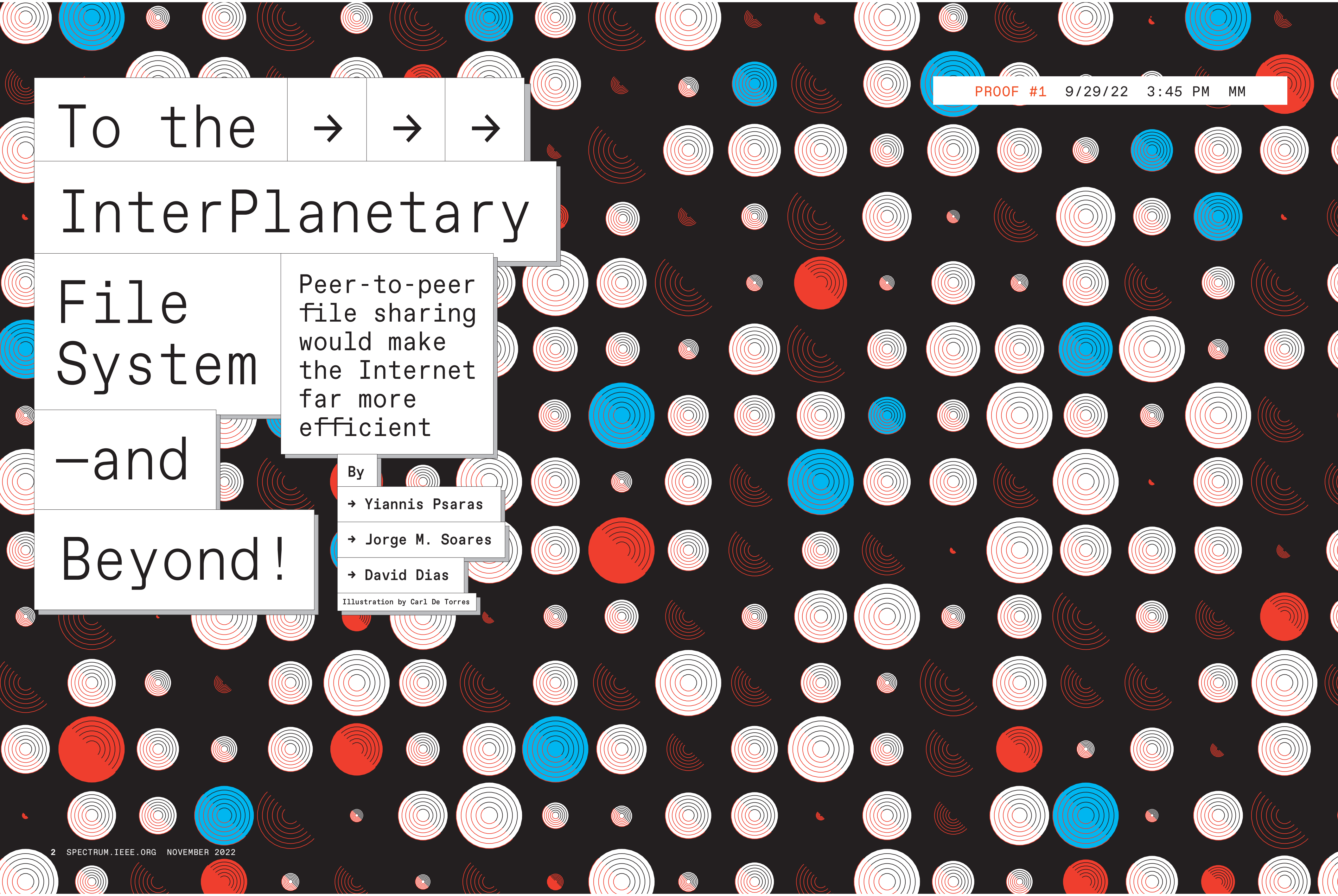


PROOF #1 9/29/22 3:45 PM MM

To the → → → InterPlanetary File System —and Beyond!

Peer-to-peer file sharing would make the Internet far more efficient

By
→ Yiannis Psaras
→ Jorge M. Soares
→ David Dias

Illustration by Carl De Torres

When the COVID-19 pandemic erupted in early 2020, the world made an unprecedented shift to remote work. As a precaution, some Internet providers scaled back service levels temporarily, although that probably wasn't necessary for countries in Asia, Europe, and North America, which were generally able to cope with the surge in demand caused by people teleworking (and binge-watching Netflix). That's because most of their networks were overprovisioned, with more capacity than they usually need. But in countries without the same level of investment in network infrastructure, the picture was less rosy: Internet service providers (ISPs) in South Africa and Venezuela for instance, reported significant strain.

But is overprovisioning the only way to ensure resilience? We don't think so. To understand the alternative approach we're championing, though, you first need to recall how the Internet works.

The core protocol of the Internet, aptly named the Internet Protocol (IP), defines an addressing scheme that computers use to communicate with one another. This scheme assigns addresses to specific devices—people's computers as well as servers—and uses those addresses to send data between them as needed.

It's a model that works well for sending unique information from one point to another, say, your bank statement or a letter from a loved one. This approach made sense when the Internet was used mainly to deliver different content to different people. But this design is not well suited for the mass consumption of static content, such as movies or TV shows.

The reality today is that the Internet is more often used to send exactly the same thing to many people, and it's doing a huge amount of that now, much of which is in the form of video. The demands grow even higher as our screens obtain ever-increasing resolutions, with 4K video already in widespread use and 8K on the horizon.

The content delivery networks (CDNs) used by streaming services such as Netflix help address the problem by temporarily storing content close to, or even inside, many ISPs, but it relies on ISPs and CDNs being able to make deals and deploy the required infrastructure. And it can still leave the edges of the network having to handle more traffic than actually needs to flow.

The real problem is not so much the volume of content being passed around—it's how it is being delivered, from a central source to many different far-away users, even when those users are located right next to one another.

Instead of asking a particular provider, "Please send me this file," your machine asks the network, "Who can send me this file?"

A more efficient distribution scheme in that case would be for the data to be served to your device from your neighbor's device in a direct peer-to-peer manner. But how would your device even know whom to ask? Welcome to the InterPlanetary File System (IPFS).

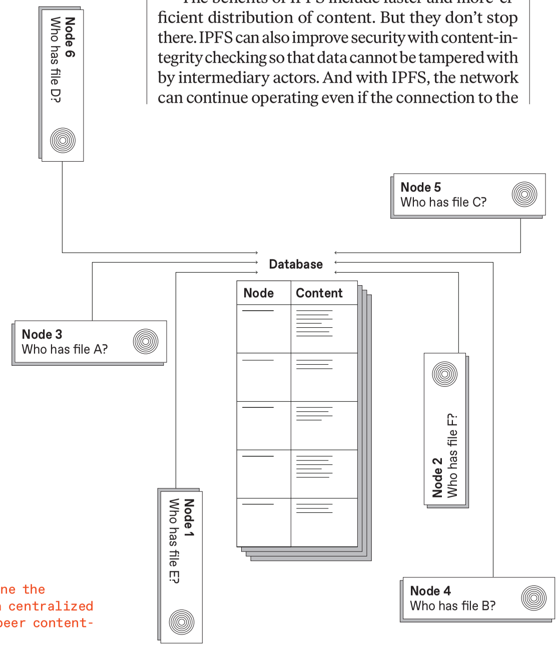
The InterPlanetary File System gets its name because, in theory, it could be extended to share data even between computers on different planets of the solar system. For now, though, we're focused on rolling it out for just Earth!

The key to IPFS is what's called content addressing. Instead of asking a particular provider, "Please send me this file," your machine asks the network, "Who can send me this file?" It starts by querying peers: other computers in the user's vicinity, others in the same house or office, others in the same neighborhood, others in the same city—expanding progressively outward to globally distant locations, if need be, until the system finds a copy of what you're looking for.

These queries are made using IPFS, an alternative to the Hypertext Transfer Protocol (HTTP), which powers the World Wide Web. Building on the principles of peer-to-peer networking and content-based addressing, IPFS allows for a decentralized and distributed network for data storage and delivery.

The benefits of IPFS include faster and more-efficient distribution of content. But they don't stop there. IPFS can also improve security with content-integrity checking so that data cannot be tampered with by intermediary actors. And with IPFS, the network can continue operating even if the connection to the originating server is cut or if the service that initially provided the content is experiencing an outage—particularly important in places with networks that work only intermittently. The IPFS also offers resistance to censorship.

To understand more fully how IPFS differs from most of what takes place online today, let's take a quick look at the Internet's architecture and some earlier peer-to-peer approaches.



One scheme used by peer-to-peer systems to determine the location of a file is to keep that information in a centralized database. Napster, the first large-scale peer-to-peer content-delivery system used this approach.

As mentioned above, with today's Internet architecture, you request content based on a server's address. This comes from the protocol that underlies the Internet and governs how data flows from point to point, a scheme first described by Vint Cerf and Bob Kahn in a 1974 paper in the IEEE Transactions on Communications and now known as the Internet Protocol. The World Wide Web is built on top of the Internet Protocol. Browsing the Web consists of asking a specific machine, identified by an IP address, for a given piece of data.

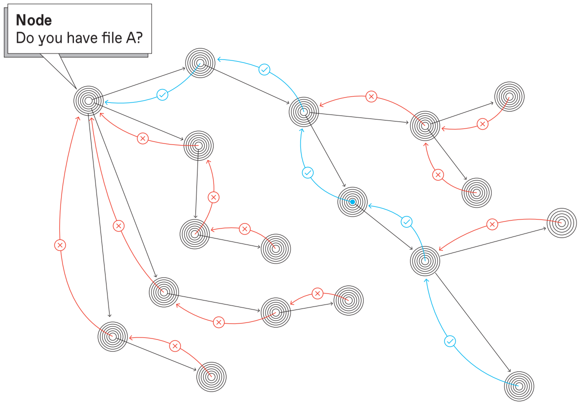
The process starts when a user types a URL into the address bar of the browser, which takes the hostname portion and sends it to a Domain Name System (DNS) server. That DNS server returns a corresponding numerical IP address. The user's browser will then connect to the IP address and ask for the Web page located at that URL.

In other words, even if a computer in the same building has a copy of the desired data, it will neither see the request, nor would it be able to match it to the copy it holds because the content does not have an intrinsic identifier—it is not content-addressed.

A content-addressing model for the Internet would give data, not devices, the leading role. Requesters would ask for the content explicitly, using a unique identifier (akin to the DOI number of a journal article or the ISBN of a book), and the Internet would handle forwarding the request to an available peer that has a copy.

The major challenge in doing so is that it would require changes to the core Internet infrastructure, which is owned and operated by thousands of ISPs worldwide, with no central authority able to control what they all do. While this distributed architecture is one of the Internet's greatest strengths, it makes it nearly impossible to make fundamental changes to the system, which would then break things for many of the people using it. It's often very hard even to implement incremental improvements. A good example of the difficulty encountered when introducing change is IPv6, which expands the number of possible IP addresses. Today, almost 25 years after its introduction, it still hasn't reached 50 percent adoption.

A way around this inertia is to implement changes at a higher layer of abstraction, on top of existing Internet protocols, requiring no modification to the underlying networking software stacks or intermediate devices.



Another approach to finding a file in a peer-to-peer network is called query flooding. The node seeking a file broadcasts a request for it to all nodes to which it is attached. If the node receiving the request does not have the file [red], it forwards the request to all the nodes to which it is attached until finally a node with the file passes a copy back to the requester [blue]. The Gnutella peer-to-peer network used this protocol.

Other peer-to-peer systems besides IPFS, such as BitTorrent and Freenet have tried to do this by introducing systems that can operate in parallel with the World Wide Web, albeit often with Web interfaces. For example, you can click on a Web link for the BitTorrent tracker associated with a file, but this process typically requires that the tracker data be passed off to a separate application from your Web browser to handle the transfers. And if you can't find a tracker link, you can't find the data.

Freenet also stores content in a distributed peer-to-peer manner that can be requested via an identifier and can even be accessed using the Web's HTTP protocol. But Freenet and IPFS have different aims: Freenet has a strong focus on anonymity and manages the replication of data in ways that serve that goal but lessen performance and user control. IPFS provides flexible, high-performance sharing and retrieval mechanisms but keeps control over data in the hands of the users.

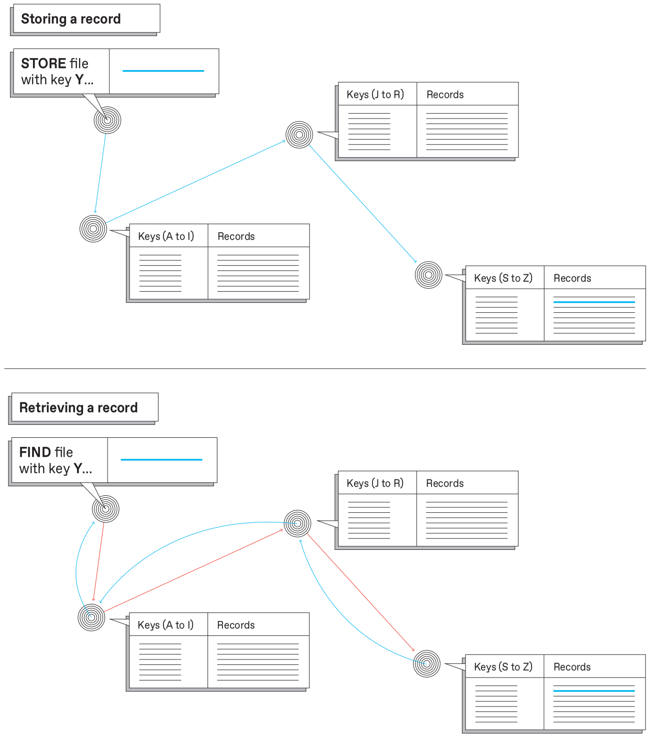
We designed IPFS as a protocol to upgrade the Web and not to create an alternative version. It is designed to make the Web better, to allow people to work offline, to make links permanent, to be faster and more secure, and to make it as easy as possible to use.

IPFS started in 2013 as an open-source project supported by Protocol Labs, where we work, and built by a vibrant community and ecosystem with dozens of organizations and hundreds of developers. IPFS is built on a strong foundation of previous work in peer-to-peer (P2P) networking and content-based addressing.

The core tenet of all P2P systems is that users simultaneously participate as clients (which request and receive files from others) and as servers (which store and send files to others). The combination of content addressing and P2P provides the right ingre-

Illustrations by Carl De Torres

To keep track of which nodes hold which files, the InterPlanetary File System uses what's called a distributed hash table. In this simplified view, three nodes hold different parts of a table that has two columns: One column (Keys) contains hashes of the stored files; the other column (Records) contains the files themselves. Depending on what its hashed key is, a file gets stored in the appropriate place [left]—depicted here as though the system checked the first letter of hashes and stored different parts of the alphabet in different places. The actual algorithm for distributing files is more complex, but the concept is similar. Retrieving a file is relatively efficient because it's possible to locate the file according to what its hash is [right].



dients for fetching data from the closest peer that holds a copy of what's desired—or more correctly, the closest one in terms of network topology, though not necessarily in physical distance.

To make this happen, IPFS produces a fingerprint of the content it holds (called a hash) that no other item can have. That hash can be thought of as a unique address for that piece of content. Changing a single bit in that content will yield an entirely different address. Computers wanting to fetch this piece of content broadcast a request for a file with this particular hash.

The fact that identifiers are unique and do not change often has people referring to IPFS as the "Permanent Web." Because identifiers never change, the network will be able to find a specific file as long as some computer on the network stores it.

Name persistence and immutability inherently provide another significant property: verifiability. Having the content and its identifier, a user can verify that what was received is what was asked for and has not been tampered with, either in transit or by the provider. This not only improves security but also helps safeguard the public record and prevent history from being rewritten.

You might wonder what would happen with content that needs to be updated to include fresh information, such as a Web page. This is a valid concern and IPFS does have a suite of mechanisms that would point users to the most up-to-date content.

The world had a chance to observe how content addressing worked in April 2017 when the government of Turkey blocked access to Wikipedia because an article on the platform described Turkey as a state that sponsored terrorism. Within a week, a full copy of the Turkish version of Wikipedia was added to IPFS, and it remained accessible to people in the country for the nearly three years that the ban continued.

A similar demonstration took place half a year later, when the Spanish government tried to suppress an independence referendum in Catalonia, ordering ISPs to block related websites. Once again, the information remained available via IPFS.

IPFS is an open, permissionless network: Any user can join and fetch or provide content. Despite numerous open-source success stories, the current Internet is heavily based on closed platforms, many of which adopt lock-in tactics but also offer users great convenience. While IPFS can provide improved efficiency, privacy, and security, giving this decentralized platform the level of usability that people are accustomed to remains a challenge.

You see, the peer-to-peer, unstructured nature of IPFS is both a strength and a weakness. While CDNs have built sprawling infrastructure and advanced techniques to provide high-quality service, IPFS nodes are operated by end users. The network therefore relies on their behavior—how long their computers are online, how good their connectivity is, and what data they decide to cache. And often those things are not optimal.

One of the key research questions for the folks working at Protocol Labs is how to keep the IPFS network resilient despite shortcomings in the nodes that make it up—or even when those nodes exhibit selfish or malicious behavior. We'll need to overcome such issues if we're to keep the performance of IPFS competitive with conventional distribution channels.

You may have noticed that we haven't yet provided an example of an IPFS address. That's because hash-based addressing results in URLs that aren't easy to spell out or type.

For instance, you can find the Wikipedia logo on IPFS by using the following address in a suitable browser: ipfs://QmRW3V9znzFW9M5FYbitSEvd5dQrPWGvPvgQD6LM22Tv8D/. That long string can be thought of as a digital fingerprint for the file holding that logo.

There are other content-addressing schemes that use human-readable naming, or hierarchical, URL-style naming, but each comes with its own set of trade-offs. Finding practical ways to use human-readable names with IPFS would go a long way toward improving user-friendliness. It's a goal, but we're not there yet.

Protocol Labs, where we work, has been tackling these and other technical, usability, and societal issues for most of the last decade. Over this time, we have been seeing rapidly increasing adoption of IPFS, with its network size doubling year over year. Scaling up at such speeds brings many challenges. But that's par for the course when your intent is changing the Internet as we know it.

Widespread adoption of content addressing and IPFS should help the whole Internet ecosystem. By empowering users to request exact content and verify that they received it unaltered, IPFS will improve trust and security. Reducing the duplication of data moving through the network and procuring it from nearby sources will let ISPs provide faster service at lower cost. Enabling the network to continue providing service even when it becomes partitioned will make our infrastructure more resilient to natural disasters and other large-scale disruptions.

Reducing the duplication of data moving through the network and procuring it from nearby sources will let ISPs provide faster service at lower cost.

But is there a dark side to decentralization? We often hear concerns about how peer-to-peer networks may be used by bad actors to support illegal activity. These concerns are important but sometimes overstated.

One area where IPFS improves on HTTP is in allowing comprehensive auditing of stored data. For example, thanks to its content-addressing functionality and, in particular, to the use of unique and permanent content identifiers, IPFS makes it easier to determine whether certain content is present on the network, and which nodes are storing it. Moreover, IPFS makes it trivial for users to decide what content they distribute and what content they stop distributing (by merely deleting it from their machines).

At the same time, IPFS provides no mechanisms to allow for censorship, given that it operates as a distributed P2P file system with no central authority. So there is no actor with the technical means to prohibit the storage and propagation of a file or to delete a file from other peers' storage. Consequently, censorship of unwanted content cannot be technically enforced, which represents a safeguard for users whose freedom of speech is under threat. Lawful requests to take down content are still possible, but they need to be addressed to the users actually storing it, avoiding commonplace abuses (like illegitimate DMCA takedown requests) against which large platforms have difficulties defending.

Ultimately, IPFS is an open network, governed by community rules, and open to everyone. And you can become a part of it today! The Brave browser ships with built-in IPFS support, as does Opera for Android. There are browser extensions available for Chrome and Firefox, and IPFS Desktop makes it easy to run a local node. Several organizations provide IPFS-based hosting services, while others operate public gateways that allow you to fetch data from IPFS through the browser without any special software.

These gateways act as entries to the P2P network and are important to bootstrap adoption. Through some simple DNS magic, a domain can be configured so that a user's access request will result in the corresponding content being retrieved and served by a gateway, in a way that is completely transparent to the user.

So far, IPFS has been used to build varied applications, including systems for e-commerce, secure distribution of scientific data sets, mirroring Wikipedia, creating new social networks, sharing cancer data, blockchain creation, secure and encrypted personal-file storage and sharing, developer tools, and data analytics.

You may have used this network already: If you've ever visited the Protocol Labs site (Protocol.ai), you've retrieved pages of a website from IPFS without even realizing it! ■

Illustrations by Carl De Torres