

# IPFS and the problems it solves

4-5 minutes

---

IPFS seeks to address problems with the current web and existing data representation / routing / transfer protocols like HTTP including:

- [Verifiability](#)
- [Resilience](#)
- [Centralization](#)
- [Performance](#)
- [Link rot](#)
- [Data sovereignty and ownership](#)
- [Off-chain storage](#)
- [Local-first software](#)
- [Vendor lock-in](#)

In this guide, you'll learn about each of the problems that IPFS solves.

This guide is part 2 of a 3-part introduction to [the basic concepts of IPFS](#). The third part, [How IPFS Works](#), covers the different subsystems that IPFS is composed of and how each one functions.

## # Verifiability

IPFS uses cryptographic hashes to verify the authenticity and integrity of files, making it difficult for malicious actors to tamper with or delete files.

## # Resilience

IPFS has no single point of failure, and users do not need to trust each other. In other words, the failure of a single or even multiple nodes in the network does not affect the functioning of the entire network, and an IPFS node can fetch data from the network as long as at least one other node in the network has that data, regardless of its location.

## # Centralization

IPFS is an open, distributed and participatory network that reduces data silos from centralized servers, making IPFS more resilient than traditional systems. No single entity or person controls, manages or owns IPFS; rather, it is a community-maintained project with multiple implementations of the protocol, multiple tools and apps leveraging that protocol, and multiple users and organizations contributing to its design and development.

## # Performance

IPFS provides faster access to data by enabling it to be replicated to and retrieved from multiple locations, and allowing users to access data from the nearest location using content addressing instead of location-based addressing. In other words, because data can be addressed based on its

contents, a node on the network can fetch that data from *any* other node in the network that has the data; thus, performance issues like latency are reduced.

## # Link rot

IPFS eliminates the problem of link rot by allowing data to be addressed by its content, rather than by its location. So, in other words, content in IPFS is still reachable regardless of its location, and does not depend on specific servers being available.

## # Data sovereignty

IPFS protects data sovereignty by enabling users to store and access data directly on a decentralized network of nodes, rather than centralized, third-party servers. This eliminates the need for intermediaries to control and manage data, giving users full control and ownership over their data.

## # Off-chain storage

IPFS enables verifiable off-chain storage by creating a link between blockchain state and content-addressed published to IPFS. This works by storing a Content IDentifier (CID) (explained in [How IPFS works](#)) in a smart contract.

## # Local-first software

IPFS benefits local-first software by providing a performant, decentralized, peer-to-peer data addressing, routing, and transfer protocol that prioritizes data storage and processing on individual devices. With IPFS, data can be stored, verified and processed locally, and then synchronized and shared with other IPFS nodes when a network connection is available.

## # Vendor lock-in

IPFS prevents vendor lock-in, as users have sovereignty over their data and infrastructure. This is enabled by content-addressing, which decouples the data from a single location or infrastructure provider. Unlike traditional cloud vendors, IPFS enables you to change data storage locations without changing things like APIs and data management. In addition, because IPFS is open-source, community-maintained and modular, users are not obligated to use a particular subsystem (described in [How IPFS works](#)). Instead, users can customize IPFS for their preferred technologies, needs and values.

## # Further reading

- New to IPFS and wanting to dive deeper into the different subsystems that IPFS is composed of and how each one works? See part 3, [How IPFS Works](#), of the [introduction to the basic concepts of IPFS](#)
- Are you looking for a list of implementations of the IPFS protocol, as well as more information on each implementation? See the [IPFS Implementations page](#).
- Are you interested in comparing IPFS to other similar technologies? See the [IPFS Comparisons page](#).