What is IPFS

IPFS stands for the Inter Planetary File System. It’s a revolutionary technology disrupting the way internet sites are accessed. The most widely used, ubiquitous protocol of accessing internet is HTTP. Let’s travel little bit into the history. HTTP stand for Hypertext Transfer Protocol. It’s a standard or common language for transferring information between clients and servers which gave a birth to the modern web. HTTP was originally proposed by Tim Berners Lee and Robert Caillaiu in 1990. HTTP allowed the exchange of Hyper Media. Hyper media refers to the non-linear medium of information consisting of not only plain text, but audio, video, graphics and hyperlinks. Multimedia is a broader non-linear medium minus the hyperlinks. Examples of non-linear medium includes hard drive, compact disk and many others. Linear-medium are things of the past like audio/video cassettes, Magnetic tape data storage.

HTTP adopted its data representation from another very successful encoding and data representation protocol called MIME. It stands for Multipurpose Internet Mail Extensions. MIME was invented to allow emails carry multimedia information.

MIME is an extension of SMTP, Simple Mail Transfer Protocol. SMTP used in sending and receiving an email was proposed in 1982 is based on TCP/IP protocol. TCP/IP developed by Robert E Kahan was born out from a project called ARPANET funded by US govt. agency ARPA, Advanced Research Project Agency. ARPANET became a reality in 1969 when four US universities were linked. Under the hood of ARPANET was the idea of exchanging information between the computers in the form of packets. These packets routed through different paths and reconstructed at the destination. Packet in network language is smallest unit of data. It is the smallest unit of information getting exchanged between the 2 different computers. ARPANET which gave a birth to TCP/IP really became the basis or the underlying language of the Internet.

Tim Berners Lee envisioned internet as more of decentralized where every machine in world will participate. HTTP protocol coupled with the browser is a big contributor to the internet revolution, created whole new set of businesses, had enormous social impact, but somewhere along the way internet became more and more centralized where few players (Yahoo, Google, Facebook, Microsoft and many others) started controlling most of the traffic.

Large exchange of information got tied up in a client-server model. Consumers of internet have less freedom on there own data and more in the hands of large players. Government agencies in rogue countries at their own will can shut an internet server at any time and deny citizens of their legal rights to have an information.

Fast Forward 2015 enter IPFS. A Stanford computer science graduate Juan Benet who migrated from Mexico at the age of 13 gave a world IPFS. IPFS is like HTTP, but instead of using location to connect to objects, it uses Peer-Peer (P2P) model (No more Client-Server) to exchange an information. Any node participating in an IPFS network can host a web site, document or any file and allow other users to share same. This means no control and no single point of failure.

These days’ Tim Berner is owing a company [SOLID](https://solid.mit.edu/). Its goal is to radically change the way current web works and allows us to take true ownership of our data.

[*P2P – Centralized Internet*](file:///C:\Users\munis\OneDrive\Articles-I%20write\Images\P2P-CenterlizedInternet.png)

In current centralized internet model addresses are accessed based on the location. Example sending a request [www.someabcsite.com](http://www.someabcsite.com) from a browser gets translated into its IP address, request is routed to central location which sends the response. In case site is rich in content a good amount of data travels between your machine and the server. It’s a waste of bandwidth. This is not the same case with IPFS. Instead of request going to a central server requests gets fulfilled from the nearest node. Chances are nearest node has the file. Once you download the contents on your computer you are both server and a client. IPFS motto is to find not where the contents are, but what is it you want to find. That’s why IPFS is rightly called “The Distributed, Permanent Web”. Here is nice [video](https://www.youtube.com/watch?v=skMTdSEaCtA) of why the IPFS by Juan Benet.

**Contents addressing in IPFS**

At the HTTP layer files/contents are referred through their address. Let’s say I want to refer to some document. I will cryptographically hash the document. Hash becomes a small and secure fingerprinting of the whole document. Example one of the IPFS drafts written by Juan Benet can be accessed as <https://ipfs.io/ipfs/QmV9tSDx9UiPeWExXEeH6aoDvmihvx6jD5eLb4jbTaKGps>

In red is the address of the draft or also called Content Identifier (CID). There is no IP involved. Instead of referring to the server where the file is located you are referring to the file itself. By typing above hash in the address of browser network is being asked which node has the content. Network connects you to the node and file gets downloaded. Future is not far where URL’s will be replaced by Hashes.

Every content address in IPFS starts with Qm. This is because files and objects in IPFS are hashed using [MultiHash](https://multiformats.io/multihash/) format and Base58 encoding.

**IPFS-Under the Hood**

Beneath the IPFS is an amalgamation of proven technologies Merkle-DAG, [Distributed Hash Table (DHT)](https://en.wikipedia.org/wiki/Distributed_hash_table), BitTorrent and Git.

One of the core principals of IPFS is how it structure and links the data. Its data structure called Inter Planetary Linked Data ([IPLD](https://ipld.io/)) is modeled on Merkle-DAG, named after an American mathematician, Ralph Merkle. DAG stands for Directed Acyclic Graph. Why Merkle tree? Its because Merkle tree is a proven technology within P2P/Blockchain as it ensures immutability and verify the transactional integrity of data. This data structure also has useful property called Deduplication. Through this property exact data contents are stored only once in a Merkle tree. IPFS stores data in blocks. Each block has a size of 256 Kb. Here is an example of how a file will be organized in IPFS using Merkle tree. Let’s say you want to store a file in IPFS. Total size of the document is 980 Kb. Since each block can only contain < 256 Kb this file will be divided into 4 blocks (1024/4). Each block will be hashed. All those 4 hashes will again be hashed to get a single hash. This last single hash will be a root hash. All 4 hashes will be linked to root hash in a tree like structure. You will provide this root hash to let someone read the contents of MS word document. For simplicity sake hash address shown below are just made up ones. In next article this will become clearer when I will show with an actual example in which we will install and add file into IPFS.

Link to your ipfs document. <https://ipfs.io/ipfs/QmB8>.......

An MS word file with some contents

File contents in bock-1

File contents in bock-2

QmB1…….

File contents in bock-3

QmB1…….

File contents in bock-4

QmB1…….

QmB8……. (Root Hash)

QmB1……. QmB2……. QmB3……. QmB4…….

In above root hash block is referring to the sub or leaf nodes. There can be sub nodes of sub nodes and so on which can make look like Merkle tree as below.

[INSERT IMAGE OF MERKLE TREE FROM THIS LINK](file:///C:\Users\munis\AppData\Roaming\Microsoft\Word\Images\MerkleHash_Tree.png)

Merkle-DAG is a fundamental part of blockchain. Bitcoin employs Merkle tree in maintaining its transactions. In reality when you add a file to IPFS, IPLD is all happening behind the scenes. Its jobs of IPFS to bring all of its sub contents together when a file is requested. A common user will hardly interact with Merkle-DAG.

**Routing and Searching**

IPFS uses Distributed Hash Table (DHT) for routing and locating the particular file requested by node. DHT is maintained when new data is added or when nodes join and leave the network. DHT will be looked to find which nodes have the content. Example when contents for hash QmV9tSDx9UiPeWExXEeH6aoDvmihvx6jD5eLb4jbTaKGps are needed, DHT table will be searched, a node address will be found and contents will be delivered. According to the white paper, small data values less than 1 kb are stored directly on the DHT. For larger values DHT stores references to the NodeIds of peers who can serve the block. Under the hood IPFS uses 9, SKademlia (extension of Kademlia) and CORAL implementation of DHT.

***PUT A IMAGE OF DHT OR DESCRIBE HOW INTERNALLY IT WORKS***

**Data Exchange**

IPFS uses Bitswap as its data exchange model. Bitswap gets the data blocks when requested by the network. Data Routes for the request is provided to Bitswap from the DHT. In second step Bitswap delivers blocks to nodes that want them.

Bitswap is the data exchanging model used in IPFS, it manages to request and send blocks to and from other peers in the network. Bitswap serves two main purposes, first is to get blocks which are requested by the network (it have routes because of DHT). The second is to send those blocks to nodes that want them.

Bitswap is a message-based protocol, as opposed to response-reply. All messages contain want lists or blocks. Upon receiving a want list, a node should consider sending out wanted blocks if they have them. Upon receiving blocks, the node should send out a notification called a ‘Cancel’ signifying that they no longer want the block. At a protocol level, bitswap is very simple.

**Some Quick Features**

In its whole network IPFS does not keep the redundant file. It also maintains versions of file. In case file was changed 3 times all 3 versions will be maintained and can be trace back.

As IP address is hard to remember so is the IPFS hash. A human readable name (Ex: www.yahoo.com) is typed in the browser which gets translated into its IP. In IPFS, IPNS comes as rescue to locate an IPFS hash.

Storing even small files on the Ethereum Blockchain networks can cost lots of gas resulting in a costly transaction. IPFS can complement Ethereum smart contracts world by storing files in its network and Ethereum will store only the hash on its network

Interesting [link](https://www.w3.org/2016/04/blockchain-workshop/slides/benet-ipfs.pdf) which describes the transition from Web 1.0 – 3.0, world of blockchains and IPFS

USE THIS LINK TO DESCRIBE HOW IPFS use Merkle tree to structure large files. Use this in second article while showing an example

<https://flyingzumwalt.gitbooks.io/decentralized-web-primer/content/ipfs-dag/lessons/files-as-dags.html>. Also this article which <https://medium.com/textileio/whats-really-happening-when-you-add-a-file-to-ipfs-ae3b8b5e4b0f>

How to keep files online forever?

*IPFS is a fantastic platform for hosting decentralized files without worrying about Ddos attacks and server problems. It just works and it’s ideal for static websites.*

*Dapps that you want to be fully decentralized. The problem is that once you add a file to the network, it disappears after about 24 hours if nobody else has it pinned. It gets garbage collected by the network.*

*So if you host a website on IPFS with the command:*

IPFS can be a standalone decentralized file system.

It can be complementary to the existing HTTP based centralized system.

We discussed it in the context of blockchain systems because it can serve an important role of decentralized storage for blockchain application that have a lot of data, but will store only the hash on the blockchain.

In this case instead of a centralized store, IPFS can be the decentralized store that work in tandem with the decentralized ledger technology of the blockchain to create a powerful solution for many storage-rich business usecases.

*Seeder/Seed*

*Why data is permanent/Immutable in IPFS?*

*Also mention how cost of smart contracts can be saved using IPFS hash links*

*Explain below what has been read so far*

*Merkel-DAG,BitSwap, GitHub*

*Can HTTP be decentralized?*

*EXPLAIN ALL ABOUT How it is different from BitTorret. One of the main differences is Bitorrent works within its own SWARM. IPFS is a much larger community*

*How does IPFS node knows what is the nearest location*

*Filecoin (RENT OUT A HARD DISK)*

*Drawbacks of IPFS*