WELCOME TO SESSION 06!

Diversification of Distributed Storage Practices

Stanford University articulated one of the most important tenets of digital preservation, LOCKSS or "Lots of Copies Keep Stuff Safe." But keeping many copies of a file in the same location has dire consequences. Diversifying where our cultural heritage, personal and otherwise, is kept is just as important as making copies at all. How can we use decentralized tooling to maximize digital preservation? What tools, both in hardware and software, exist to help facilitate keeping our files safe, and how do we network them together for the purpose of community archiving? We'll learn about the pros and cons of running your own server, maintaining your own preservation hardware, offline storage, online storage, and security. Two case studies, the Current Museum and TRANSFER Data Trust, will ground these tools in real-world examples.

OVERVIEW

01 Intro

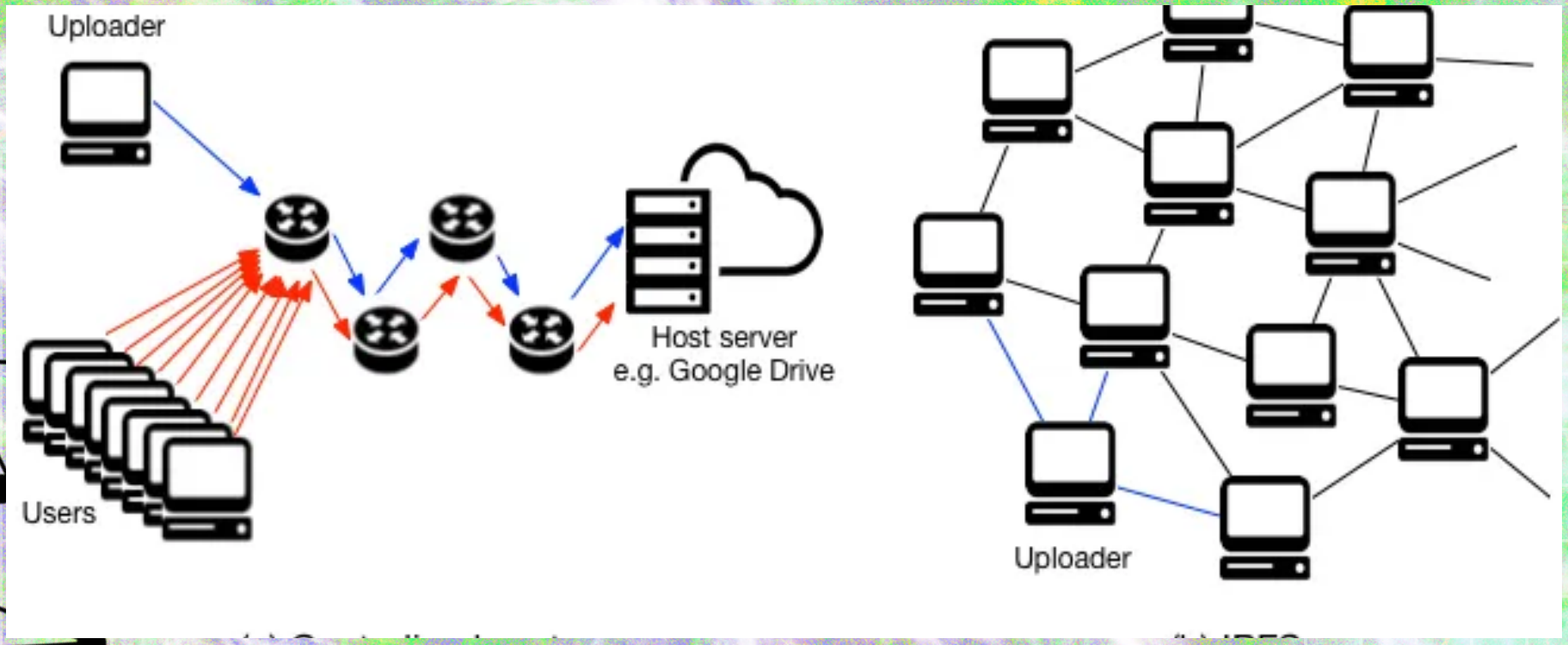
02 Digital Dark Age

03 Traditional Storage

04 Disk Arrays

05 Distributed File Systems

06 Decentralized Storage



LOCKSS

(LOTS OF COPIES KEEP STUFF SAFE)

The Printing Press

The printing press revolutionized information storage by distributing knowledge beyond handwritten manuscripts. This shift away from centralized control, with scribes copying limited texts, democratized access to information and marked a move towards a more decentralized storage model

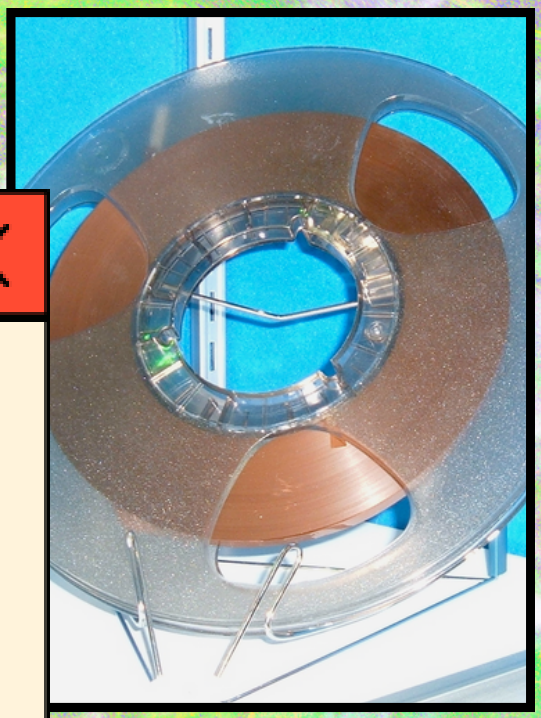
Libraries

By collecting and preserving redundant knowledge across various physical locations, libraries fostered the spread of ideas and challenged the centralized control of information held by a single source.

ARE WE IN THE DIGITAL DARK AGE?

Vast amounts of data, estimated at over 70% by some metrics, are already considered inaccessible due to format obsolescence, lost decryption keys, and inadequate archiving practices. This poses a serious challenge to preserving historical documents, emails, social media posts, and other digital artifacts that chronicle our era. Archivists are racing against time to develop robust preservation strategies, migrate data to new formats, and raise public awareness of this critical issue. The sheer volume of information and the rapid pace of technological change make this a daunting task, but the potential loss of our digital heritage is too great to ignore.

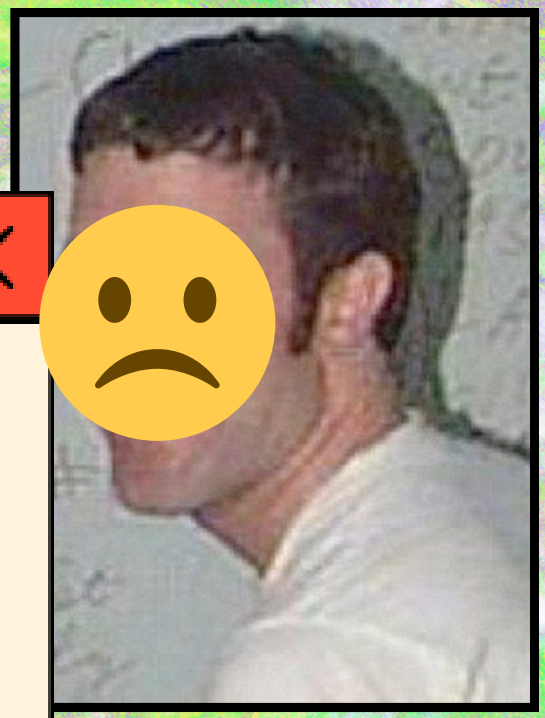
DATA LOSS EXAMPLE



UNIVERSAL MUSIC GROUP

A 2008 fire at Universal Studios Hollywood archives destroyed an estimated 120,000 to 175,000 irreplaceable primary recordings. This included original works by music legends like Buddy Holly, Chuck Berry, John Coltrane, and countless others, representing a significant loss for the music industry and history.

DATA LOSS EXAMPLE



MYSPACE

MySpace's poorly communicated server migration in 2011 resulted in the loss of a staggering 200TB of user data. This included irreplaceable photos, music, and other social media related assets.

DATA LOSS EXAMPLE

PIXAR: TOY STORY

Pixar narrowly avoided catastrophe in 1998 when a file deletion command mistakenly erased most of Toy Story 2's animation data. The team's initial panic subsided when they discovered a backup on a home computer, saved by a supervising technician working remotely due to maternity leave.

STORAGE RETRIEVAL TYPES

HOT

frequently accessed assets that needs to be retrieved quickly

NEARLINE

for data retrieval once per month or less, using “spin-down” to save money

COLD/ARCHIVAL

infrequently accessed information for long-term preservation, prioritizing cost-efficiency over speed. Think of it as a remote vault for rarely needed data, accessed only a few times a year or even less.

OFFLINE/TERTIARY

rarely accessed, on physical media like magnetic tape, offering the lowest cost but slowest access. Think of it as a dusty box in the storage room, holding data accessed only in exceptional circumstances, perhaps every few years or even decades.

CENTRALIZED STORAGE

Direct Attached Storage

Solid State Drive (SSD)

Hard Drive (HDD)

Network Attached Storage

A dedicated file-level server on a network that provides data storage access to multiple devices

Storage Area Networks

high-performance networks specifically designed to connect and share block-level storage devices among multiple servers.

Optical or Tape

utilize lasers or magnetic fields to encode data on physical discs or tapes

BREAKING DOWN DATA

Striping

splits data across multiple storage devices for increased speed and redundancy

Chunking

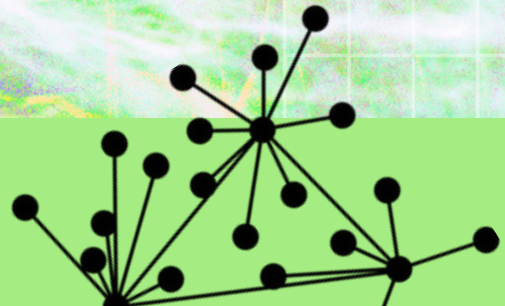
breaking down large data files into smaller, more manageable pieces

Sharding

partitions a large database into smaller, more manageable segments spread across multiple servers for improved performance and scalability.

Blocking

split into smaller pieces and stored across multiple devices or servers



INDEXING/FINDING

FAT (WINDOWS)

breaks file down into smaller pieces and stores them in available clusters. The FAT table is then updated to mark those clusters as "used" and link them together to represent the entire file

NTSF (WINDOWS)

keeps a log of changes and allows you to control who can access different files, making it more secure and flexible

EXT4 (LINUX)

keeps a backup list of changes (journal) before updating the main catalog (file system), making searches and organization (querying and indexing) more reliable in case of errors

HFS+/APFS (IOS)

uses copy-on-write, inline metadata, and efficient cataloging to keep file information up-to-date and readily accessible

DISK ARRAYS

A storage system that groups multiple hard drives or solid-state drives together, functioning as a single high-capacity storage unit with improved performance and redundancy compared to individual drives.

RAID

(REDUNDANT ARRAY OF INEXPENSIVE DISKS)

RAID 0 (striping): splits data across multiple disks in a way that optimizes read and write performance, divided Into stripes.
No data redundancy. Failure of a single disk in the array results in complete data loss.

RAID 1 (mirroring): creates an exact duplicate of the data on a secondary disk.
Failure of a single disk in the array does not result in data loss.

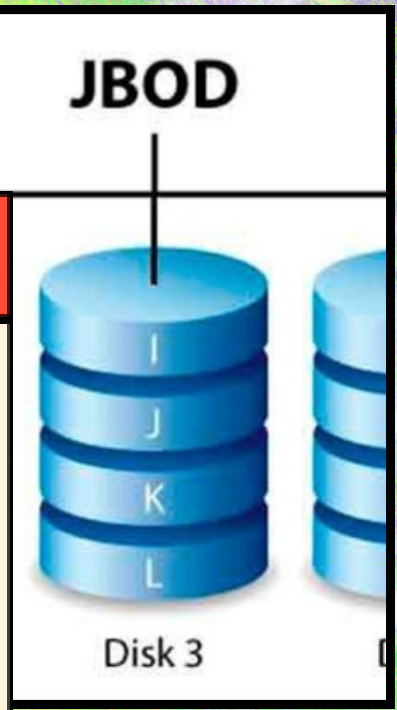
RAID 3 stripes data across multiple disks and dedicates a single disk for storing parity information.

RAID 5 addresses the write bottleneck issue of RAID 3 by distributing parity information across all data disks in the array

JBOD

(JUST A BUNCH OF DISKS!)

A storage configuration that combines multiple disks into a single large volume for increased storage capacity, offering a cost-effective and scalable solution. However, unlike RAID, JBOD doesn't provide data redundancy, so if a disk fails, the data on that disk is lost.



NETWORK ATTACHED STORAGE (NAS)

Processing Power: NAS devices typically have weaker CPUs compared to computers. They are optimized for data storage and retrieval, not running demanding applications.

Operating System: NAS devices run lightweight operating systems focused on file management and network connectivity. These systems lack the features and capabilities of a full desktop OS needed for everyday computing tasks.

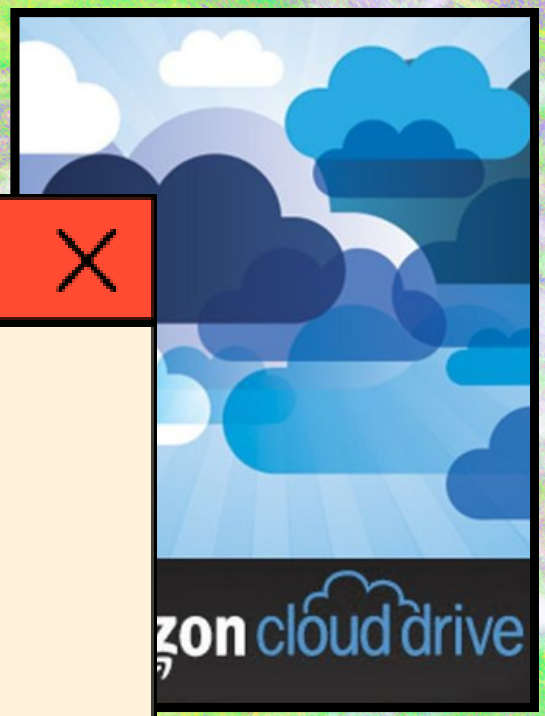
Limited Input/Output: NAS devices are designed for network file access. They typically have limited I/O options like one or two Ethernet ports and might lack USB ports or dedicated video outputs for connecting a monitor and peripherals.

RAM: NAS devices generally have less RAM compared to computers. This limited memory can hinder performance when running multiple processes.

DISTRIBUTED FILE SYSTEMS

A distributed file system allows multiple computers in a networked environment to access the same files or directories located on different physical devices within the network. It can be used by companies with large amounts of data stored on multiple servers or devices such as laptops, smartphones or tablets. The purpose of a distributed file system is to allow users to access shared files from any device connected to the network without having to physically move the files around. It also helps protect against data loss by providing redundancy across multiple systems so that if one system fails, the data can still be recovered from another source.

CLOUD STORAGE



Cloud storage leverages a distributed file system architecture with a multi-tenant access model. Data is fragmented and replicated across geographically dispersed data centers managed by cloud service providers (CSPs). These CSPs offer RESTful APIs or platform-specific SDKs for programmatic data interaction and integration with applications.

Cloud storage is a model that allows users to store data on remote servers, usually hosted by a third party, instead of on their own computer's hard drive or other storage device. Users can access their data via an internet connection, and upload files to the servers using an internet connection.

DECENTRALIZED STORAGE

Data is fragmented and distributed across a network of independent devices (nodes).

No single point of control, users retain ownership and control over their data.

Files are protected by a network formed of lots of different stakeholders rather than a single company.

Censorship-resistant file sharing.

TORRENTING (P2P)

Torrenting operates on a principle similar to decentralized storage. Instead of downloading a file from a single source (like a cloud server), you download it in pieces from a network of other users who already have parts of the file. Each user's computer acts as a temporary storage location for the file fragments. Once you have all the pieces, your computer assembles them into the complete file. This distributed approach eliminates the need for a central server and leverages the collective storage capacity of the users in the torrent network.

IPFS

Content-Addressing: In IPFS, files are addressed by their content, not their location. Each file has a unique cryptographic hash (CID) that acts like a fingerprint. This ensures data integrity and allows anyone with the CID to retrieve the exact file, regardless of its physical location on the network.

Pinning: Since data is distributed across the network, files can become unavailable if the nodes storing them go offline. "Pinning" allows users to explicitly store a copy of a file on their local node or a dedicated pinning service. This ensures continued access to important data even if some nodes disappear.

IPNS (InterPlanetary Namespaces): IPFS uses IPNS to create human-readable names for content addresses. Imagine an IPNS as a nickname for a complex CID. This makes it easier to share and reference content within the network, similar to how domain names work on the traditional web.

DAT PROTOCOL (P2P)

Dat bridges the gap between BitTorrent's scalable P2P distribution and Git's version control functionalities, offering a unique solution for decentralized data management.

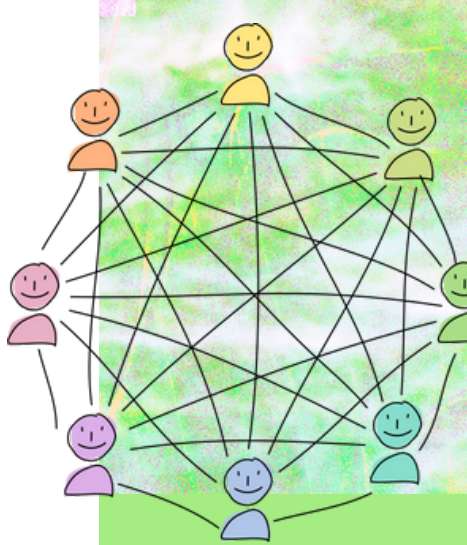
How is Dat different than IPFS?

IPFS and Dat share a number of underlying similarities but address different problems. Both deduplicate content-addressed pieces of data and have a mechanism for searching for peers who have a specific piece of data. Both have implementations which work in modern Web browsers, as well as command line tools.

The two systems also have a number of differences. Dat keeps a secure version log of changes to a dataset over time which allows Dat to act as a version control tool. The type of Merkle tree used by Dat lets peers compare which pieces of a specific version of a dataset they each have and efficiently exchange the deltas to complete a full sync. It is not possible to synchronize or version a dataset in this way in IPFS without implementing such functionality yourself, as IPFS provides a CDN and/or filesystem interface but not a synchronization mechanism.

Dat prioritizes speed and efficiency for the most basic use cases, especially when sharing large datasets. Dat does not duplicate data on the local filesystem, unlike IPFS which duplicates on import (although IPFS now has experimental support for no-copy imports). Dat's pieces can also be easily decoupled for implementing lower-level object stores. See hypercore and hyperdb for more information.

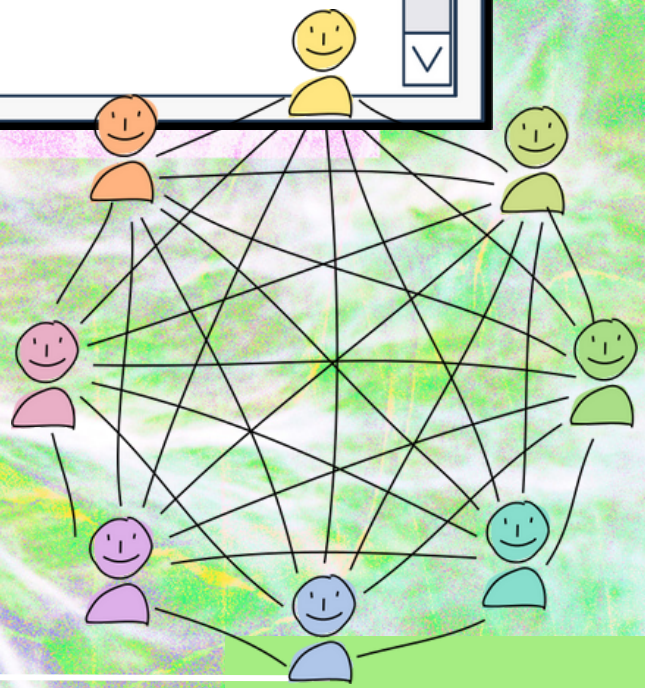
In order for IPFS to provide guarantees about interoperability, IPFS applications must use only the IPFS network stack. In contrast, Dat is only an application protocol and is agnostic to which network protocols (transports and naming systems) are used.



TAHOE-LAFS

BitTorent, except created for file storage and focused on encryption, with subdirectories and automatic redundancy up to 10x .

Can utilize RAIN arrays (Reliable Array of Independent Nodes), a model for building a fault-tolerant cluster of storage and compute systems that uses error-correcting codes to partition data across nodes. It's similar to RAID (Redundant Array of Independent Disks), but RAIN is applied across nodes instead of disk arrays.



FILECOIN (COLD STORAGE)

Filecoin operates as a Layer 1 (L1) blockchain, utilizing a unique consensus mechanism called "Proof-of-Replication." In this system, storage providers prove they are reliably storing the data they hold, earning FIL tokens as rewards. This mechanism incentivizes network participation and data integrity.

Retrieval Deals: Users who want to store or retrieve data interact with storage providers through "retrieval deals." These deals specify the data, storage duration, and retrieval costs using FIL tokens.

Open-source and Programmable: The Filecoin network is open-source, allowing anyone to contribute to its development. Additionally, it offers programmability through smart contracts, enabling developers to build decentralized applications on top of the L1 blockchain.

FILECOIN (COLD STORAGE)

Storage Miner:
CPU: Multi-core processor with high clock speeds (e.g., AMD Ryzen Threadripper or Intel Xeon)
RAM: Minimum 32GB DDR4 RAM, ideally 64GB or more for optimal performance
Storage: Several terabytes (TB) of high-performance NVMe solid-state drives (SSDs) for storing data efficiently. Additionally, a separate SSD for the operating system is recommended.
Network: Reliable and high-bandwidth internet connection (Gigabit Ethernet or higher recommended)

Retrieval Miner:
CPU: Multi-core processor with decent clock speeds (e.g., AMD Ryzen 5 or Intel Core i5)
RAM: 16GB DDR4 RAM is a good starting point, potentially expandable based on workload
Storage: While storage space can be helpful for caching retrieved data, the primary focus is on network bandwidth. A moderate-sized SSD is sufficient.
Network: Reliable internet connection with good upload speeds is essential (consider fiber optic options)

Verifier Node:
CPU: Multi-core processor (e.g., AMD Ryzen 3 or Intel Core i3)
RAM: 8GB DDR4 RAM should be sufficient
Storage: An SSD with at least 1TB of storage is recommended to store the blockchain data.
Network: Stable internet connection
Additional Considerations:

Operating System: Filecoin supports various operating systems like Linux (Ubuntu, Debian), Windows 10 (with limitations), and macOS (unofficially). Linux is generally the preferred choice for stability and performance.
Security: Filecoin node security is crucial. Regularly update your operating system and software, implement strong passwords, and consider hardware security measures like a dedicated machine for your node.
Software: You'll need to install the Filecoin Lotus software, which can be downloaded from the Filecoin website.

ARWEAVE

Blockweave Architecture: Arweave leverages a blockweave structure instead of a traditional blockchain. Data blocks interconnect in a web-like fashion, allowing for faster validation and data retrieval compared to linear chains.

Proof-of-Access Consensus: Unlike proof-of-work in blockchains, Arweave miners secure the network through proof-of-access. Miners demonstrate they have access to specific historical data before adding new blocks, incentivizing storage and network integrity.

Data Endowment: Arweave prioritizes “permanent” data storage. Users pay a one-time fee in AR tokens to store data. A portion covers initial storage, while the majority goes into a sustainable endowment that earns interest and perpetually funds future storage needs.

Decentralized “Permaweb”: Arweave facilitates the permaweb, a permanent and decentralized network of websites and applications accessible through standard browsers. This enables censorship-resistant information sharing.

Native AR Token: AR tokens serve as the fuel for the Arweave ecosystem. Users pay AR for data storage, and miners receive AR rewards for maintaining the network. This economic model incentivizes network participation and long-term data preservation.

SOLID (PODS)

Data Ownership: SOLID emphasizes user ownership of data. Users store their data in their own pods, not on servers controlled by a single entity.

Pod Access and Management: The specifications define how users control access to their pods. This includes user authentication, authorization mechanisms, and how users grant access to specific data or applications.

Data Formats and Queries: The specifications outline the data formats supported by SOLID pods and how applications can query and interact with that data. This ensures interoperability between different applications and pod providers.

Security and Privacy: Security protocols and best practices are outlined to protect user data within pods. This covers user authentication, data encryption, and access control mechanisms.

HDFS (HADOOP)

Focus: HDFS excels at storing and managing massive datasets within an organization or cluster. IPFS and Filecoin, on the other hand, prioritize decentralized data storage accessible to anyone on the network.

Centralization vs. Decentralization: HDFS relies on a centralized architecture with Namenodes and Datanodes. IPFS and Filecoin have no central point of control, making them more resistant to censorship and data loss if a server fails.

Data Ownership: In HDFS, data technically resides within the cluster controlled by the organization. With IPFS, users retain ownership and control over their data on the decentralized network.

Scalability: HDFS scales well for large datasets within a cluster, but adding new nodes can become complex. IPFS, by design, is inherently scalable as more users join the network and contribute storage capacity.

Cost: While HDFS requires investment in hardware and maintenance, IPFS leverages a peer-to-peer model. Filecoin incentivizes storage with FIL tokens, potentially leading to a more cost-effective solution for long-term storage, especially for smaller datasets.

CONSIDERATIONS

Data Privacy: By default, IPFS stores data in a publicly accessible network. Anyone can access the data if they have the content address (CID). For sensitive data, encryption is crucial before uploading it to IPFS.

Data Recovery: While designed for permanence, data recovery on Arweave can be complex. Lost encryption keys could result in permanent data loss. Implementing robust key management strategies is crucial.

Reliance on Nodes: Data retrieval depends on the availability of nodes storing the data. Malicious nodes could potentially tamper with data or return incorrect results. Using trusted pinning services can mitigate this risk.

Performance: Retrieving data from a decentralized network can sometimes be slower than accessing data from a centralized server. This is because the network needs to locate the data across potentially many different storage locations.

ENVIRONMENTAL IMPACT

Reduced Hardware Dependency: Distributing data across multiple devices can leverage existing computing power, potentially reducing the need for massive, energy-hungry data centers.

Renewable Energy Integration: Decentralized storage networks can be designed to integrate with renewable energy sources, leading to a more sustainable storage infrastructure.

However, there are also environmental concerns:

Increased Network Activity: Decentralized networks often involve more data replication and transfer compared to centralized storage, potentially leading to higher overall energy consumption.

STORAGE TYPES FOR EXERCISE

CENTRALIZED

- Cloud Storage
- Hard Disk

DECENTRALIZED

- IPFS
- Areweave
- Dat
- Torrenting

BLOCKCHAIN-BASED

- Filecoin
- DatDot