utilize multi-core CPUs.	Can effectively utilize modern multi-core processors.
Developer Burden	Low, simple model.	Deterministic model requires pre-declaration of state dependencies; optimistic model does not have this requirement.
Conflict Handling	No handling needed, sequential execution naturally avoids conflicts.	Deterministic model avoids conflicts in advance; optimistic model detects and resolves them post-execution.
Ideal Workload	Suitable for all workloads, but performance is limited.	Suitable for workloads with a large number of independent transactions; can achieve orders-of-magnitude performance improvement.
Representative Projects	Ethereum (EVM)	Solana, Aptos, Sui, Monad, megaETH
Source: Pharos Research		

Case Study 1: Monad — Achieving Parallel EVM within an L1

Monad is a parallel L1 fully compatible with the EVM. It combines MonadBFT, asynchronous execution, parallel/pipelined execution, and JIT/storage optimizations to significantly increase execution concurrency and end-to-end confirmation efficiency, all while maintaining Ethereum's semantics and developer interfaces. Public materials position it with a ~400ms block time, ~800ms finality, and 10,000 TPS[1]. Execution and ordering are pipelined to extend the available time budget per round, complemented by parallel scheduling and a specialized state database (MonadDB) to reduce storage access overhead.

- Core Architecture:
    - MonadBFT: Provides a low-latency, fork-resistant BFT consensus and explicitly decouples consensus ordering from execution. Transactions are first totally ordered by the consensus layer and then enter an independent, asynchronous execution pipeline. The execution side uses read/write sets and hotspot keys for multi-core parallel scheduling. After execution, conflict detection is performed, and only the conflicting segments undergo a minimal-scope re-execution. Finally, state changes are committed in batches to shorten the critical path.

- Asynchronous & Parallel Execution: Ordering and execution are pipelined asynchronously and combined with parallel scheduling and JIT compilation to improve throughput.

- MonadDB: The state layer is supported by MonadDB, which enhances parallel read/write throughput and smooths out persistence jitter through layered caching, asynchronous I/O, batch flushing, and snapshot mechanisms. It also plans the key space on a per-account/storage-slot basis to reduce write-write conflicts.

Case Study 2: megaETH — Achieving Parallel EVM in an L2 (Real-time Execution Layer)

megaETH is a parallel L2 for Ethereum, centered around the separation of execution, commitment, and verification. The Sequencer is responsible for parallel execution and block assembly. Replicas directly apply state updates (direct-read replication) instead of re-executing transactions. The Prover anchors the results to Ethereum using lightweight proofs or data availability solutions, creating a combined goal of "millisecond-level execution latency + L1-inherited finality." Public materials state its throughput target is ~100,000 TPS[2].

- Core Architecture:

  - Parallel EVM on L2: Introduces a Parallel EVM at L2, organized into three stages: Execution–Commit–Verify, with roles divided among the Sequencer, Replica, and Prover (or DA layer).

  - Execution and Finality Path: The Sequencer performs multi-core parallel and pipelined execution within EVM semantics, producing transaction results and state diffs. These diffs are distributed as a state stream to Replicas, which apply them directly rather than re-executing. The Prover/DA layer publishes proofs/data to anchor them on Ethereum L1 (or EigenDA), achieving traceable economic finality and a millisecond-level interactive experience.

  - Scheduling, Consistency, and Optimization: Scheduling is based on read/write sets and hotspot key heuristics, with non-conflicting transactions prioritized for concurrency and potential conflicts handled through optimistic parallelism and rollbacks. State diffs are kept orderly and consistent using version markers or height anchors. The system dynamically trades off between batch packing and stream-based committing based on L1/DA fees and bandwidth, creating a high throughput-to-cost ratio of "execute once, apply everywhere."

Monad and megaETH both fall under the category of Parallel EVM but operate at different system boundaries. The former implements a "first-order, then-execute asynchronously" parallel pipeline within an L1, maximizing the benefits of parallel read/writes through the MonadDB state layer optimization to provide low confirmation latency and stable determinism on-chain. The latter, at L2, outsources execution and consensus via a "Sequencer executes—Replica direct-read replicates—L1/DA anchors" model, combining a near-real-time interactive experience with economic finality inherited from L1. Both adhere to EVM semantic compatibility to lower the cost of application migration, but they make different engineering trade-offs regarding the source of finality, state propagation paths, and cost structures. Monad opts for an integrated on-chain design for determinism and path simplicity, while megaETH uses state stream replication and a pluggable DA layer to achieve extreme scalability in throughput and interactive latency.

2.3 Trend Three: Modularity—The Deconstruction of the Monolithic Blockchain

Modularity is one of the most disruptive concepts in blockchain architecture in recent years. It challenges the "one chain does it all" monolithic design, advocating for the separation of a blockchain's core functions—Execution, Settlement, Consensus, and Data Availability (DA)—to be handled by different specialized layers.

Case Study: Celestia—A Dedicated Data Availability Layer

Celestia is the world's first modular blockchain network focused exclusively on data availability. Through upgrades like Shwap, Celestia has significantly accelerated DAS and reduced storage overhead, creating engineering headroom for larger blocks and greater light node participation. Its mainnet is still in the Mainnet Beta phase and continues to iterate. Its core mission is singular: to reliably order transactions and prove to the entire network that the data for these transactions is accessible. Celestia itself does not execute any smart contracts; instead, it provides a secure and scalable data foundation for execution layers (such as Rollups) built on top of it.

- Core Architecture:
    - Data Availability Sampling (DAS): This is Celestia's flagship technology. It allows resource-constrained light nodes (e.g., those running on mobile phones or browsers) to verify the integrity of block data with extremely high probability (e.g., 99.9%) without downloading the entire block. Light nodes achieve this by randomly downloading and verifying tiny "samples" of the block data. This mechanism dramatically lowers the barrier to entry for participating in network validation.
    - 2D Reed-Solomon Encoding: To ensure the security of DAS, Celestia uses a technique called "erasure coding" before publishing block data. It arranges the original data into a two-dimensional matrix and calculates redundant parity data, such that even if a significant portion of the original data is lost (e.g., hidden by a malicious block producer), honest nodes can still reconstruct the complete block from the remaining data and parity data.
    - Namespaced Merkle Trees (NMTs): Celestia's block data is divided into different "namespaces," with each Rollup or application having its own dedicated space. NMTs are a special data structure that allows a Rollup application to download and verify only the data relevant to its own namespace, completely ignoring data from other applications in the block. This greatly improves efficiency and saves costs.
- Scalability Flywheel: Celestia's architecture creates a positive "scalability flywheel." The more light nodes there are on the network, the stronger their sampling capacity becomes, which in turn allows for the secure support of larger blocks. Larger blocks mean higher data throughput, which can accommodate more Rollups and transactions. This virtuous cycle enables Celestia's capacity to grow securely along with the prosperity of its ecosystem.

The emergence of modular architecture fundamentally changes the economics and development model of launching a new blockchain. In the past, creating a new Layer 1 public chain required building a validator community from scratch, attracting billions of dollars in staked capital to ensure security, and designing and implementing complex consensus mechanisms—an extremely high-

barrier process. The advent of DA layers like Celestia offers a "consensus-as-a-service" model. Developer teams can focus entirely on building their unique application logic and execution environment (e.g., a virtual machine optimized for gaming) and then "plug it in" to Celestia, much like calling an API. By paying data publication fees in TIA tokens, they can share the security guarantees provided by Celestia. This transforms blockchain development from a massive, high-risk infrastructure engineering task into a more agile, application-focused process, akin to the cloud services era, poised to foster a thriving ecosystem of numerous specialized and interoperable blockchains.

03 / Horizontal Comparison: A Data-Driven Evaluation of Next-Generation Public Chains

The superiority of a theoretical architecture must ultimately be validated by real-world data. This chapter will integrate real-time and historical data from platforms like Chainspect and Nakaflow to conduct a comprehensive horizontal comparison of the representative projects discussed earlier. The goal is to reveal their true performance in terms of performance, decentralization, and security, as well as the trade-offs behind them.

Table 2: Comparison of Key Metrics for Next-Generation High-Performance Public Chains

	Pharos	Monad	megaETH	Celestia
Core Innovation	Layered DAG & Hybrid Consensus	Parallel EVM	Parallel EVM (Execute-Commit-Verify Separation)	Modular Data Availability Layer
Consensus Mechanism	Hybrid (DAG+BFT)	MonadBFT (PoS)	Composable (Paired with DA/Settlement Layer)	CometBFT (PoS)
Theoretical Peak TPS	> 10,000	10,000	10,000	N/A (DA Layer)
Real-time TPS	Approx. 247 [5]	Approx. 142 [6]	Approx. 133 [7]	N/A (DA Layer, not applicable)
Time to Finality (TTF)	< 2s	~0.8s (400ms Block Time, 800ms Finality)	1 – 10 ms execution latency (Economic finality inherited from Ethereum L1)	N/A (DA Layer)
Number of Active Validators	< 50 (Permissioned)	N/A	N/A	~100

Note: The TPS metrics and statistical windows for different projects/dashboards are not entirely consistent, and are mostly testnet readings. For comparison purposes only.
Source: Pharos Research

3.1 Performance and Scalability Analysis

The data in the table reveals a clear trend: the representative chains currently operate primarily on "theoretical values," and real-time TPS has not yet reached a state of stable, comparable public figures. Pharos specifies a theoretical TPS of >10,000 and a TTF of <2 seconds[3], emphasizing the establishment of a parallel closed loop for "propagation-ordering-finality" within a single chain, achieved through a layered Directed Acyclic Graph (DAG) combined with a hybrid consensus mechanism. Monad targets a theoretical TPS of ~10,000 with a finality of ~0.8 seconds (400ms

block time / 800ms finality), attempting to compress the confirmation path within an L1 by decoupling ordering and execution in its Parallel EVM. megaETH, on the L2 track, aims for an execution latency of 1–10ms and a near-real-time interactive experience, while anchoring its economic finality to Ethereum L1. As a DA layer, Celestia does not directly produce comparable TPS or TTF metrics; its value is demonstrated by providing a scalable data publication and availability guarantee for upper-layer execution layers (Rollups / Parallel EVM) through mechanisms like DAS and NMTs.

Compared to the traditional "TPS-first" narrative, this set of metrics places a greater emphasis on "low-latency finality + system-level scalability." Parallel EVM (L1/L2) solutions are more aggressive in their end-to-end confirmation times. The DAG approach seeks stable, second-level finality within a single chain. The DA approach indirectly magnifies system throughput by "hosting more execution layers." It is important to note that the absence of real-time TPS data does not imply a lack of capability; rather, it is more a consequence of current network load, client maturity, hardware barriers, and inconsistent monitoring standards. At this stage, TTF and interaction latency often reflect user experience and engineering maturity more accurately than a single TPS figure.

3.2 Decentralization and Security Analysis

The tension between decentralization and performance manifests differently across these approaches. As shown in Table 2, Pharos currently employs a permissioned validator network (fewer than 50 validators) to serve Real-World Asset Finance (RWAFi) and enterprise-grade DeFi scenarios by providing clear accountability and predictable performance. Celestia has approximately 100 active validators, reflecting the growing adoption of DA networks for open participation and availability guarantees. Monad and megaETH have not yet provided stable figures for active validators or the Nakamoto Coefficient, as they are still in a phase of rapid architectural and implementation evolution.

From a macro perspective, aggressive parallelization and the pursuit of low latency tend to raise the computational and bandwidth requirements for nodes. This can, in the early stages, limit the set of potential participants, leading to a structural contradiction where "performance supply outpaces decentralization." The rise of modular DA, by decoupling execution from data publication, allows the network to advance decentralization at different paces on different layers.

Regarding security boundaries, L1 Parallel EVM (Monad) tends to complete ordering and finality directly on-chain, resulting in a short and clear path to determinism. L2 Parallel EVM (megaETH) primarily relies on a layered security model of "millisecond-level execution + L1 economic finality," where execution is fast but finality depends on the anchor layer. DAG (Pharos) provides determinism and a low TTF through checkpointed BFT. DA (Celestia) treats "data availability" as a public good for system security. Overall, these projects make different choices regarding the Blockchain Trilemma: Parallel EVM leans more towards latency and throughput, DAG emphasizes single-chain consistency and determinism, and DA focuses on system-level scalability and open participation. In the foreseeable future, the projects that can establish a sustainable growth trajectory balancing finality experience, operational complexity, and the decentralization curve will be better positioned to convert their theoretical advantages into long-term network competitiveness.

04 / Conclusion and Outlook: The Future Landscape of Consensus Mechanisms

After an in-depth analysis of the three major trends—parallel execution, DAG architecture, and modularity—it is clear that the evolution of public chain consensus mechanisms has moved beyond a monolithic era and into a new epoch of diversification, specialization, and combinatorial innovation. These trends are not mutually exclusive; rather, they represent a coordinated, multi-dimensional assault on the Blockchain Trilemma.

4.1 Summary of Trends: A Multi-Front War on the Trilemma

In summary, the consensus innovations of next-generation high-performance public chains exhibit the following core characteristics:

- Optimization from Computation to Communication: Whether it's parallel execution engines or the "virtual voting" in DAGs, the essence is to optimize the communication and coordination patterns among nodes, reduce unnecessary consensus overhead, and allocate more resources to processing actual transactions.
- Specialization and Division of Labor: The rise of modular architecture marks a shift in the industry from pursuing "all-in-one" monolithic chains to building "collaborative" ecosystems composed of specialized components. This allows each layer to focus on solving specific problems, thereby breaking through the limitations of a single architecture on a system-wide level.
- Emphasis on Developer Experience: Aptos's optimistic parallelism, Sui's object model, and the convenience Celestia provides for Rollup developers all demonstrate that the new generation of public chains is placing developer experience at the core of their strategy, aiming to lower the barrier to innovation.

4.2 Future Outlook: The Promising Fusion of Modularity and Parallel Execution

Looking ahead, the most promising direction lies in the deep integration of modularity and parallel execution. An ideal future high-performance blockchain stack might look like this:

- Foundation Layer: A highly decentralized and secure Data Availability (DA) and consensus layer. A modular DA layer, similar to Celestia, would provide a trusted data foundation and shared security for the entire ecosystem through technologies like Data Availability Sampling (DAS).
- Execution Layer: Multiple specialized Rollups employing parallel execution engines. These execution layers could be customized for specific application scenarios. For example, a

Rollup designed for DeFi might adopt a Parallel EVM architecture to be compatible with the existing ecosystem, while a Rollup for gaming or social applications might use an object model like Sui's or a system like Aptos's Block-STM to achieve ultimate performance and state management capabilities.

This combination of a "modular foundation + parallelized engine" can skillfully resolve the inherent contradictions of the Blockchain Trilemma. The foundation layer focuses on ensuring security and decentralization, while the execution layer can offload the burden of consensus to focus entirely on scalability. This architecture not only theoretically enables higher overall performance but also provides unprecedented flexibility and possibilities for the diversification and mass adoption of blockchain applications.

4.3 Recommendations for Industry Participants

4.3.1 Recommendations for Developers

In selecting a technology stack, it is crucial to overcome the singular pursuit of "theoretical TPS figures." Instead, a more comprehensive set of evaluation benchmarks should be adopted, including Time to Finality (TTF), end-to-end interaction latency, toolchain maturity, and the semantic alignment with the target business logic. Execution environments centered on an object model (e.g., Sui) are better suited for high-concurrency, strongly isolated scenarios such as NFT and gaming assets. In contrast, parallelization solutions that preserve EVM semantics (e.g., Pharos, Monad, megaETH) balance ecosystem compatibility with performance enhancements, which is more conducive to the reuse of existing smart contracts and infrastructure.

Simultaneously, a modular mindset should be actively embraced. Deploying applications as sovereign rollups or choosing a suitable combination of settlement and Data Availability (DA) layers can provide greater freedom for customization and evolution under the premise of shared security. This approach allows for the gradual enhancement of the stack's internal capabilities and risk resilience along a trajectory encompassing the toolchain, operations, and governance.

4.3.2 Recommendations for Investors

Project evaluation should shift from the "high-TPS narrative" back to the fundamentals of security and decentralization. Key metrics to scrutinize include the Nakamoto Coefficient (NC), the number and distribution of active validators, client diversity, and the barrier to entry for light node participation. It is also crucial to track the feasibility and phased achievements of a project's "decentralization roadmap." Furthermore, investors must understand the engineering trade-offs each project makes within the Blockchain Trilemma. For instance, low-latency parallel execution often raises hardware requirements, which can limit the pool of potential participants in the early stages, while modularity decouples security and scalability, resolving trade-offs through system-level coordination. Investment decisions should be based on the alignment between a team's technical roadmap and its target market, assessing the achievable path and milestones for converting "theoretical advantages" into "network effects and cash flow" within a 12–24 month timeframe.

4.3.3 Recommendations for the Industry

The industry narrative should evolve from the zero-sum competition of "L1 killers" to a more collaborative "multi-chain, multi-layer ecosystem." Future value is more likely to emerge from standardized cross-chain communication, shared security frameworks, and combinatorial innovation among pluggable DA, settlement, and execution components, rather than from the monopoly of a single chain. To this end, the industry should promote interoperability standards for messaging and proof formats, encourage the adoption of light-node-friendly protocols (like DAS), and establish replicable best practices for regulatory compliance, data availability, and privacy protection. Only when the execution, settlement, and data layers evolve in concert with clear interfaces can developers and capital achieve scalable innovation and sustainable growth on a more predictable track.

In conclusion, the revolution in consensus mechanisms is far from over. We are at an exciting inflection point, moving from monolithic competition to modular collaboration, and from sequential execution to a parallel era. The projects and participants who can profoundly understand and harness these new paradigms will undoubtedly gain a decisive advantage in the race to define the future of blockchain.

References

[1] Monad, https://docs.monad.xyz

[2] Github, https://github.com/megaeth-labs

[3] Pharos, https://www.pharosnetwork.xyz/blog/pharos-testnet-is-live-sailing-towards-rwa-adoption

[4] Nervos, https://www.nervos.org/knowledge-base/blockchain_trilemma

[5] Pharosscan, https://testnet.pharosscan.xyz

[6] Monadexplorer, https://testnet.monadexplorer.com

[7] oklink, https://www.oklink.com/megaeth-testnet

Appendix

Term	Abbreviation	Explanation
Blockchain Trilemma / Scalability Trilemma	—	Refers to the structural trade-off in blockchain design where it is difficult to simultaneously optimize decentralization, scalability, and security.
Decentralization	—	The degree of distribution of network control, encompassing node count, geographic distribution, stake concentration, and governance openness.
Security	—	The network's ability to resist malicious actions (e.g., 51% attacks, censorship, double-spending) and maintain immutability.
Scalability	—	The ability to expand transaction load and state size without sacrificing security and decentralization.
Throughput	TPS	The number of transactions processed per unit of time (Transactions Per Second). The metric often varies with the statistical window.
Time to Finality	TTF	The time required for a transaction to become irreversible. The most relevant user experience metric for perceived latency.
Finality (Deterministic/Economic)	—	Deterministic finality is irreversibility directly provided by BFT/checkpoints; economic finality relies on the cost and game-theoretic guarantees of an upper layer (e.g., L1).
Block Time	—	The target interval for block generation; related to but not synonymous with TTF.
Directed Acyclic Graph	DAG	A ledger/event structure that uses a graph instead of a linear chain, supporting asynchronous propagation and concurrent confirmation.
Layered DAG	—	A structure that organizes event propagation and total ordering/finality into layers (e.g., Event Layer + Checkpoint/BFT) to balance concurrency and consistency.
Checkpoint / Milestone	—	An anchor point that imposes a global order on a set of events within an interval, facilitating finality decisions and traceability.
Byzantine Fault Tolerance	BFT	A consensus paradigm that enables a system to reach agreement despite a portion of nodes failing or acting maliciously.
Hybrid Consensus	—	A composite consensus mechanism that combines the asynchronous propagation advantages of a DAG with the deterministic finality of BFT.
CometBFT	—	The successor to Tendermint, a typical PoS BFT consensus engine used by projects like Celestia.
MonadBFT	—	Monad's PoS+BFT consensus, emphasizing low-latency ordering and decoupling from execution.
Proof of Work	PoW	A consensus mechanism where block creation rights are won through computational work; secure but has limited throughput and high energy consumption.
Proof of Stake	PoS	A block creation/voting mechanism weighted by staked equity; energy-efficient and commonly combined with BFT.
Parallel Execution	—	The simultaneous processing of multiple transactions that do not have state conflicts, fully utilizing multi-core resources to improve throughput and reduce latency.
Deterministic Parallelism	—	A model where transactions pre-declare their read/write sets, allowing the system to schedule non-conflicting transactions in parallel based on dependencies, thus avoiding conflicts.
Optimistic Parallelism	—	A model that executes transactions in parallel first, then detects conflicts afterward and locally re-executes/re-orders, reducing the burden on developers.
Parallel EVM	—	Solutions (e.g., Monad, megaETH) that introduce parallel/pipelined execution and conflict re-execution without changing EVM semantics.
Pipelined Execution	—	A technique that breaks down the order-execute-commit/verify process into stages that are processed in parallel, improving hardware utilization and end-to-end efficiency.
Read/Write Set	—	The set of state keys a transaction accesses; fundamental for parallel scheduling and conflict detection.
Conflict Re-execution	—	When a state conflict is detected after parallel execution, the involved transactions are locally re-executed sequentially to restore consistency.
State Diff	—	The set of state changes produced by execution; used for replication, fast application, and proof generation.
State Stream Replication	—	A method where state diffs produced by a Sequencer are streamed to replica nodes, which apply them directly instead of re-executing (e.g., megaETH).
Mempool	—	A cache for pending transactions that have not yet been included in a block; congestion can lead to fee spikes and confirmation delays.
Execution Layer	—	The layer responsible for transaction execution and state machine updates (L1 or L2/Rollup).
Settlement Layer	—	The layer responsible for settling state and resolving disputes between different execution environments.
Consensus Layer	—	The layer responsible for transaction ordering and reaching consensus; can be decoupled from the execution layer.
Data Availability	DA	The guarantee that data published on-chain can be accessed and verified by anyone; a prerequisite for Rollup security.
Data Availability Sampling	DAS	A technique allowing light nodes to verify the integrity of a block with high probability by randomly sampling small pieces of it, supporting larger blocks and more light nodes.

Namespaced Merkle Tree	NMT	A data commitment structure indexed by namespaces, enabling applications to verify only the data subset relevant to them.
2D Reed–Solomon Encoding	2D RS	A technique that uses erasure codes to redundantly encode block data in a 2D matrix, ensuring it can be recovered even if parts are hidden or lost.
Rollup	—	A scaling solution that performs execution off-chain (L2) but posts data/proofs to an underlying layer (e.g., Ethereum or Celestia) for security and scalability.
Sovereign Rollup	—	A Rollup that operates and governs itself independently on a DA layer, not reliant on a single L1 for settlement and governance.
Sequencer	—	A role responsible for collecting, ordering, batching, and (in some designs) executing transactions in parallel.
Light Node	—	A node that does not store the full state but verifies security through sampling/proofs; has a low barrier to entry, promoting decentralization.
Nakamoto Coefficient	NC	The minimum number of independent entities required to compromise the network; a higher number indicates greater decentralization.
Gas Fee	—	The unit for pricing the resources used for transaction execution and data publication; significantly affected by congestion and scaling strategies.
Blob	—	A large payload dedicated to data publication (e.g., Ethereum's EIP-4844 blobs or a DA layer's blobspace).
EVM Compatibility	—	Maintaining consistency with the EVM's semantics and developer interfaces, facilitating ecosystem migration and tool reusability.

Contributors

Authors: pixelpanda (𝕏@realgc193222)

Reviewers: Colin Su, Grace Gui, NingNing, Owen Chen

Design: Alita Li

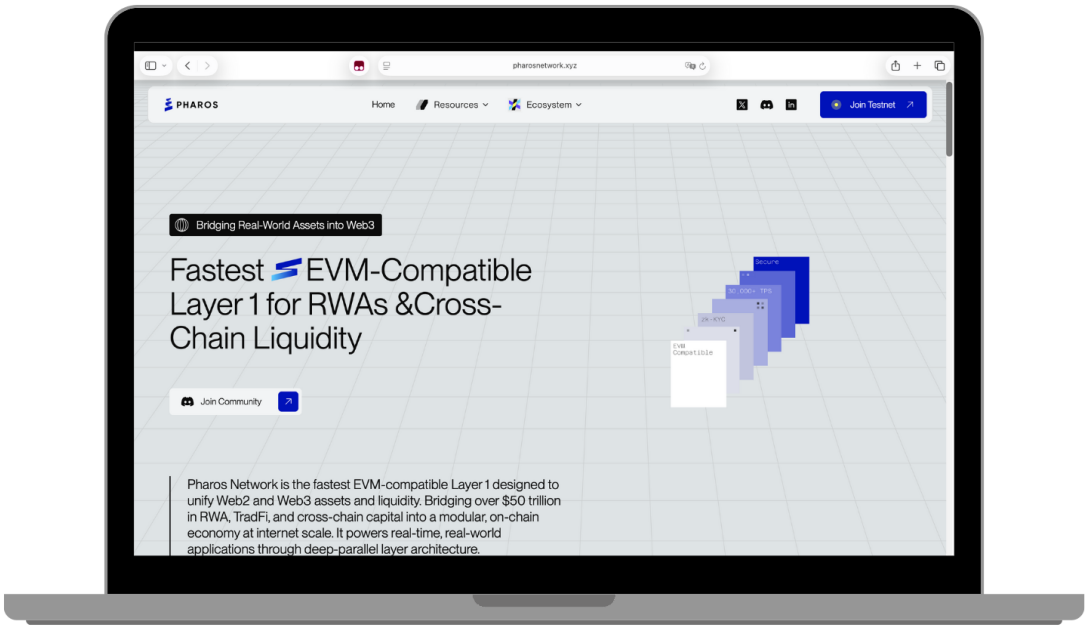
Disclaimer

This material is prepared by Pharos Research for the purpose of providing general information. It does not constitute and should not be deemed as investment, legal, accounting, or tax advice, nor does it form an offer, solicitation, or recommendation with respect to any securities, cryptographic assets, or strategies. The information and opinions contained herein may be derived from internal or third-party sources. While efforts are made to ensure their reliability, their accuracy, completeness, or timeliness is not guaranteed. Any decisions made and risks arising therefrom shall be borne solely by the reader. Past performance is not indicative of future results. This material may contain forward-looking statements (including forecasts and scenarios), which are subject to uncertainties and not guaranteed to be achieved. Cryptographic assets are highly volatile, and total loss may occur. They are also exposed to risks such as liquidity, technology, smart contract, counterparty, and compliance risks. To the extent permitted by law, the Research Institute and/or its affiliates or researchers may hold positions in the relevant assets, have business relationships with relevant entities, or otherwise have interests that may affect the objectivity of opinions. This material is not intended for persons in restricted jurisdictions. Reading, following, or subscribing to this material does not constitute a client relationship. Without prior written permission, no institution or individual may reproduce, copy, modify, or distribute this material. Any quotation shall be objective and complete, with the source clearly credited as "Pharos Research".

Contact

Pharos Network is a next-generation public blockchain for Real-World Assets (RWA) and stablecoins, focused on asset tokenization and on-chain circulation. We connect traditional institutions with the Web3 ecosystem, enrich the types of on-chain assets, expand revenue sources, and meet the allocation needs of a broader range of investors. Meanwhile, we help traditional enterprises unlock sustainable value on-chain through customized solutions. Boasting profound professional expertise and top-tier technical capabilities, our team builds a secure, efficient, and scalable infrastructure, providing institutions with a comprehensive decentralized ecosystem for onboarding assets onto the blockchain. We welcome strategic partners with a long-term perspective to co-build an open, compliant, and sustainable RWA ecosystem. For industry exchanges with us, please contact: chris@pharoslabs.xyz

Pharos' Official Website: https://www.pharosnetwork.xyz/



WeChat Official Account: Pharos Research

微信搜一搜

Pharos Research

PHAROS
RESEARCH

From RWA to On-Chain Finance.
Mapping Real-World Value.