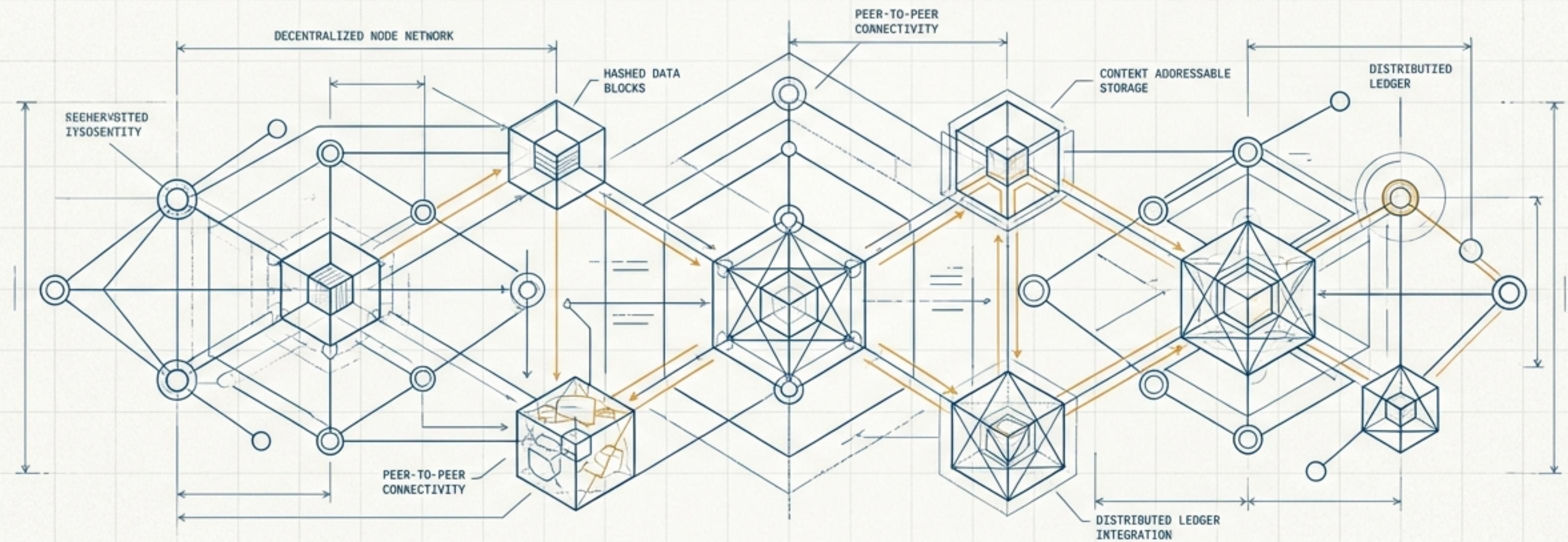


Building a More Resilient Web with IPFS

A Developer's Journey from Concept to Creation



IPFS is a Set of Building Blocks for a Better Web.

It uses open protocols for addressing, routing, and transferring data based on what it is, not where it is. The fundamental goal is to make data:



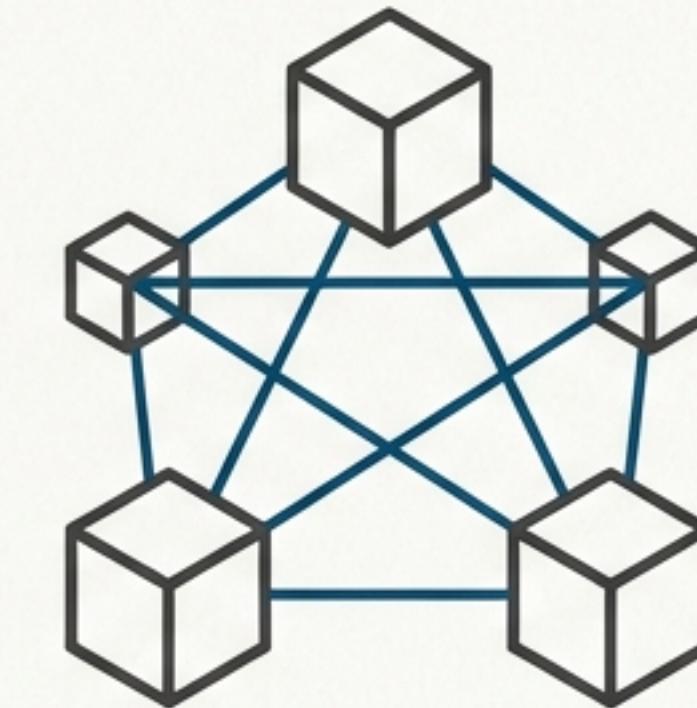
Content-Addressed

Retrieve data based on its unique hash (fingerprint), not its location.



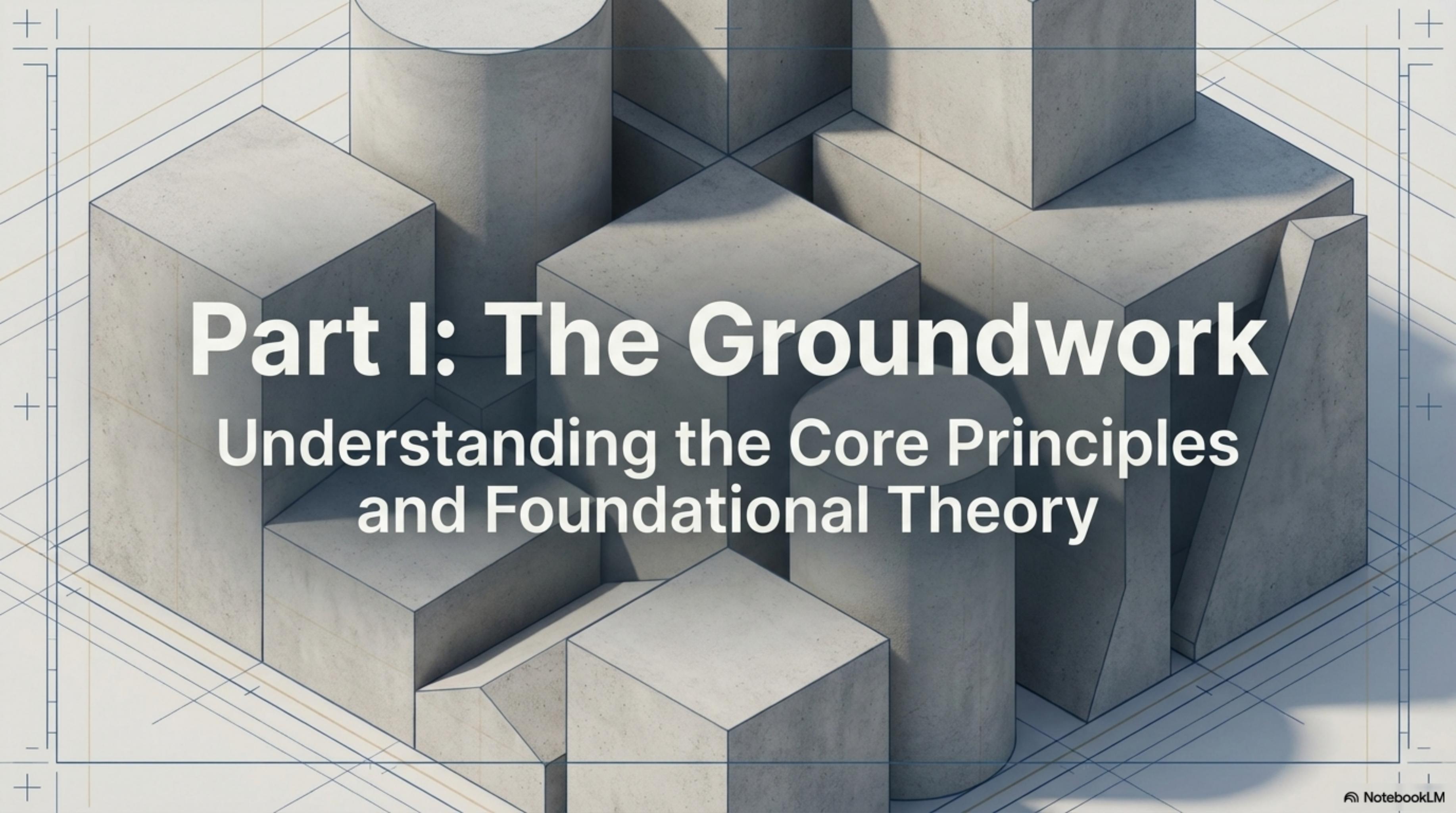
Verifiable

Easily confirm that you've received the exact data you requested, without trusting the provider.



Unstoppable

Data is served by a distributed network of peers, eliminating single points of failure.

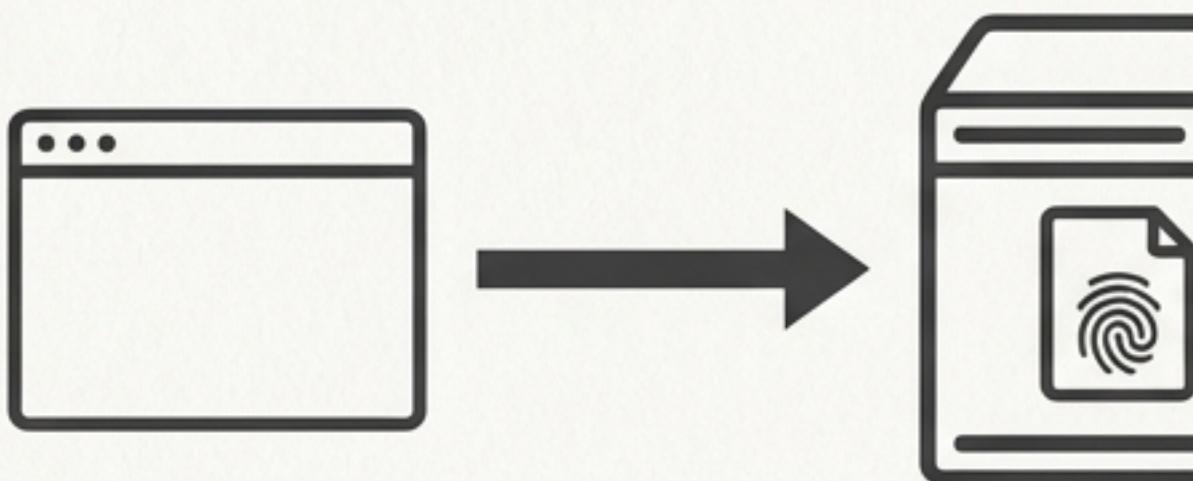


Part I: The Groundwork

Understanding the Core Principles and Foundational Theory

The Core Shift: From Location to Content Addressing

The Old Way: Location-Addressed



- Points to a *location*.
- Content can change or disappear.
- Server-dependent.

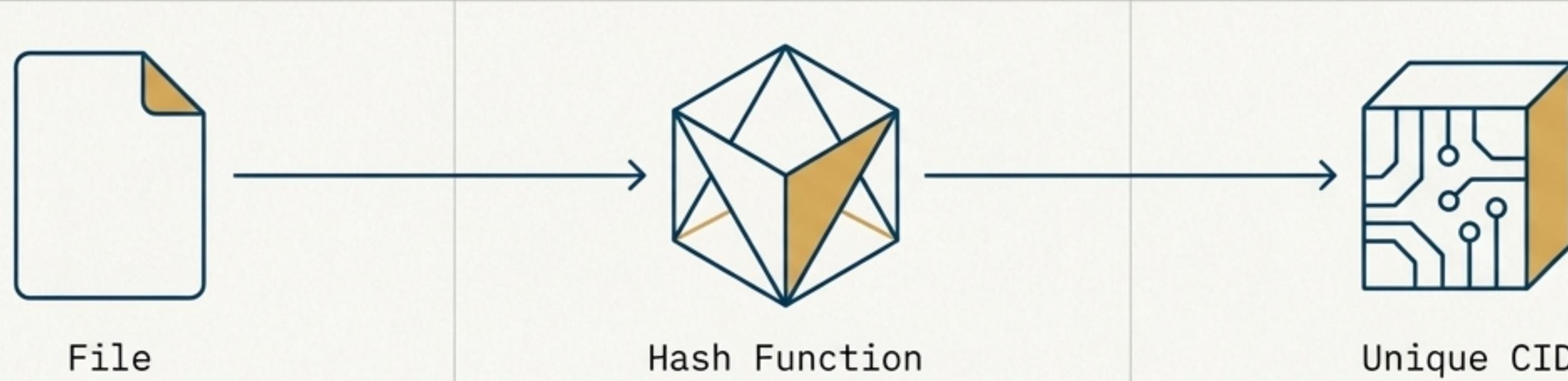
The IPFS Way: Content-Addressed



- Points to the *content* itself via a unique **Content Identifier (CID)**.
- The CID is a hash of the content, meaning the content is immutable.
- Peer-to-peer and verifiable.

The Theory Behind the Technology

Three key theoretical ideas provide the foundation for IPFS:

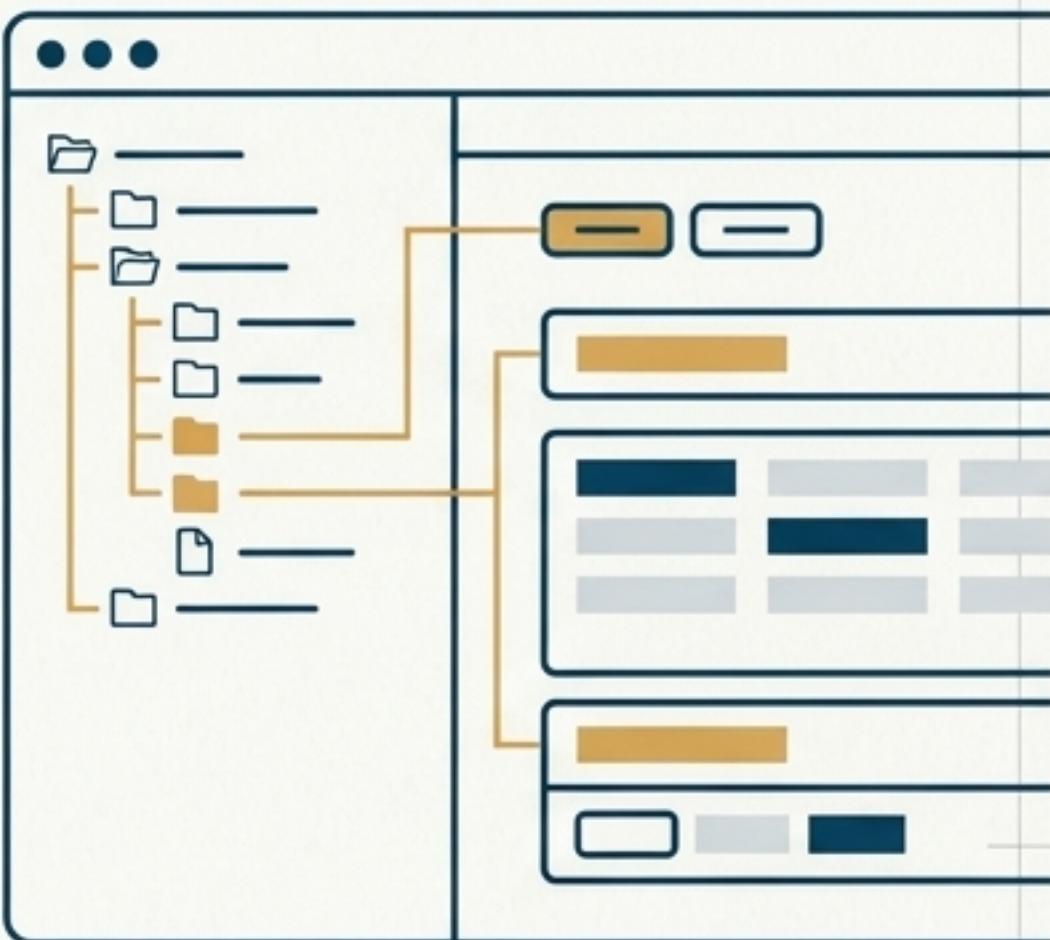
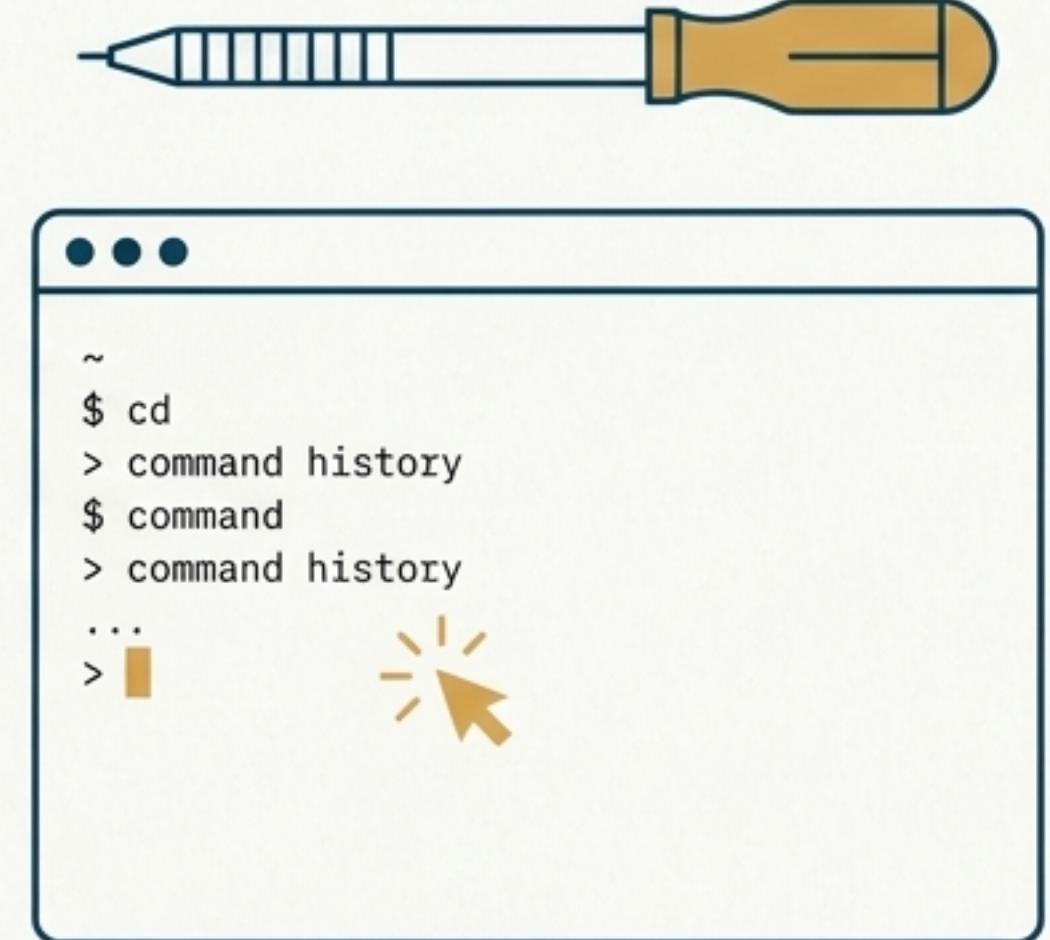


- 1.**
Hashing: Every piece of content is given a unique, fixed-length fingerprint (a hash). This is the basis for the Content Identifier (CID). Any change to the file results in a completely new hash.
- 2.**
Immutability: Because data is addressed by its hash, it cannot be changed without also changing its address. This guarantees data integrity.
- 3.**
Persistence: Data remains on the network as long as at least one node decides to keep it (a process called 'pinning'). This creates a resilient, permanent web.



```
code.js X

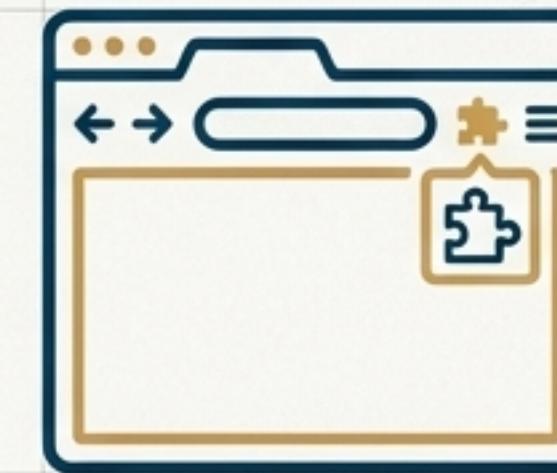
1 import "code";
2 import { from "Ipfs";
3 import { upp } from "content";
4
5 function doin() {
6   const app = "https://devmarp/uthor.js";
7
8   const name = Feeapple()
9   return {
10     images: "local",
11     seating: "https://detoarom.org/"
12   }
13 };
14 return ausr;
15 };
```



Part II: The Workshop

Getting Hands-On with IPFS Clients and Data

Your First Step: Choose Your IPFS Client



IPFS Desktop

A graphical user interface (GUI) with a bundled IPFS node (Kubo). Manage files, peers, and explore content visually.

Best for: Visual interaction and easy file management.

Kubo

A command-line interface (CLI) implementation written in Go. The power-user tool for scripting and server-side interaction.

Best for: Automation, back-end development, and direct control.

IPFS Companion

A browser extension that simplifies access to IPFS resources and lets your browser understand `ipfs://` and `ipns://` addresses natively.

Best for: Everyday browsing and interacting with the distributed web.

How to Retrieve Data from the Network

Primary Method: Public Gateways

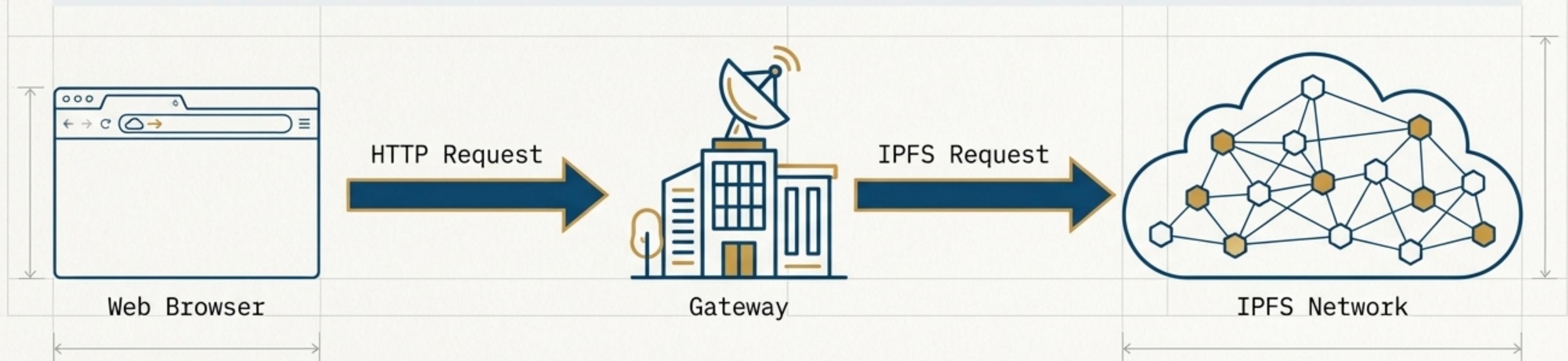
Gateways act as a bridge between the traditional web and IPFS. Simply use a gateway URL with a file's CID.

```
# Accessing a file via a public gateway:  
https://ipfs.io/ipfs/bafybeig.../image.png
```

Alternative: IPFS Companion

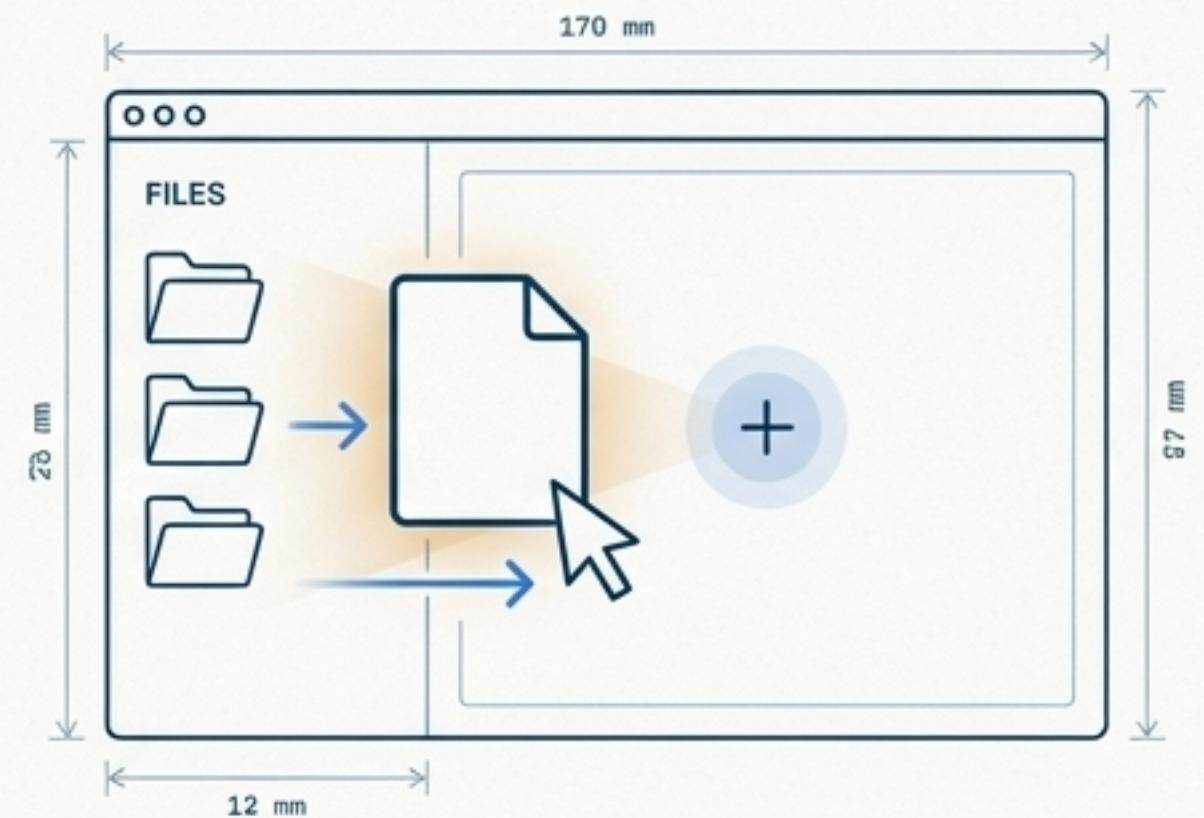
Install the browser extension to resolve native IPFS addresses directly.

```
ipfs://bafybeig.../image.png
```



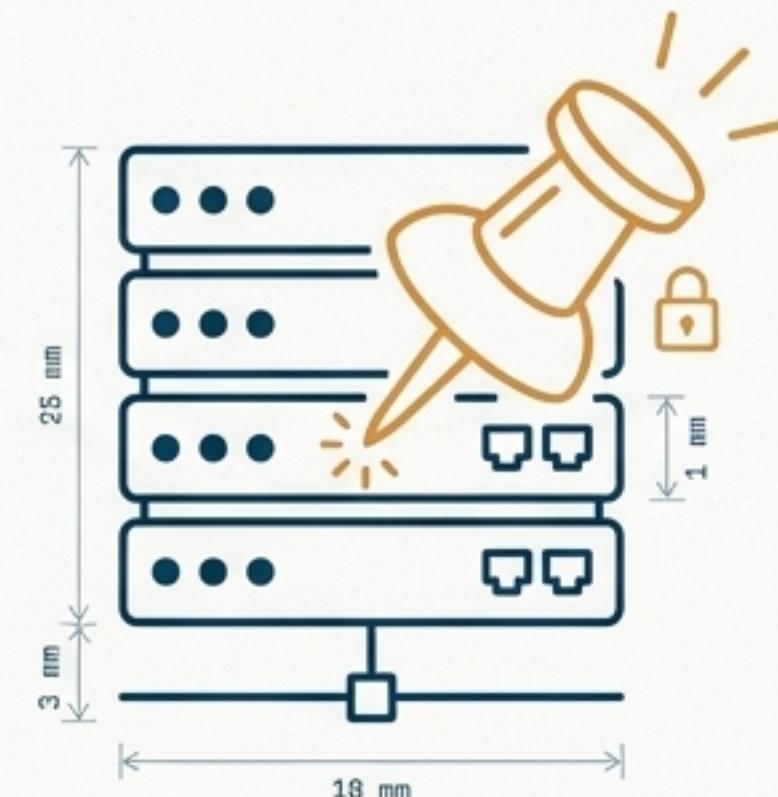
How to Provide Data to the Network

Method 1: Using IPFS Desktop



The user interface provides simple drag-and-drop or file import functionality to add your content to your local IPFS node. Your node then announces the content to the network, making it available to others.

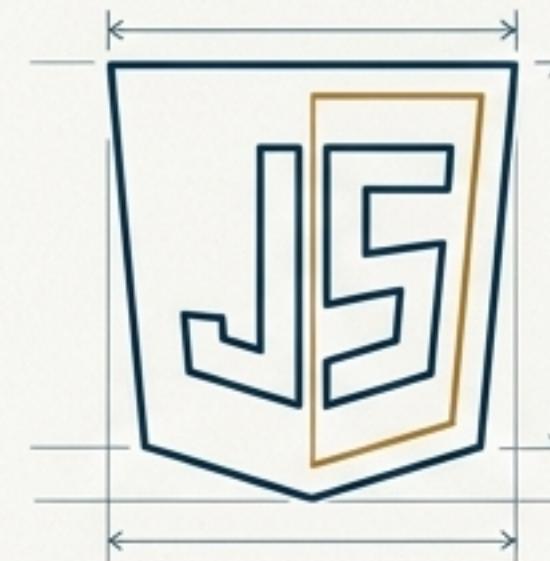
Method 2: Using a Pinning Service



For data persistence without running your own node 24/7, use a pinning service. These services run IPFS nodes on your behalf and 'pin' your content to ensure it remains available on the network.

Three Paths to Building with IPFS

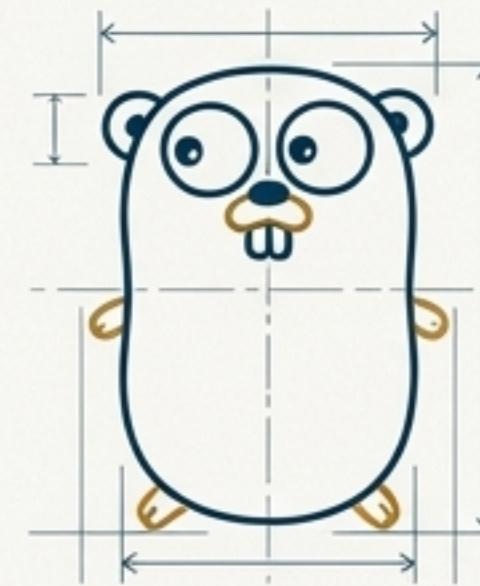
Path 1: IPFS-Native Web Apps (JavaScript)



Build applications that run directly in the browser using JavaScript libraries.

Key Tool: [Helia](#), the modern JS implementation of IPFS.

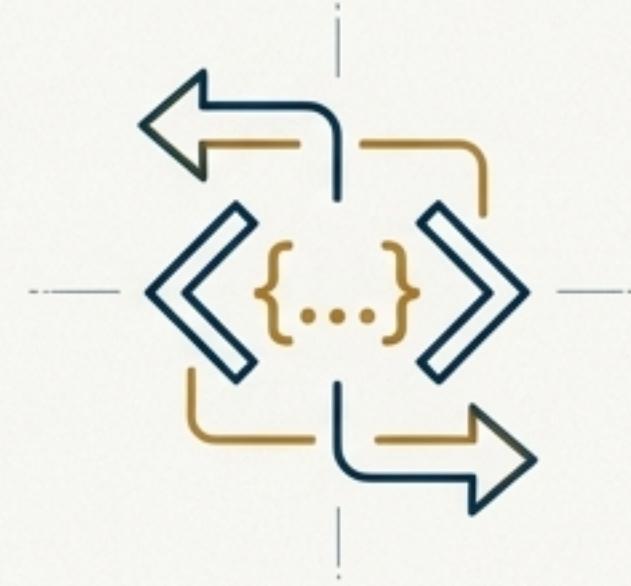
Path 2: Server-Side & CLI Apps (Go)



For back-end services, system-level tools, or high-performance needs.

Key Tool: [Kubo](#), the reference implementation in Go.

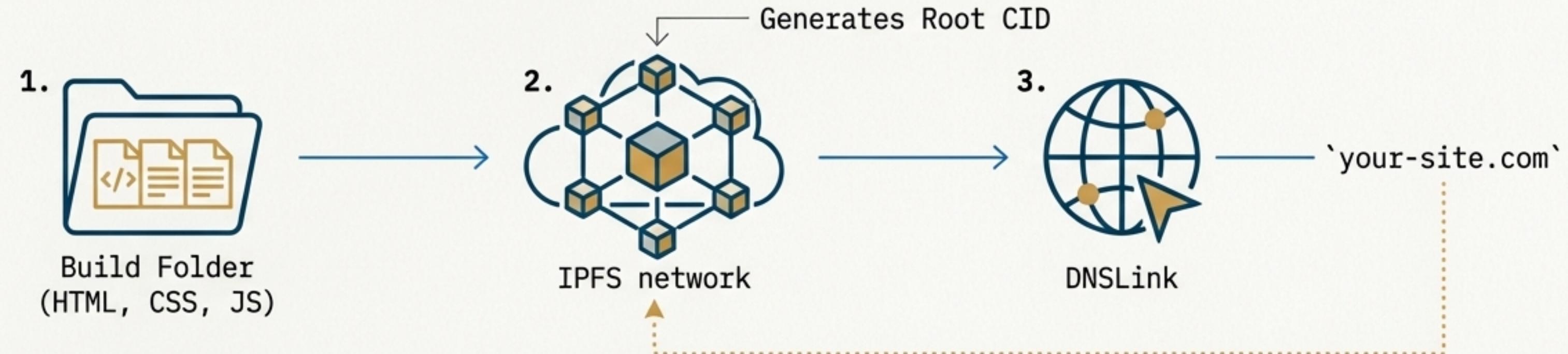
Path 3: Agnostic Apps via HTTP



Control an IPFS Kubo node from any language using its robust [Kubo RPC API](#). Retrieve data in any environment using an [IPFS Gateway](#).

Key Tool: [Kubo RPC API](#)

Powerful Use Case: Deploying Decentralized Static Sites



The Process

1. **Add Your Site:** Add your static site's build directory (HTML, CSS, JS) to IPFS. This generates a single root CID for your entire site.
2. **Ensure Availability:** Pin the root CID using a pinning service to guarantee your site stays online.
3. **Create a Human-Readable Link:** Use **DNSLink** to point a traditional domain name (e.g., `your-site.com`) to your IPFS content's CID. This provides a stable address even when you update your site.

Benefit: Your website is now served peer-to-peer, is censorship-resistant, and has no single point of failure.

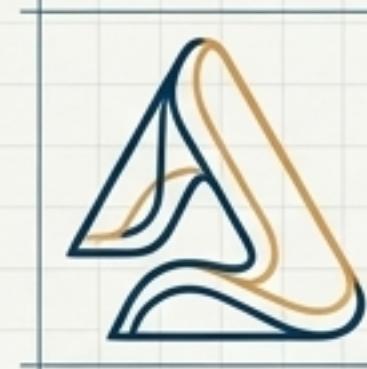


Part III: The Frontier

Exploring the Ecosystem, Real-World Applications, and Community

IPFS in the Wild: Real-World Case Studies

IPFS provides foundational infrastructure for a diverse range of applications.



Audius

A decentralized music streaming platform that uses IPFS for hosting audio content, ensuring it's owned by the artist and controlled by the community.

Music Storage via IPFS CID



Snapshot

A decentralized voting system that uses IPFS to store proposals and votes off-chain, making governance verifiable and gas-free.

Off-chain Governance Data



Fleek

A platform that simplifies building and deploying websites and applications on the open web, using IPFS for hosting and storage.

IPFS-based Web Hosting

Other notable projects: Arbol, LikeCoin, Morpheus.Network.

Your Path Forward: Troubleshooting and Support



→ **When you encounter issues...**

Check the Troubleshooting Guide: Your first stop for common problems related to retrieving or providing data to the network.

→ **When you need expert help...**

Consult the Support Guide: Learn the best practices for asking questions to get the fastest possible assistance from IPFS experts in the community forums and chat channels.

You Are Now Part of the Ecosystem. Get Involved.

IPFS is supported by a global community of developers, designers, writers, and activists. Here's how you can join:



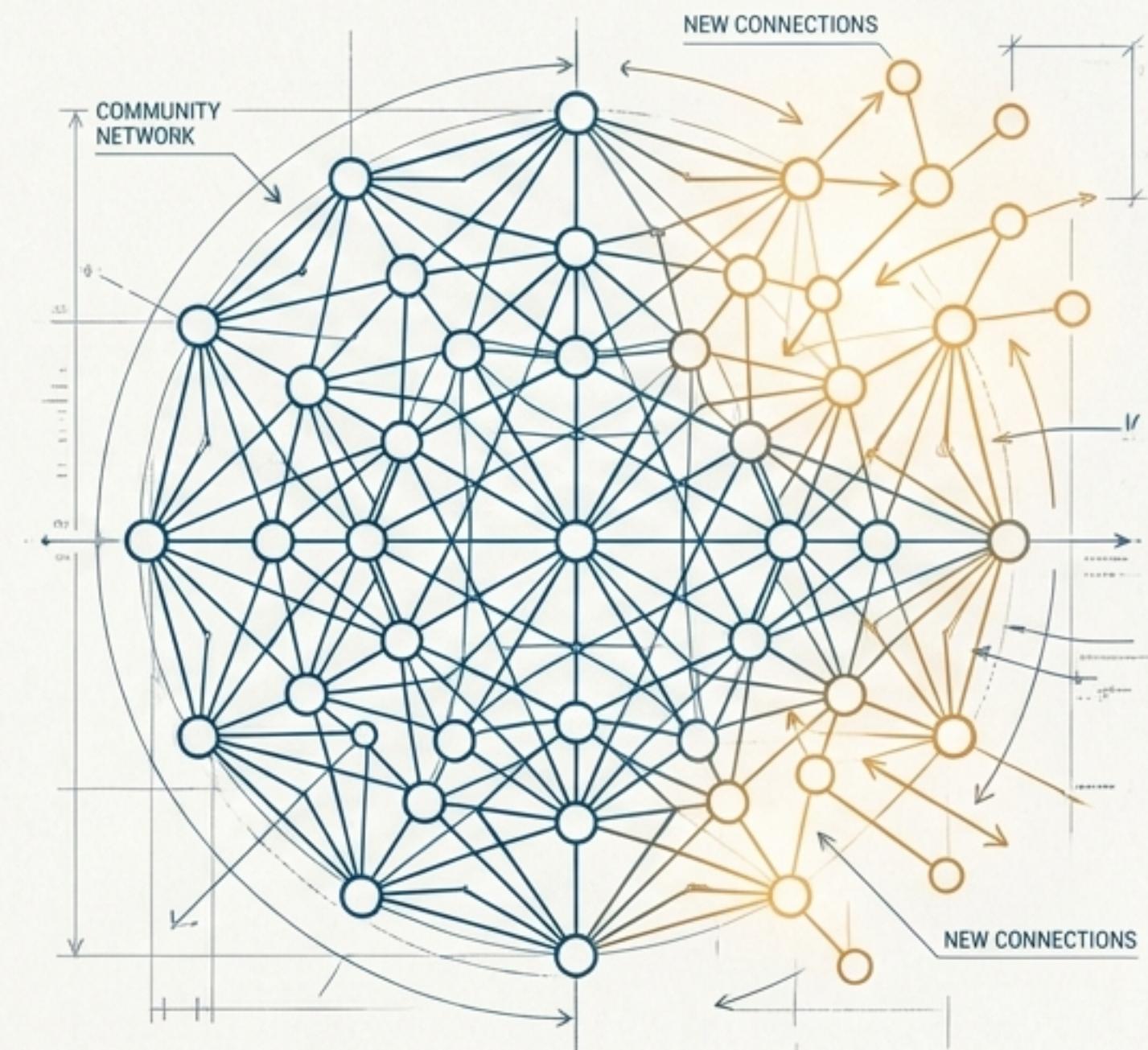
****Find Events & Resources**:** Explore the main Community section of the docs for meetups, conferences, and more.



****Ask Questions & Share Knowledge**:** Join the official forums and discussion channels.



****Contribute to the Project**:** Find opportunities to contribute to the code, documentation, or design.



The future of the web is collaborative. Let's build it together.