

Ethereum's shift to a [rollup-centric roadmap](#) induced explosive growth in modular scaling designs. This growth was initially a success, eliminating \$100+ gas fees and unlocking entirely new application designs. But only a few short years later, Ethereum and its rollups face a new, critical problem: state fragmentation.

Fundamentally, state fragmentation is a scaling problem. Recently, the modular community has pushed middleware solutions that merge existing rollups into a single system, (seemingly) achieving the holy grail of blockchain scaling - horizontal scalability. However, these solutions come with significant trade-offs. A new generation of Ethereum L2s is rethinking scalability from first principles, applying vertical and horizontal techniques to deliver end-game performance.

## Blockchain Scaling (kinda-not-really)

There are two frameworks for scaling a blockchain:

### 1. Vertical scaling

increases the resources available to a single node. This is Solana's scaling philosophy of maximizing hardware utilization, as well as the parallelized VM movement.

### 1. Horizontal scaling

distributes the workload across multiple servers. This is the scaling philosophy of sharded L1 blockchains such as Near and Ethereum (Danksharding).

Rollups are often mistakenly considered a horizontal scaling solution for Ethereum. However, each rollup, and every blockchain for that matter, is defined by the ledger it maintains, meaning, rollups are separate systems from Ethereum. This fundamental oversight of basic database scaling principles has left the Ethereum ecosystem with an existential challenge to solve: state fragmentation.

## The Challenge Ahead

State fragmentation across L2s has spiraled into a major problem for Ethereum. Fragmentation is defined by three new problems.

1. Compromised network effects of global state, including liquidity and user distribution, is rooted in the tens (if not [hundreds](#)) of Ethereum adjacent L2s that push for app-specific infrastructure.
2. Compounding complexity of interoperability standards and social coordination, which has been the catalyst of [over \\$2.8 billion of bridging hacks](#).
3. Objectively worsened developer and user experiences. Developers must work with convoluted network abstractions to offer a seamless user experience, or

users are forced to manage the complexity of interacting cross-chain, including asset bridging, wallet network management, and multiple gas tokens.

And these problems are getting worse by the day. Limited by existing infrastructure, price-sensitive applications are forced into isolation to maintain reliably low transaction fees. As the next cycle approaches, a vicious snowball effect is set to take place; as L2 congestion fees increase, more developers are forced to opt for app-specific infrastructure, exacerbating the (already) pervasive problems associated with state fragmentation. In a few years, it would not be a surprise if the inability of L2s to solve for state fragmentation led to the downfall of the Ethereum ecosystem's application dominance.

## Solving State Fragmentation

State fragmentation is fundamentally a scaling problem in which the onus remains on the L2 to scale without fracturing composability. There are two approaches L2s can take to solve for scalability.

1. Merge existing rollups into a single system, thus achieving "horizontal scalability".
2. Start fresh and design a scalable L2.

## Working Backwards

The first approach is quite popular amongst incumbent L2s. The merging of rollups is achieved by using middleware to establish a notion of a single system. Effectively, these solutions facilitate communication across rollups through shared consensus guarantees. Such solutions include shared sequencers, shared provers, and various L3 architectures.

While the teams and projects working on these solutions are strong, a middleware-centric approach to solving L2 scalability comes with major tradeoffs, including:

1. Weakest chain vulnerabilities: the security and properties of the network are inherited from the worst piece of the

stack.

2. A fragmented developer environment: differing environments tied together with middleware still results in a fractured development experience.
3. Global social coordination of cross-rollup standards: to merge existing rollups, hundreds (potentially thousands) of stakeholders need to agree on best practices.

More critically, it distracts L2 teams from solving the open problems of congestion fee pricing and single-actor censorship, which require significant engineering and research efforts.

## A Fresh Start

Ethereum L2s can be scaled vertically by altering the execution environment of a rollup node to increase hardware utilization; such projects include Eclipse and Movement Labs who are building rollups utilizing the SVM and MoveVM, respectively. This approach holds high promise for near-term scalability improvements; however, it requires Ethereum developers to adopt a new tech stack.

Alternatively, L2s can horizontally scale by (re)-introducing execution sharding, which would allow the network to scale by adding new nodes. This approach promotes decentralization, has higher theoretical scaling limits, and allows for vertical scaling optimizations if necessary. Given these advantages, =nil; Foundation has designed a sharded L2 called =nil;.

=nil; optimizes to preserve Ethereum's core values of decentralization, censorship resistance, and permissionless-ness. =nil; is the first verifiable sharding architecture based on a novel design, zkSharding

. It enables the scaling properties of the above post-facto horizontal scaling frameworks with the added benefit of a single, integrated development environment. This gives developers access to the scale of 1000s of rollups from a single network. More importantly, =nil; ensures users reliably low transaction fees even amidst peak transaction load.

Furthermore, =nil; solves congestion fees by dynamically splitting and merging state across shards based on the demand for state access. This dynamic behavior allows =nil; to keep transaction fees reliably low (<\$0.01). Altogether, =nil; Foundation's mission is to offer an alternative path for Ethereum L2 scaling that more tightly aligns with Ethereum's core values and the demand for L2 execution.

## Closing Thoughts

Despite the many challenges ahead, the future for Ethereum L2s looks more promising than ever. As L2 designs mature and we enter the next generation of scaling solutions, there are two prevailing divides: working backward vs. starting fresh, and horizontal vs. vertical scaling. =nil; Foundation is excited to apply over five years of research and engineering towards a fresh start on horizontal scalability. For more information, check out =nil; Foundation's [zkSharding announcement](#).

Sharding is dead. Long live sharding.

Special thanks to Walt Smith (cyberfund), Sven Williamson and Luke Pearson (Polychain), Jonah Burian, Yuan Han Li, Anay Simunovic (Blockchain Capital)