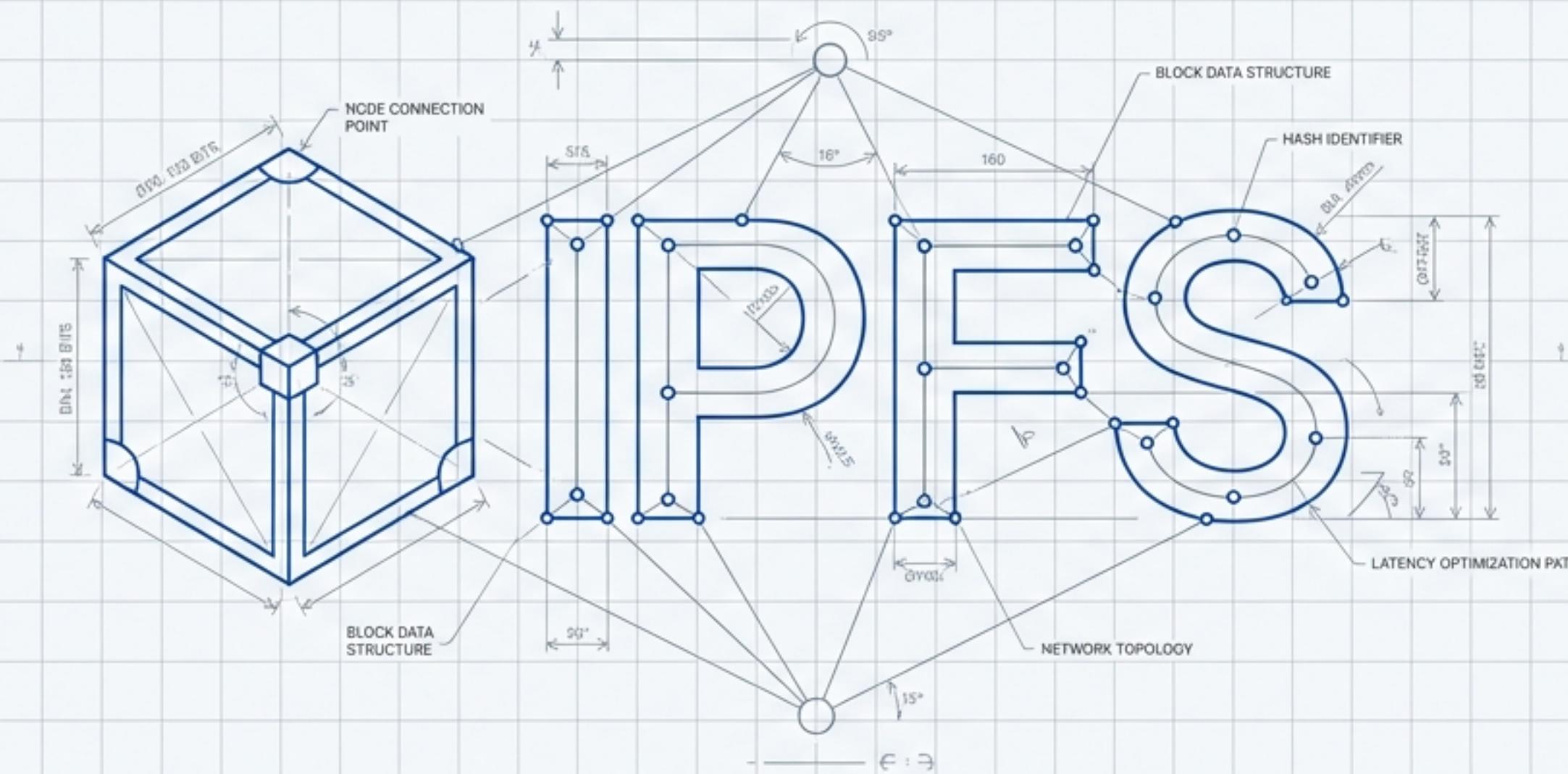


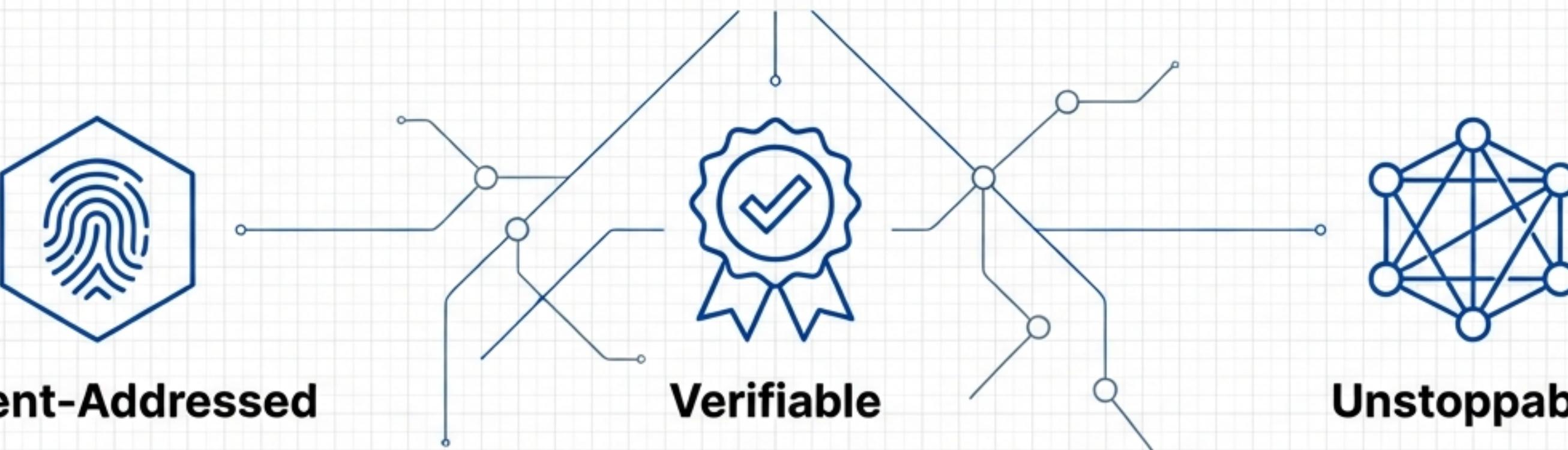
The Web, Reimagined.



An architectural guide to the InterPlanetary File System (IPFS).

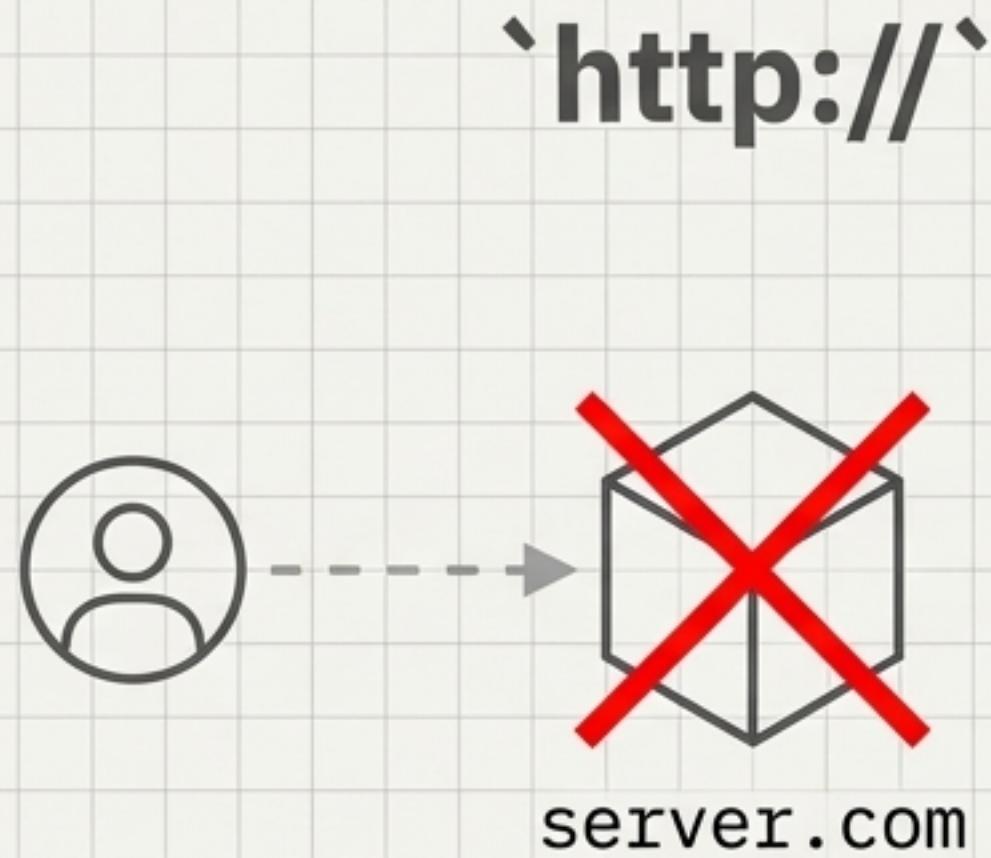
The Building Blocks for a Smarter Web

IPFS provides open protocols to make your data smarter:
content-addressed, verifiable, and unstoppable.



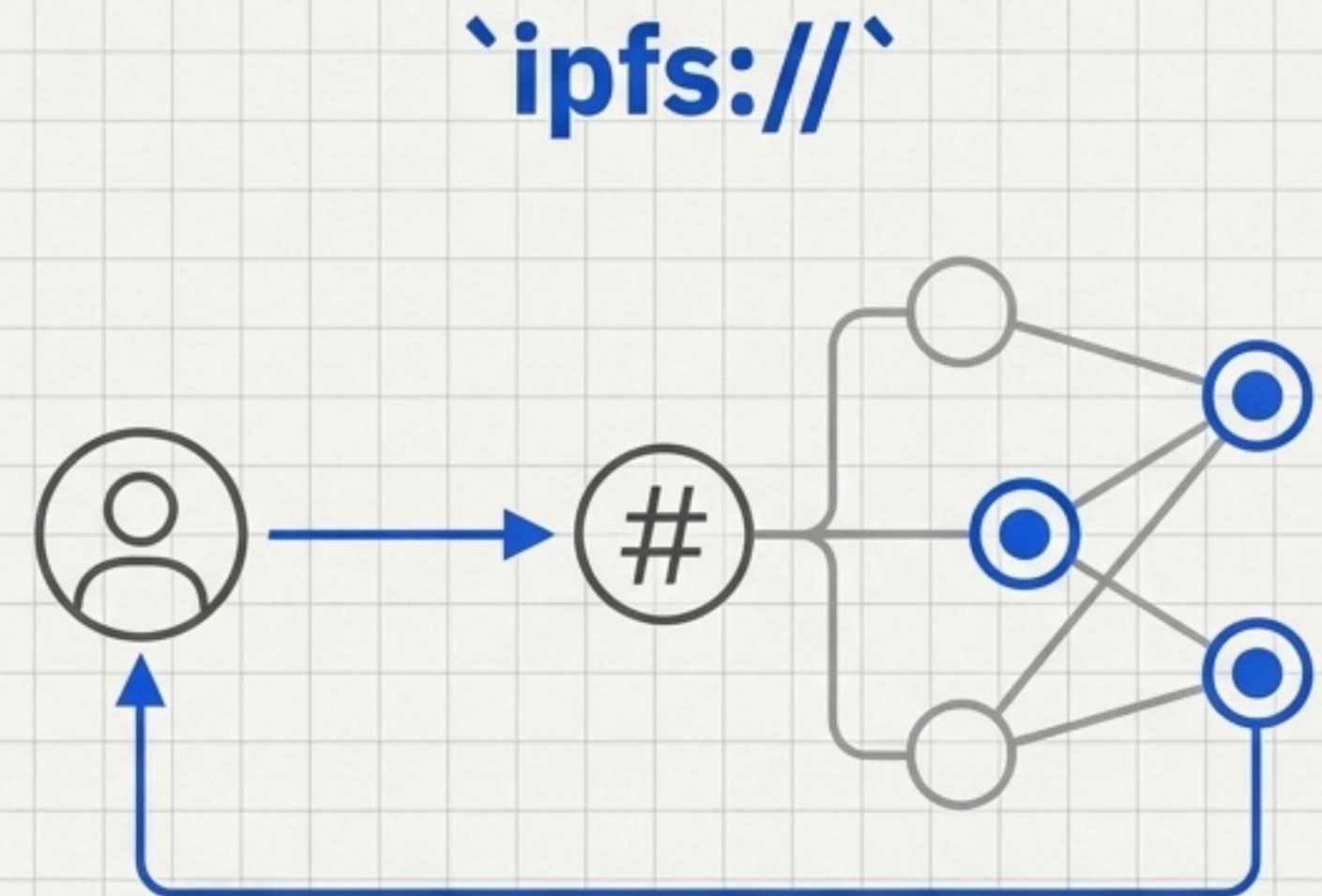
On a technical level, IPFS is a set of open protocols for addressing, routing, and transferring data on the web, built on the foundational ideas of content addressing and peer-to-peer networking.

The Fundamental Shift: From *Where* to *What*



`http://example.com/images/cat.jpg`

Relies on a single, fragile location.



`ipfs://bafybeig...q5ztc5a`

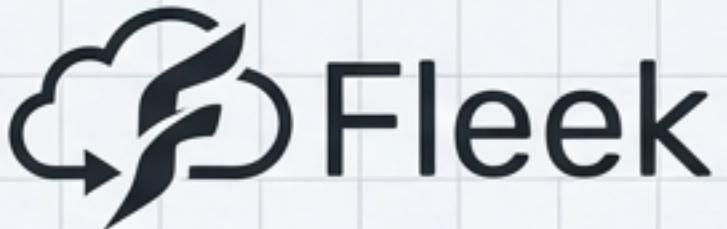
*Relies on the content's unique hash.
Verifiable and resilient.*

Powering the Decentralized Ecosystem

Many popular and innovative projects are built with IPFS as a foundational layer.



Morpheus.Network



Snapshot



Arbol

Explore hundreds more in the ecosystem directory and the awesome-ipfs list.

Choose Your Journey

How do you want to engage with IPFS? Select a path to get started with the right tools and guides for your goal.



Retrieve Data

Fetch files and data from the IPFS network.



Provide Data

Publish your own content and make it available to others.



Deploy a Site

Host a static website or decentralized application (dApp).



Build an App

Integrate IPFS into your application natively or over HTTP.

Journey 1: Retrieve Data from the Network

Quickly access data on IPFS, no programming required.

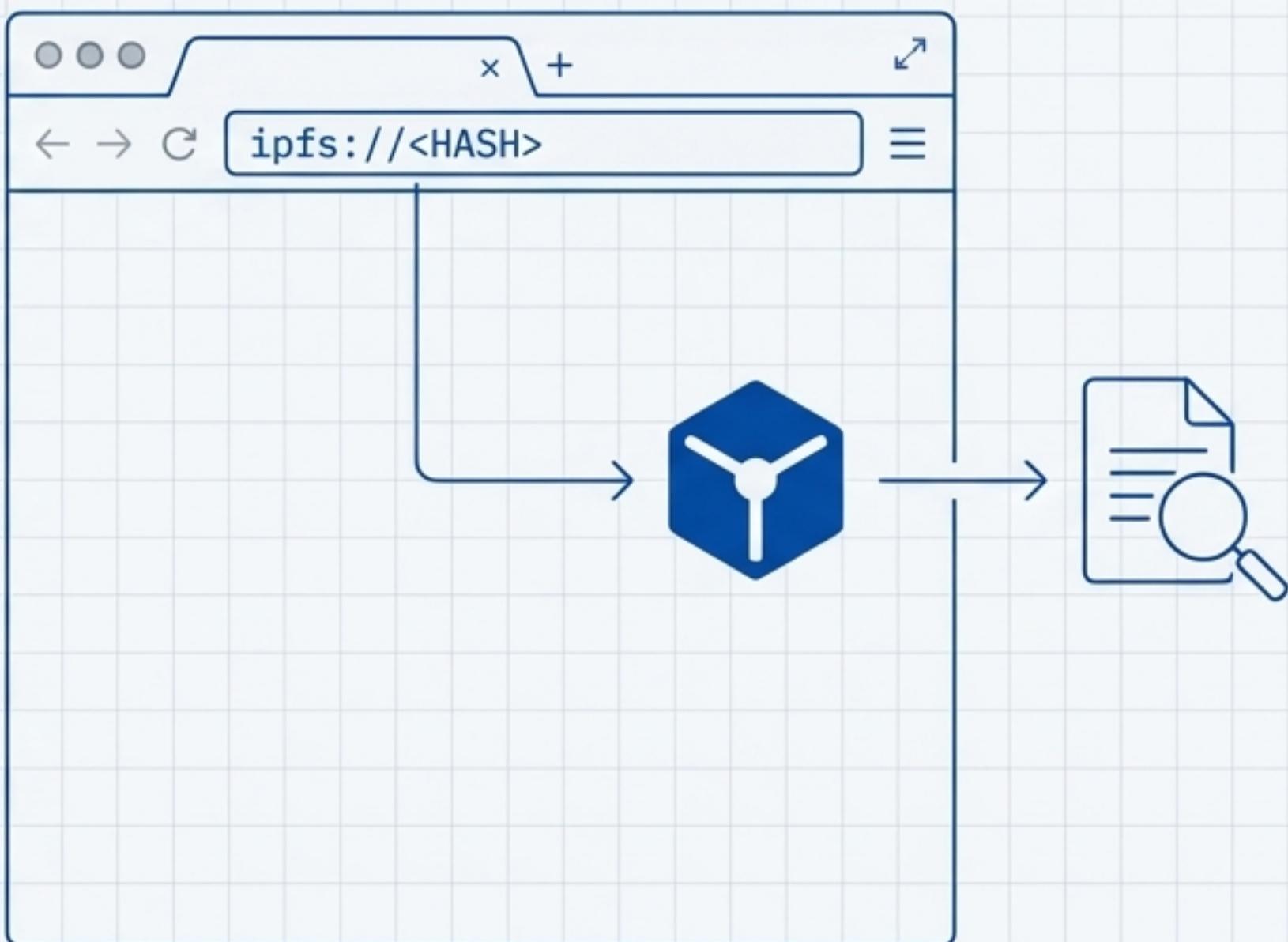
Method A: Use an IPFS Gateway

The simplest way to start. Fetch any data using its Content Identifier (CID) through a public HTTP gateway.

https://ipfs.io/ipfs/<YOUR_CID>

Method B: Install IPFS Companion

Upgrade your browser to resolve `ipfs://` and `ipns://` addresses directly. This browser extension redirects requests to your local node or a gateway, bringing native IPFS support to your web experience.



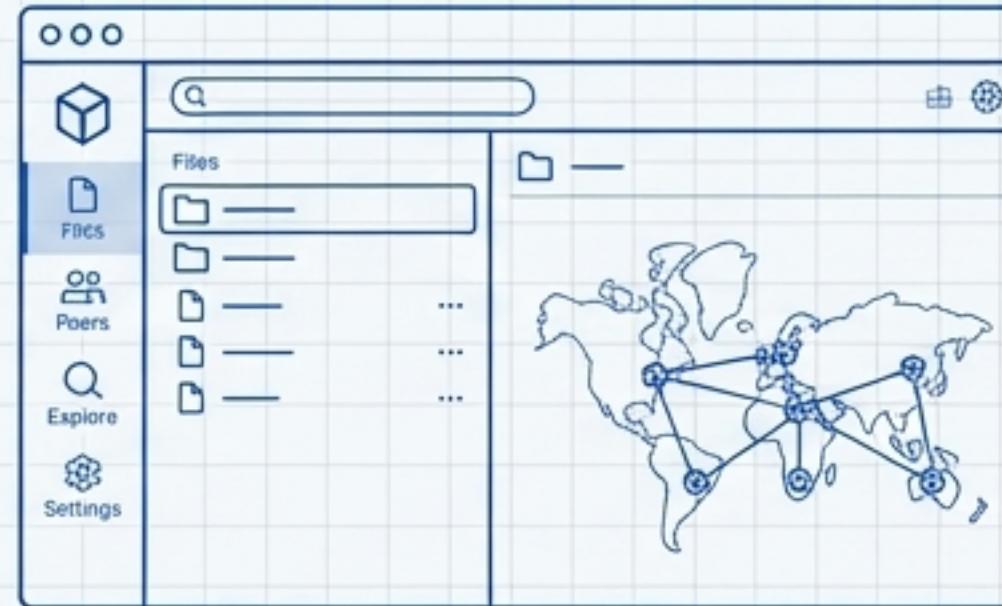
Journey 2: Provide Data to the Network

Publish your content and ensure it remains available.

Method A: IPFS Desktop

The all-in-one solution. Install IPFS Desktop to run a full IPFS node (Kubo) with a user-friendly interface. Manage your files, monitor peers, and explore content directly from your machine.

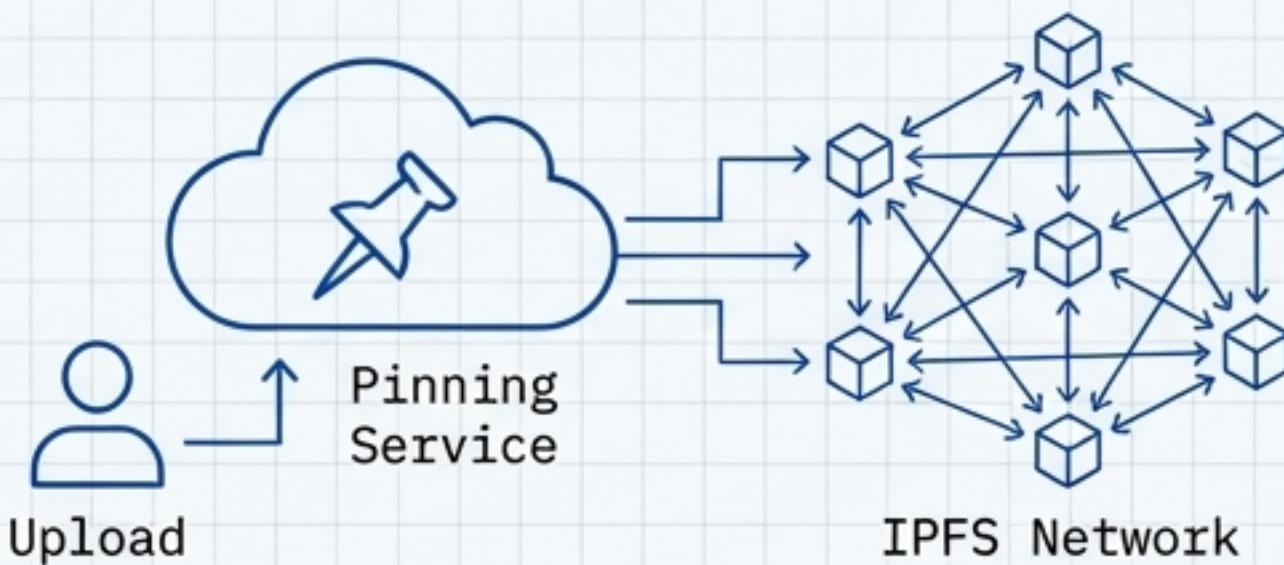
Publish content to the network with IPFS Desktop.



 IPFS Desktop |  Kubo

Method B: Use a Pinning Service

For data persistence without running your own node. Pinning services ensure your data is always available on the network by hosting it on their IPFS nodes.

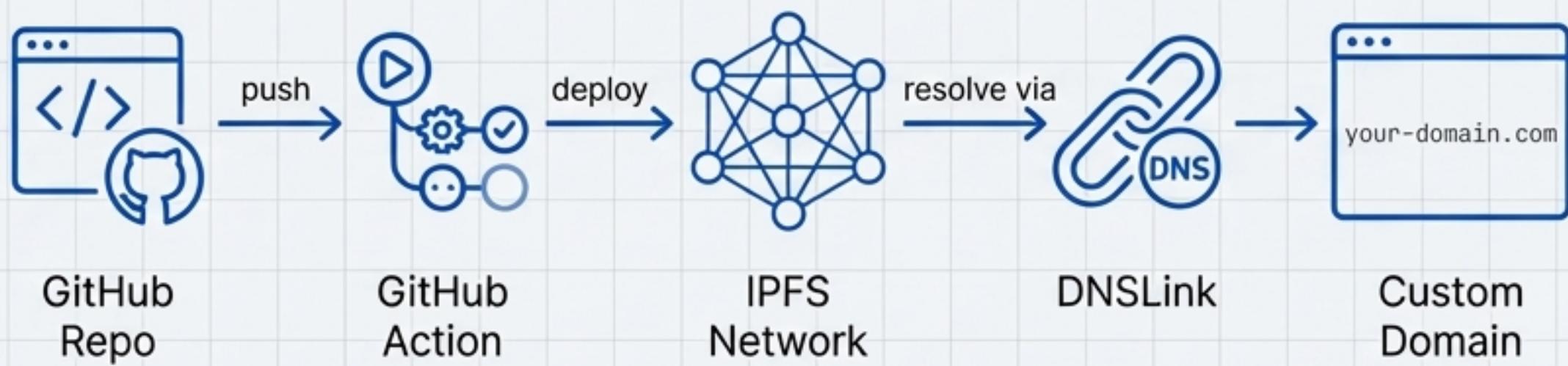


Journey 3: Deploy Your Static Site or dApp

Leverage IPFS for censorship-resistant, resilient web hosting.

Step 1: Deploy with Automation

Use a GitHub Action to automatically build and deploy your static site to the IPFS network on every push.



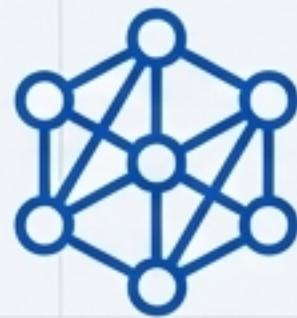
Step 2: Use a Custom Domain

Set up a DNSLink gateway to serve your site from a human-readable domain name (e.g., `your-domain.com`) instead of a long hash.

Extra Tip: The docs provide guides for configuring popular static site generators (like Hugo, Jekyll, etc.) for publishing to IPFS.

Journey 4: Build Your Application

Integrate IPFS at the core of your app. You have two primary approaches.



****Build Natively****

Use IPFS implementations directly in your application's language for maximum control and peer-to-peer capabilities.

P2P

In-browser

Backend Services



****Build via HTTP****

Interact with a remote IPFS node using familiar HTTP APIs for interoperability and simplicity.

Agnostic

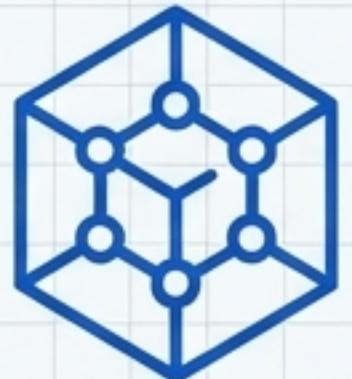
Interoperable

Standard APIs

Build Path A: Natively with IPFS Implementations

For IPFS-native apps with direct network interaction

For JavaScript Developers

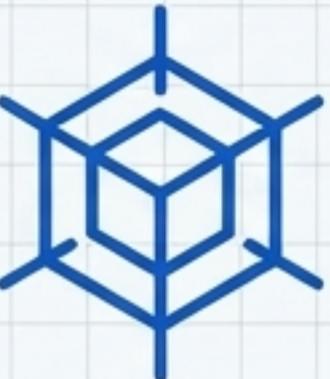


Helia

Tool*: Helia

The premier JS implementation for building IPFS-native apps in the browser or Node.js. Check the “IPFS in web apps” guide for detailed tutorials.

For Go Developers / CLI Users



Kubo

Tool*: Kubo

The original and most feature-rich implementation of IPFS. Ideal for backend services or interacting with IPFS from the command line.

For Other Languages

Many other implementations exist, written in different languages and tailored to specific needs and use cases.

Build Path B: Via Standard HTTP

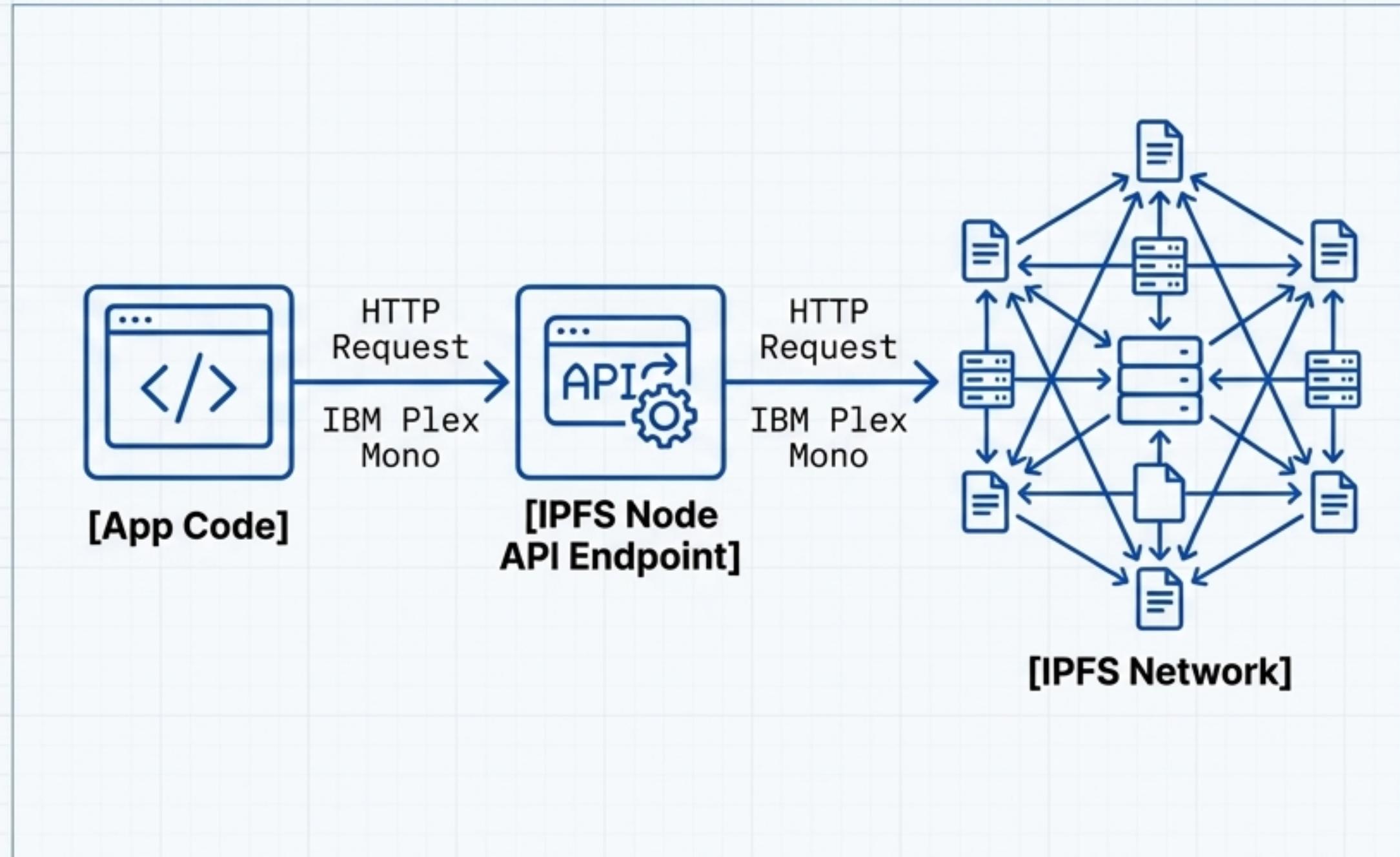
For interoperability across all IPFS implementations.

For Control: Kubo RPC API

Control a remote Kubo node programmatically using its comprehensive HTTP API. Multiple clients are available in various languages, allowing you to manage your node from any application stack.

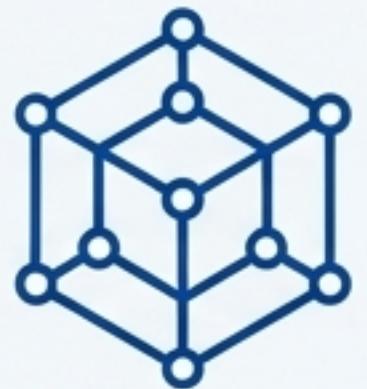
For Retrieval: IPFS Gateways

For a simple, runtime-agnostic HTTP interface for retrieving data. Gateways provide a universal way to fetch content from the IPFS network without needing a specific client library.



Your Core IPFS Toolkit

A summary of the primary implementations and tools for getting started.



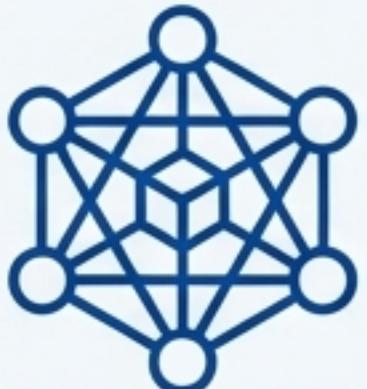
Kubo

The reference implementation in Go. Powerful for backends and the CLI.



IPFS Desktop

The all-in-one desktop app bundling Kubo with a user-friendly UI.



Helia

The modern JavaScript implementation for building IPFS-native web apps.



IPFS Companion

The browser extension that brings native `ipfs://` support to your browser.

If you run into issues, the official documentation includes a comprehensive troubleshooting guide.

The Blueprint: Go Deeper

Understand the theory, architecture, and history behind IPFS.

Recommended Reading from the Docs:

The Glossary

Look up key terms and definitions.

Basic Concepts

Learn what IPFS is and isn't, and the problems it solves.

Ideas and Theory

Dive into core concepts like hashing, immutability, and persistence.

Subsystems and Components

Explore the architecture that makes IPFS work.

Case Studies

See how other complex systems leverage IPFS.

The Project Section

Understand the project's history and ecosystem status.

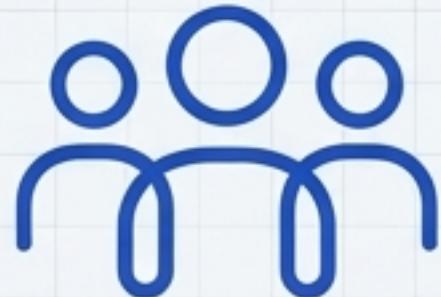
Join the Builders

IPFS has a bustling community of designers, developers, writers, and activists. Here's how to get involved.



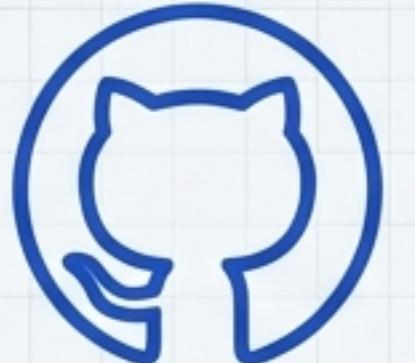
For Technical Support

Are you developing with IPFS and need expert help? See the guide to getting technical help and support for the fastest assistance.



For Community Engagement

Explore events, resources, and contribution opportunities in the Community section of the docs.



To Improve the Docs

Suggest new content or give feedback by opening an issue or editing the page on GitHub.