

Monolithic vs. modular blockchains

Blockchains instantiate [replicated state machines](#) : the nodes in a permissionless distributed network apply an ordered sequence of deterministic transactions to an initial state resulting in a common final state.

In other words, this means that nodes in a network all follow the same set of rules (i.e. , an ordered sequence of transactions) to go from a starting point (i.e. , an initial state) to an ending point (i.e. , a common final state). This process ensures that all nodes in the network agree on the final state of the blockchain, even though they operate independently.

This means blockchains require the following four functions:

- Execution
 - entails executing transactions that update the state correctly. Thus, execution must ensure that only valid transactions are executed,i.e.
 - , transactions that result in valid state machine transitions.
- Settlement
 - entails an environment for execution layers to verify proofs, resolve fraud disputes, and bridge between other execution layers.
- Consensus
 - entails agreeing on the order of the transactions.
- Data Availability
 - (DA) entails making the transaction data available. Note that execution, settlement, and consensus require DA.

Traditional blockchains,i.e. monolithic blockchains , implement all four functions together in a single base consensus layer. The problem with monolithic blockchains is that the consensus layer must perform a lot of different tasks and it cannot be optimized for only one of these functions. As a result, the monolithic paradigm limits the throughput of the system.

As a solution, modular blockchains decouple these functions among multiple specialized layers as part of a modular stack. Due to the flexibility that specialization provides, there are many possibilities in which that stack can be arranged. For example, one such arrangement is the separation of the four functions into three specialized layers.

The base layer consists of DA and consensus and thus, is referred to as the Consensus and DA layer (or for brevity, the DA layer), while both settlement and execution are moved on top in their own layers. As a result, every layer can be specialized to optimally perform only its function and thus, increase the throughput of the system. Furthermore, this modular paradigm enables multiple execution layers,i.e. , [rollups](#) , to use the same settlement and DA layers. [[Edit this page on GitHub](#)] Last updated: [Previous page Introduction](#) [Next page Celestia's data availability layer](#) []