

Monad for users

Monad is a high-performance Ethereum-compatible L1, offering users the best of both worlds: portability and performance.

From a portability perspective, Monad offers full bytecode compatibility for the Ethereum Virtual Machine (EVM), so that applications built for Ethereum can be ported to Monad without code changes, and full Ethereum RPC compatibility, so that infrastructure like MetaMask or Etherscan can be used seamlessly.

From a performance perspective, Monad offers 10,000 tps of throughput, i.e. 1 billion transactions per day, while offering 1 second block times and 1 second finality. This allows Monad to support many more users and far more interactive experiences than existing blockchains, while offering far cheaper per-transaction costs.

What's familiar about Monad?

From a user perspective, Monad behaves very similarly to Ethereum. You can use the same wallets (e.g. MetaMask) or block explorers (e.g. Etherscan) to sign or view transactions. The same apps built for Ethereum can be ported to Monad without code changes, so it is expected that you'll be able to use many of your favorite apps from Ethereum on Monad. The address space in Monad is the same as in Ethereum, so you can reuse your existing keys.

Like Ethereum, Monad features linear blocks, and linear ordering of transactions within a block.

Like Ethereum, Monad is a proof-of-stake network maintained by a decentralized set of validators. Anyone can run a node to independently verify transaction execution, and significant care has been taken to keep hardware requirements minimal.

What's different about Monad?

Monad makes exceptional performance possible by introducing parallel execution and superscalar pipelining to the Ethereum Virtual Machine.

Parallel execution is the practice of utilizing multiple cores and threads to strategically execute work in parallel while still committing the results in the original order. Although transactions are executed in parallel "under the hood", from the user and developer perspective they are executed serially; the result of a series of transactions is always the same as if the transactions had been executed one after another.

Superscalar pipelining is the practice of creating stages of work and executing the stages in parallel. A simple diagram tells the story:

?

When doing four loads of laundry, the naive strategy is to wash, dry, fold, and store the first load of laundry before starting on the second one. The pipelined strategy is to start washing load 2 when load 1 goes into the dryer. Pipelining gets work done more efficiently by utilizing multiple resources simultaneously.

Monad introduces pipelining to address existing bottlenecks in state storage, transaction processing, and distributed consensus. In particular, Monad introduces pipelining and other optimizations in four major areas:

- [MonadBFT](#)
- (pipelined HotStuff consensus with additional research improvements)
- [Deferred Execution](#)
- (pipelining between consensus and execution to significantly increase the execution budget)
- [Parallel Execution](#)
- [MonadDb](#)
- (high-performance state backend)
-

Monad's client, which was written from scratch in C++ and Rust, reflects these architectural improvements and results in a platform for decentralized apps that can truly scale to world adoption.

Why should I care?

Decentralized apps are replacements for centralized services with several significant advantages:

- Open APIs / composability
- : decentralized apps can be called atomically by other decentralized apps, allowing developers to build more complex functionality by stacking existing components.
- Transparency
- : app logic is expressed purely through code, so anyone can review the logic for side effects. State is transparent and auditable; proof of reserves in DeFi is the default.
- Censorship-resistance and credible neutrality:
- anyone can submit transactions or upload applications to a permissionless network.

- Global reach
- : anyone with access to the internet can access crucial financial services, including unbanked/underbanked users.
-

However, decentralized apps need cheap, performant infrastructure to reach their intended level of impact. A single app with 1 million daily active users (DAUs) and 10 transactions per user per day would require 10 million transactions per day, or 100 tps. A quick glance at [EthTPS.info](#) - a useful website summarizing the throughput of existing EVM-compatible L1s and L2s - shows that no EVM blockchain supports even that level of throughput right now.

Monad materially improves on the performance of an EVM-compatible blockchain network, pioneering several innovations that will hopefully become standard in Ethereum in the years to come.

With Monad, developers, users, and researchers can reuse the wealth of existing applications, libraries, and applied cryptography research that have all been built for the EVM.

How do I use Monad?

Monad's first public testnet will go live in the coming months.

When it does, you will be able to add an appropriate RPC url and ChainId to your Ethereum-compatible wallet and begin using Monad like you would any other Ethereum-compatible network. Until then, stay tuned!

[Previous Briefings](#) [Next Monad for developers](#) Last updated 5 months ago

On this page * [What's familiar about Monad?](#) * [What's different about Monad?](#) * [Why should I care?](#) * [How do I use Monad?](#)