

Peergos Private Encrypted Filesystem

Value Proposition vs Alternatives in This Survey
16 August 2026 · Focus Note (related to Segments 04, 06, 11)

1. The Core Claim

Peergos provides a **private, encrypted, capability-controlled filesystem** whose blocks can live on a content-addressed substrate (IPFS-compatible, local disk, or S3). Data is encrypted **on the device** before it leaves. The storage layer only ever sees ciphertext. Filenames, structure, and social metadata are designed so the server cannot usefully read them. Access is granted by cryptographic capabilities. Only the author or people they explicitly share with can decrypt. Sharing uses hybrid post-quantum key exchange (X25519 + ML-KEM-1024).

2. How the Pieces Fit

User device → Chunk (~5 MiB) → random symmetric keys → encrypt client-side → Encrypted content-addressed blocks → Block store: local | S3-compatible | IPFS-compatible (server / network sees only ciphertext)

Unshared data: quantum-resistant path (hashing + symmetric only). Shared data: hybrid KEM protects against harvest-now-decrypt-later.

3. Comparison Table

Concern	Peergos	IPFS	Syncthing	Solid	Matrix
Default confidentiality	High (client encrypted)	None (public)	High on your devices	Not mandatory	Messages yes; not FS
Who can read	Capability holders only	Anyone with CID	Approved devices	WAC/ACP subjects	Room members
Server sees plaintext?	No	Yes if plaintext added	No / optional enc	Often yes	Metadata yes
Fine-grained share	Yes (capabilities)	Manual	Folder/device	Yes (ACL)	Room-level
Metadata protection	First-class	None	Limited	Limited	Limited
On content-addressed store	Yes	Is the store	No	No	No
PQ sharing deployed	Yes (hybrid)	No	No	No	No

vs IPFS: IPFS gives addressing and distribution, not confidentiality. Peergos is the layer that turns content-addressed blocks into a private capability-controlled filesystem.

vs Syncthing: Excellent multi-device sync among trusted devices; not a global capability-addressable encrypted FS for selective third-party access.

vs Solid: Strong authorization; data encryption at rest on the Pod is not mandatory — the server remains in the confidentiality trust boundary. Peergos removes the server from that boundary for content.

vs Matrix: Strong E2EE messaging; not a general encrypted hierarchical filesystem.

vs Nostr / ActivityPub / SSB: Social protocols; private storage is outside their design centre.

4. Architectural Value

- Threat model: storage provider / network observer learns nothing useful about content or structure.
- Selective disclosure via capabilities without a trusted central ACL service.
- Portable cryptographic identity + encrypted blocks → move hosts, keep friends and access.
- Composable with IPFS: keep content addressing; add confidentiality and access control.
- Forward-looking: hybrid post-quantum sharing already deployed.

5. Honest Limitations

- Almost no independent third-party clients (vertical ecosystem).
- Smaller network effect than ActivityPub, Nostr, or Matrix.
- Full platform stack, not a thin library.
- Availability and DoS still matter; encryption does not solve them.
- Key/capability management and recovery are critical.

6. When Peergos Is the Rational Choice

Choose Peergos when: you need a private hierarchical FS with selective sharing; storage may be untrusted; metadata leakage is in scope; you want capabilities not server ACLs; PQ protection for shared data matters; you can accept a smaller integrated ecosystem.

Choose something else when: public social network effect (AP/Nostr); mature multi-client messaging (Matrix); simple continuous sync of your devices (Syncthing); content addressing without confidentiality (IPFS); structured personal data with app interoper and acceptable Pod-server trust (Solid).

7. One-Sentence Summary

Peergos turns content-addressed (or ordinary) storage into a private, capability-controlled, client-encrypted filesystem that remains unreadable to the storage layer and to anyone you have not explicitly shared with — a property none of the other systems in this survey provide as a first-class, integrated design.

Primary sources: peergos.org, book.peergos.org, 2025 year-in-review, GitHub releases (hybrid ML-KEM), Segments 04/06/11 of this series.