

Why blockchain?

A simple mental model for the 'what' and 'why'. A blockchain is decentralized agreement among a diverse set of participants about two things:

1. An official ordering (ledger) of transactions
2. An official state of the world, including balances of accounts and the state of various programs.
- 3.

In modern blockchains such as Ethereum, transactions consist of balance transfers, creation of new programs, and function calls against existing programs. The aggregate result of all transactions up to now produces the current state, which is why agreement about (1) implies agreement about (2).

A blockchain system has a set of protocol rules which describe how a distributed set of nodes which are currently in sync will communicate with each other to agree upon a new list of transactions that each should apply. Induction keeps the nodes in sync: they start with the same state and apply the same transactions, so at the end of applying a new list of transactions, they still have consistent state. (This essay will ignore the details of how such a system of nodes achieves agreement, but you can refer to the documentation of Monad's [consensus mechanism](#) for further details.)

Shared global state enables the development of decentralized apps - apps that live "on the blockchain", i.e. on each of the nodes in the blockchain system. A decentralized app is a chunk of code (as well as persistent, app-specific state) that can get invoked by any user, who does so by submitting a transaction pointing to a function on that app. Each of the nodes in the blockchain is responsible for correctly executing the bytecode being called; duplication keeps each node honest.

An example app

Decentralized apps can implement functionality that we might otherwise expect to be implemented in a centralized fashion. For example, a very simple example of a decentralized app is a Virtual Bank (typically referred to in crypto as a Lending Protocol).

In the physical world, a bank is a business that takes deposits and loans them out at a higher rate. The bank makes the spread between the high rate and the low rate; the borrower gets a loan to do something economically productive; and you earn interest on your deposits. Everyone wins!

A Virtual Bank is simply an app with four major methods: deposit, withdraw, borrow, and repay. The logic for each of those methods is mostly bookkeeping to ensure that deposits and loans are being tracked correctly:

...

Copy class VirtualBank: def deposit(sender, amount):

transfer amount from sender to myself (the bank)

do internal bookkeeping to credit the sender

def withdraw(sender, amount):

ensure the sender had enough on deposit

do internal bookkeeping to debit the sender

transfer amount from myself (the bank) to sender

def borrow(sender, amount):

...

def repay(sender, amount);

...

...

In Ethereum, or in Monad, someone can write code for this Virtual Bank and upload it; then anyone can utilize it for borrowing and lending, potentially far more easily than when trying to get access to banking services in their home country.

This simple example shows the power of decentralized apps. Here are a few other benefits to call out:

- 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.
-

[Previous Asynchronous I/O](#) [Next Why Monad: decentralization + performance](#) Last updated 5 months ago

On this page