Decentralised storage platforms

BUIDLx 2 Amsterdam

Feb 13th, 2019

Contents

- What is decentralised storage?
- Comparison of decentralised storage platforms

What is decentralised storage?

- Data spread out over multiple nodes
- Availability
- Redundancy
- Blockchain values, for data storage
  - Data integrity / fault tolerance
  - No single point of failure/control

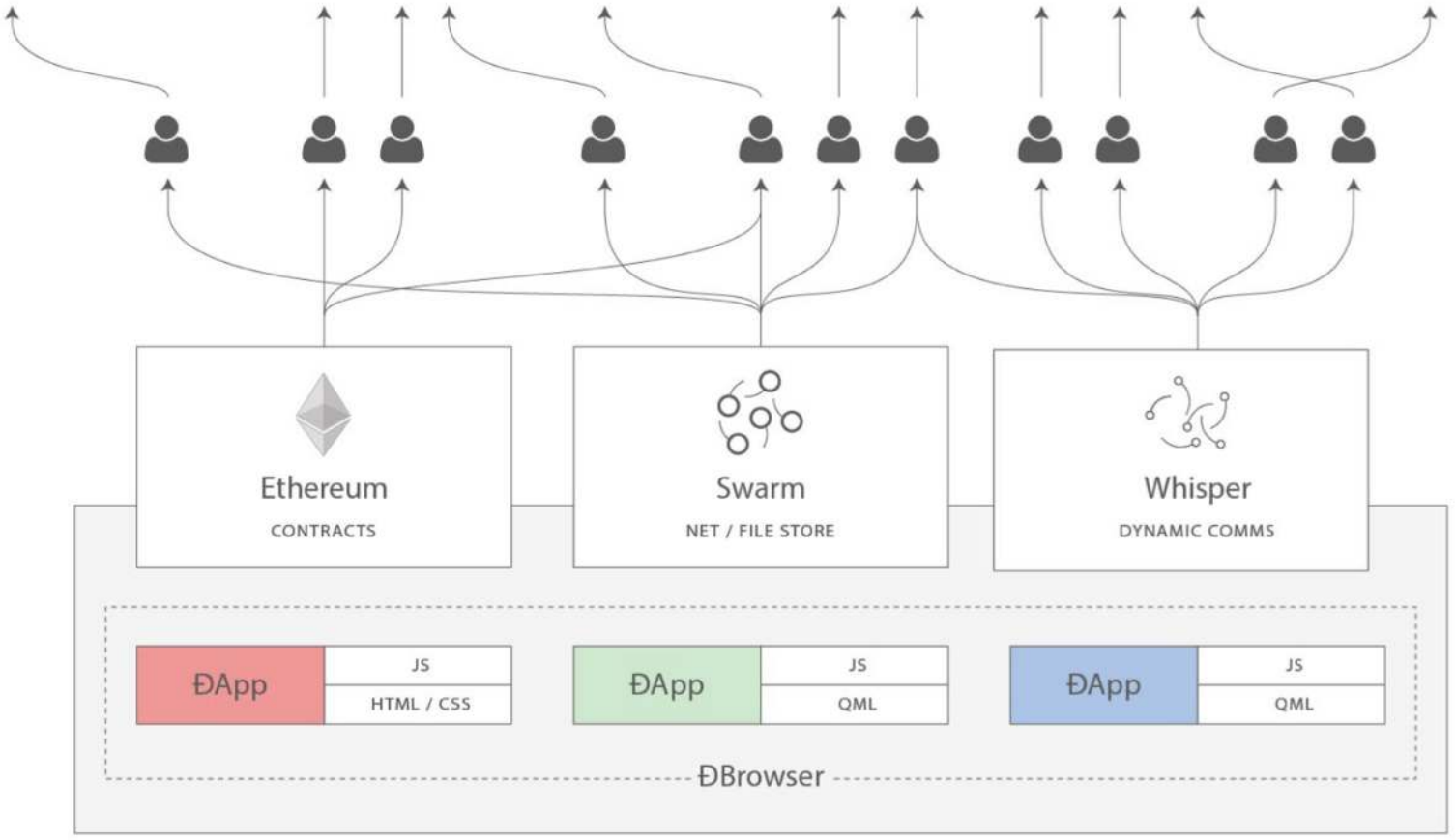
Differences with on-chain data storage

- Costs
- Technical limitations
- Efficiency

Decentralised storage platforms

- Swarm
- IPFS
- Storj
- Sia

Swarm


Ethereum
CONTRACTS
Swarm
NET / FILE STORE
Whisper
DYNAMIC COMMS
ĐApp
JS
HTML / CSS
ĐApp
JS
QML
ĐApp
JS
QML
ĐBrowser

Swarm – Technical architecture

- Content -> chunks (returning a root hash)
- Stored in a *distributed preimage archive*
- Kademlia topology + node discovery
- Redundancy
- Content collections (manifests)

Swarm – Persistence & Incentivisation

- Garbage collection & caching
- Storage Insurance (PoC 0.4)
  - Reward for chunk retrieval
  - Storage promises & collateral
  - Peer to peer negotiations

Swarm – Usage

- Uploading

```
swarm up --bzzapi http://localhost:8500 /path/to/example.md
```

- Accessing

```
swarm --bzzapi http://localhost:8500 down bzz:/<hash>
curl http://localhost:8500/bzz:/<hash>
```

- ENS Support

```
swarm down bzz://theswarm.eth/example.pdf
```

- CLI, JSON-RPC, HTTP, Javascript libraries
- Very early in development, experimental code

IPFS



IPFS – Technical architecture

- Similar to Swarm
- Kademlia topology
- Block Exchange (BitSwap)
  - Nodes choose what to store
- Storage in a MerkleDAG
  - Objects / Links
- Name service (IPNS) / DNS

IPFS – Persistence & Incentivisation

- Garbage collection
- Pinning
- Incentivisation through FileCoin
  - Storage Miners paid to store data
  - Collateral + Proof-of-Spacetime
  - Retrieval Miners paid to retrieve data
  - Public p2p storage & retrieval markets

IPFS – Usage

- Uploading

```
ipfs add /path/to/example.md
```

- Accessing

```
ipfs get <hash>
curl "http://localhost:5001/api/v0/get?arg=<ipfs-path>
```

- Low-level interaction

```
ipfs dht / ipfs bitswap / ipfs block / ...
```

- IPNS Support

```
http://localhost:8080/ipns/your.domain
```

- CLI, HTTP, Javascript / Go libraries + additional bindings

Storj



Storj – Technical architecture

- More traditional cloud storage alternative
- Focus on S3 compatibility
- Kademlia topology
- Files are split up
- *Reed-Solomon erasure coding* for replication
- *Satellite services* for user to node interface
  - Payments, reputation, …
  - Semi-trusted by node and user

k	n	Exp. factor	$P(D \mid p = 10\%)$
1	1	1	90.483741803595962%
1	2	2	98.2476903693578 27%
1	3	3	99.6400506816910 51%
1	10	10	99.999988857452166%
1	16	16	99.999999998036174%

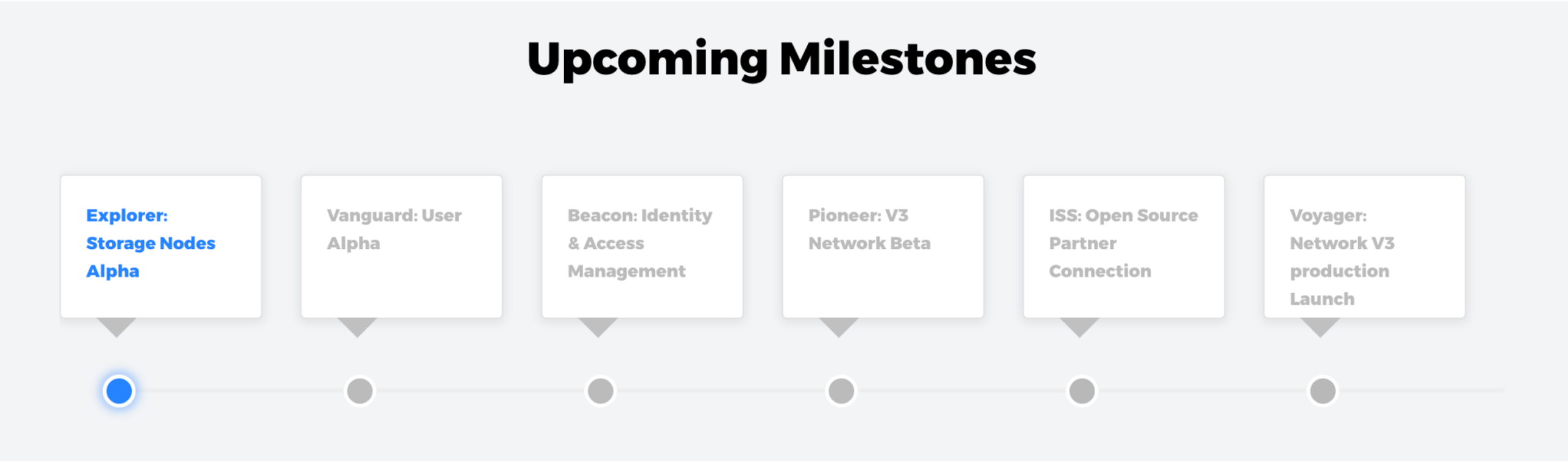
k	n	Exp. factor	$P(D \mid p = 10\%)$
2	4	2	99.207366813274616%
4	8	2	99.8588689854113 26%
8	16	2	99.9954624068782 60%
16	32	2	99.999994620652776%
20	40	2	99.999999807694154%
32	64	2	99.999999999 90544%

Storj – Persistence & Incentivisation

- Time-to-live for chunks
- Allows deletion
- Storage node audits & collateral
- Month-by-month payments by satellite
- Garbage collection for unpaid storage

Storj – Usage

- No user interface available yet



Sia



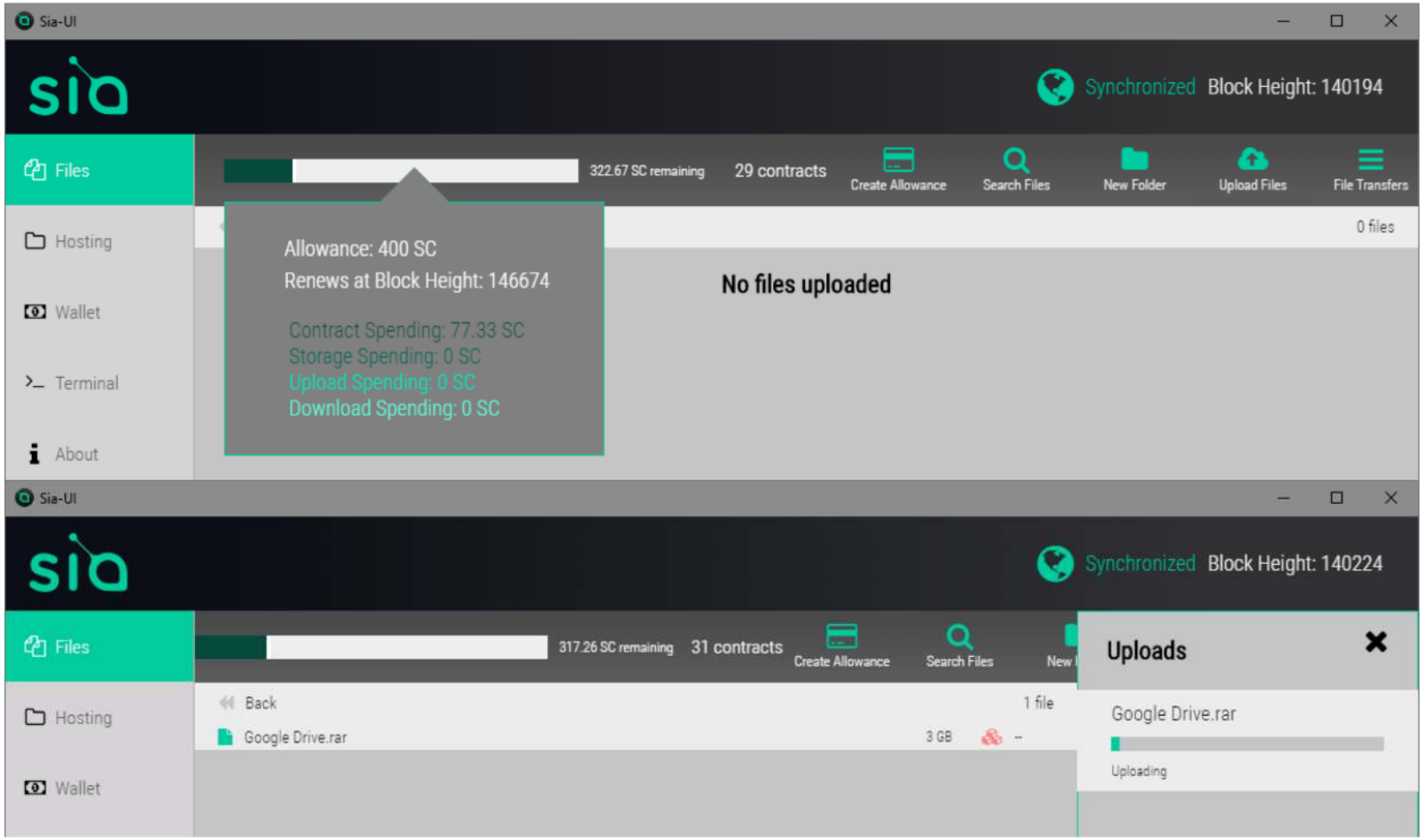
Sia – Technical architecture

- Similar to Storj
- *Reed-Solomon erasure coding*
- Client application handles distribution
  - Splits up files
  - Creates contracts with hosters
  - Pieces files back together

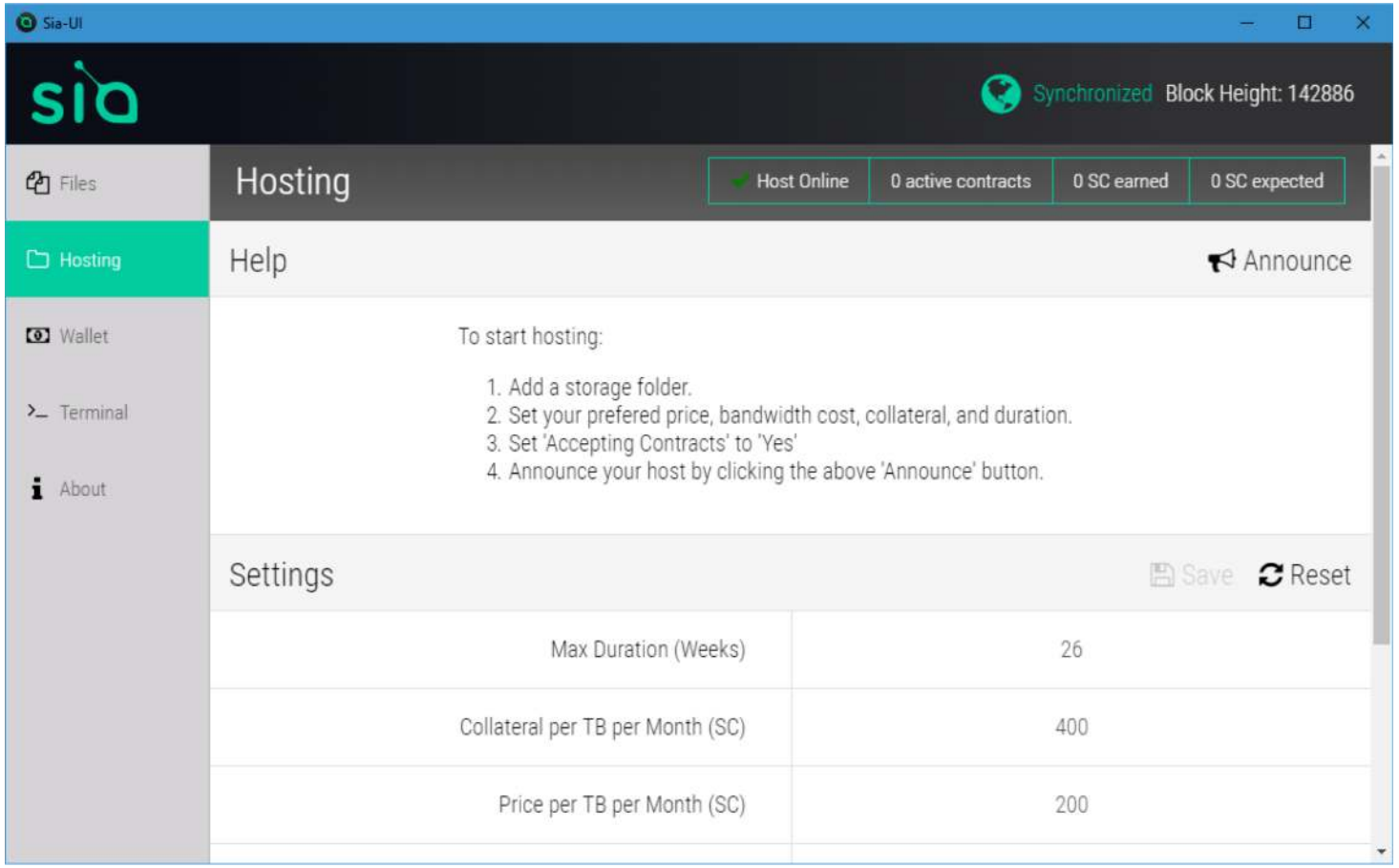
Sia – Persistence & Incentivisation

- Storage contracts (default 90 days)
  - Paid for by user
  - Proof of storage & collateral
- Data deleted after contract
- Data can be changed inside contract

Sia – Usage


Sia-UI
Synchronized Block Height: 140194
Files
Hosting
Wallet
Terminal
About
322.67 SC remaining
29 contracts
Create Allowance
Search Files
New Folder
Upload Files
File Transfers
0 files
No files uploaded
Allowance: 400 SC
Renews at Block Height: 146674
Contract Spending: 77.33 SC
Storage Spending: 0 SC
Upload Spending: 0 SC
Download Spending: 0 SC
Sia-UI
Synchronized Block Height: 140224
Files
Hosting
Wallet
317.26 SC remaining
31 contracts
Create Allowance
Search Files
Back
1 file
Google Drive.rar
3 GB
Uploads
Google Drive.rar
Uploading

Sia – Usage


Sia-UI
Synchronized Block Height: 142886
Files
Hosting
Wallet
Terminal
About
Hosting
Host Online
0 active contracts
0 SC earned
0 SC expected
Help
Announce
To start hosting:
1. Add a storage folder.
2. Set your prefered price, bandwidth cost, collateral, and duration.
3. Set 'Accepting Contracts' to 'Yes'
4. Announce your host by clicking the above 'Announce' button.
Settings
Save
Reset
Max Duration (Weeks)
26
Collateral per TB per Month (SC)
400
Price per TB per Month (SC)
200

Recap and Q&A

- Swarm, IPFS, Storj, Sia
- Swarm and IPFS integrate well with Ethereum
  - Development of decentralised apps
  - Immutable data storage
- Storj and Sia are like regular cloud storage platforms
  - End-user products
  - Deletable and mutable content
- All platforms are early and experimental
  - Lack of storage incentivisation

Swarm Resources & Further Reading

- https://swarm-guide.readthedocs.io/en/latest/
- https://github.com/ethersphere/Swarm/wiki/roadmap
- http://swarm-gateways.net/bzz:/theswarm.eth/ethersphere/orange-papers/1/sw%5E3.pdf

IPFS Resources & Further Reading

- https://docs.ipfs.io/
- https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf
- https://github.com/ipfs/specs
- https://decentralized.blog/understanding-the-ipfs-white-paper-part-1.html
- https://decentralized.blog/understanding-the-ipfs-white-paper-part-2.html
- https://filecoin.io/filecoin.pdf

Storj Resources & Further Reading

- https://docs.storj.io/docs
- https://storj.io/storjv3.pdf

Sia Resources & Further Reading

- https://sia.tech/docs/
- https://blog.sia.tech/how-to-put-data-on-the-sia-network-784499a65b
- https://siasetup.info/guides/hosting_on_sia
- https://siasetup.info/guides/renting_on_sia
- https://siawiki.tech/renter/contracts
- https://blog.sia.tech/getting-started-with-private-decentralized-cloud-storage-c9565dc8c854