

Decentralized Platforms: Comprehensive Comparative Analysis

Section 6 – Platform Deep Profile: IPFS
Version 3.0 | 16 August 2026 | Segment 06

6.1 Overview

Attribute	Value
Full name	InterPlanetary File System
Type	Content-addressed p2p protocol suite / distributed storage layer
Primary identity	None (content addressed; peers have libp2p PeerIDs)
Core primitive	Content Identifier (CID) + Merkle DAG / IPLD
Official site / docs	https://ipfs.tech • https://docs.ipfs.tech • https://specs.ipfs.tech
Primary design goal	Permanent, verifiable, location-independent content addressing

IPFS is infrastructure, not a social network or encrypted personal storage product. Encryption, access control and social features are left to higher layers (Peergos is one such layer).

6.2 History and Design Philosophy

Created by Juan Benet; first released 2015. Central insight: location-based addressing (URLs) is fragile; content should be addressed by what it is (cryptographic hash). Principles: content addressing, Merkle DAGs / IPLD, transport agility, verifiability. IPFS is a suite of specifications and implementations, not a single application.

6.3 Architecture

[illegible]

6.4 Content Identifiers (CIDs)

A CID is a self-describing content address. CIDv1 $\approx$ multibase + version + multicodec + multihash. Same content $\rightarrow$ same CID; any change $\rightarrow$ different CID; anyone can verify by re-hashing.

CIDv0 (legacy): Qm... CIDv1: bafybei... or bafk... File → chunks → leaf CIDs → intermediate nodes → root CID

Large files are chunked; blocks form a Merkle DAG (often UnixFS). The root CID identifies the whole object.

6.5 Mutable Naming & Implementations

IPNS – signed mutable pointer from a key to a CID. **DNSLink** – DNS TXT records pointing a domain to an IPFS path.

Implementation	Lang	Role
Kubo	Go	Reference daemon; CLI, RPC, gateway; production infrastructure
Helia	TypeScript	Modern modular JS/TS; preferred for browsers & Node
Boxo	Go	Libraries for building IPFS apps in Go
IPFS Desktop / Companion	–	Desktop UI and browser extension
IPFS Cluster	Go	Pinset orchestration across nodes

6.6 Persistence, Status, Strengths & Limitations

IPFS does not guarantee permanence. Data stays available only while nodes provide it. Strategies: self-pin, pinning services, Filecoin/incentives, IPFS Cluster. Public gateways are convenient but centralizing.

Status Aug 2026: Mature infrastructure; Kubo + Helia active; improved browser connectivity; large peer/CID counts; some commercial gateways changing models (e.g. Infura IPFS ending); still default substrate for many Web3/NFT/decentralized-web projects.

Strengths: excellent addressing & integrity; location independence; large multi-implementation ecosystem; flexible IPLD data model; strong substrate role. **Limitations:** public by default; persistence not automatic; retrieval depends on providers; public gateways re-centralize; no native access control or social layer; operational complexity for reliability.

6.7 Comparative Position

Dimension	vs Peergos	vs Solid	vs Nostr	vs Syncthing
Content addressing	Native (shared)	Not native	Not native	Not native
Default confidentiality	Public	Depends	N/A	Encrypted
Persistence	Pinning/incentives	Pod provider	Relay policy	Local devices
Access control	None native	WAC/ACP	None (storage)	Device-level
Primary role	Infrastructure	Data ownership	Social protocol	Continuous sync

IPFS is best understood as a verifiable content-addressed distribution and storage substrate. Private storage, personal data ownership, and social systems typically layer on top of or alongside it.

6.8 Key Primary Sources

- <https://ipfs.tech> • <https://docs.ipfs.tech> • <https://specs.ipfs.tech>
- <https://github.com/ipfs/kubo> • <https://github.com/ipfs/helia>
- Content addressing and “How IPFS works” guides; IPLD and multiformats specs

End of Section 6 – IPFS • Next: Section 7 – Secure Scuttlebutt (SSB)