# Understanding IPFS in Depth (1/6): A Beginner to Advanced Guide

## A Complete Guide Including IPLD, Libp2p, MultiFormats & Filecoin

This post marks the first in a new “***Understanding IPFS in Depth***” series which will help anybody to understand the underlying concepts of IPFS. We will keep the series as fun and engaging as possible😊

This series is divided into 6 parts:

* ***Understanding IPFS in Depth (1/6): A Beginner to Advanced Guide:***In this part, we will try to understand What IPFS is, why do we need it and What we can do with it. We will cover all the underlying components of IPFS in brief (which will be explained in depth in further parts) and see how they work together. ***If you want a short summary and don’t want to understand what’s happening “under the hood”, then this part is for you***😊
* ***Understanding IPFS in Depth (2/6): What is Interplanetary Linked Data (IPLD)?:***In this part, we will dive into the data model of the content-addressable web. We will explore the details and specs of IPLD and play with it to get more familiar with it. You can read it here:

**[Understanding IPFS in Depth(2/6): What is Interplanetary Linked Data(IPLD)?](https://medium.com/towardsblockchain/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b)**

[A Complete Guide to IPLD: Its Significance, How it Works and Practical Examples](https://medium.com/towardsblockchain/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b)

[medium.com](https://medium.com/towardsblockchain/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b)

* ***Understanding IPFS in Depth (3/6): What is Interplanetary Naming System (IPNS)?:***In this part, we will dive into the naming System of the distributed web. We will discuss it’s working, specs and play with it. We will also compare it to today’s naming system, aka the DNS. We will create a list of pros and cons of IPNS vs DNS. You can read it here:

**[Understanding IPFS in Depth (3/6): What is Interplanetary Naming System(IPNS)?](https://hackernoon.com/understanding-ipfs-in-depth-3-6-what-is-interplanetary-naming-system-ipns-9aca71e4c13b)**

[Why do need IPNS, How to Use it and its Comparison with DNS](https://hackernoon.com/understanding-ipfs-in-depth-3-6-what-is-interplanetary-naming-system-ipns-9aca71e4c13b)

[hackernoon.com](https://hackernoon.com/understanding-ipfs-in-depth-3-6-what-is-interplanetary-naming-system-ipns-9aca71e4c13b)

* ***Understanding IPFS in Depth (4/6): What is Multiformat?***In this part, we will talk about Why we need Multiformat, How it works and What you as a user/developer can do with it?

**[Understanding IPFS in Depth (4/6): What is Multiformat?](https://hackernoon.com/understanding-ipfs-in-depth-4-6-what-is-multiformats-cf25eef83966)**

[Every choice in computing has a tradeoff. This includes formats, algorithms, encodings, and so on. And even with a…](https://hackernoon.com/understanding-ipfs-in-depth-4-6-what-is-multiformats-cf25eef83966)

[hackernoon.com](https://hackernoon.com/understanding-ipfs-in-depth-4-6-what-is-multiformats-cf25eef83966)

* ***Understanding IPFS in Depth (5/6): What is Libp2p?***In this part, we will study the networking Layer of IPFS and what it contributes to the awesomeness of IPFS. We will go through it’s working, specs and play around with it to understand it more clearly.

**[Understanding IPFS in Depth (5/6): What is Libp2p?](https://medium.com/coinmonks/understanding-ipfs-in-depth-5-6-what-is-libp2p-f8bf7724d452)**

[What’s it’s Significance, how it works and Building a libp2p Chat Application](https://medium.com/coinmonks/understanding-ipfs-in-depth-5-6-what-is-libp2p-f8bf7724d452)

[medium.com](https://medium.com/coinmonks/understanding-ipfs-in-depth-5-6-what-is-libp2p-f8bf7724d452)

* ***Understanding IPFS in Depth (6/6): What is Filecoin?***In this part, we discuss the incentivization layer of IPFS, filecoin. We discuss its whitepaper and its implementation specs including DSN (Distributed Storage Network), Proof-of-replication, Proof of Storage, Data storage and retrieval markets and Smart contract implementation on Filecoin protocol. We also discuss some flaws in filecoin protocol which are not mentioned in the whitepaper and suggest some improvements in filecoin protocol. ***You can read this part here***:

**[Ultimate Guide to Filecoin: Breaking Down Filecoin Whitepaper & Economics](https://medium.com/swlh/ultimate-guide-to-filecoin-breaking-down-filecoin-whitepaper-economics-9212541a5895)**

[What is Filecoin, How it Works? Understanding Technical & Economic Aspects of Filecoin.](https://medium.com/swlh/ultimate-guide-to-filecoin-breaking-down-filecoin-whitepaper-economics-9212541a5895)

[medium.com](https://medium.com/swlh/ultimate-guide-to-filecoin-breaking-down-filecoin-whitepaper-economics-9212541a5895)

You can [***follow me on medium***](https://medium.com/@vaibhavsaini_67863) to get updated whenever a new part is published. You can also ***find me on twitter*** [***@vasa\_develop***](https://twitter.com/vasa_develop)

In case you want to know about similar projects, here is a list:

**[StoragePedia: An Encyclopedia of 5 Blockchain Storage Platforms](https://hackernoon.com/storagepedia-an-encyclopedia-of-5-blockchain-storage-platform-8aa13c630ace)**

[An In-depth Comparison of all major Distributed/Decentralized Storage Platforms](https://hackernoon.com/storagepedia-an-encyclopedia-of-5-blockchain-storage-platform-8aa13c630ace)

[hackernoon.com](https://hackernoon.com/storagepedia-an-encyclopedia-of-5-blockchain-storage-platform-8aa13c630ace)

If you like high-tech Web3 concepts explained in simple words with interactive tutorials, then head here.

**[Home - SimpleAsWater](https://simpleaswater.com/ipfs?ref=medium_ipfs)**

[SimpleAsWater, a Community Platform to Learn, Build, Collaborate & Discover Dapps and Web 3.0 Stories.](https://simpleaswater.com/ipfs?ref=medium_ipfs)

[simpleaswater.com](https://simpleaswater.com/ipfs?ref=medium_ipfs)

Hope you learn a lot about IPFS from this series. Let’s start!

When you ask someone about the latest Avenger’s movie, they probably aren’t going to say something like “*the one on this server, at this sub-domain, under this file path, slash Marvel dash Avengers dot mp4*”. Instead, they’re probably going to describe the content of the video: “*the one in which half of the universe is destroyed by Thanos…*”. This is obviously an intuitive way to think about content for humans but is generally *NOT* how we access content on the web today. Having said that, decentralized protocols such as [IPFS](https://ipfs.io/) actually *do* use this type of [*content addressin*](https://medium.com/textileio/enabling-the-distributed-web-abf7ab33b638)*g (labeling and finding content using the content of the file)* to find content on the decentralized web. In this article, we’ll explore a little bit how this whole process works, what are the different components involved and how they work together. We will do this by adding a file to IPFS and then exploring what happens when we add a file to IPFS.

**How Does It Work?**

Let’s start by adding a photo to IPFS. We will add this one…

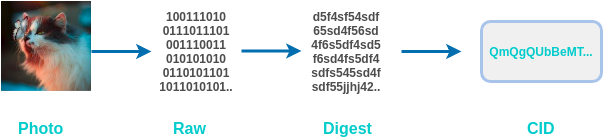
**[Instant photo by Karina Vorozheeva (@\_k\_arinn) on Unsplash](https://unsplash.com/photos/rW-I87aPY5Y)**

[Download this free HD photo of cat, butterfly, animal and insect by Karina Vorozheeva (@\_k\_arinn)](https://unsplash.com/photos/rW-I87aPY5Y)

[unsplash.com](https://unsplash.com/photos/rW-I87aPY5Y)

BTW, you will have to install IPFS on your system to follow along with me. You can install it from [here](https://docs.ipfs.io/introduction/install/). After installing IPFS, you have to start IPFS *daemon (the software which communicates with the IPFS network in order to add and retrieve data from the network).*You can start the daemon by ipfs daemon

When you add the photo to IPFS, this is what happens:



What happens when we add a photo on IPFS?

On terminal I get this:



You can see the final hash here: QmQgQUbBeMTnH1j3QWwNw9LkXjpWDJrjyGYfZpnPp8x5Lu

But we don’t see anything related to the 2 steps (*Raw*and*Digest*). This all happens under the hood.

When we added the image, we converted the image into the *Raw*data which computer can understand. Now, to make it ***content-addressable*** (the thing that we talked about above) we have to come up with a method by which we can convert this image data to a label which uniquely identifies it’s content.

This is where hash functions come into play.

Hash functions take the data (any data from texts, photos, whole bible etc.) as input and give us an output (*Digest*) which is ***unique*** with respect to its output. If we change even a pixel in this image then the output will be different. This shows its ***tamper-proof***property, hence making IPFS a ***Self-certifying File System***. So, if you transfer this image to anybody, he/she can check that if the photo received has been tampered with or not.

Also, you cannot tell what was the input (in this case, cat photo), but just seeing its output (the *Digest*). So, this also ensures a great amount of ***security***.

Now we pass the *Raw*image data into [SHA256](https://en.wikipedia.org/wiki/SHA-2) hash function and get the unique *Digest.*Now, we need to convert this Digest into a CID (Content Identifier). This CID is what IPFS will search for when we try to get back the image. For doing this, IPFS uses something called [*Multihash*](https://multiformats.io/multihash/#the-multihash-format)*.*

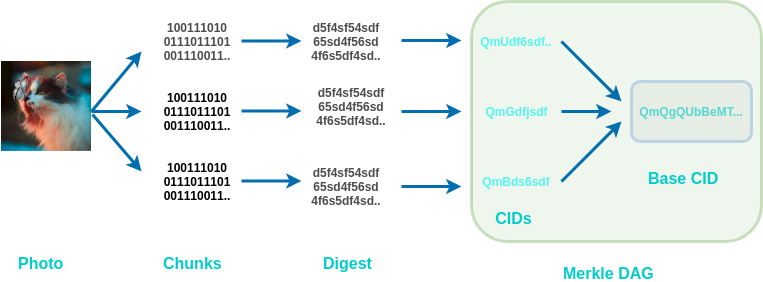
To understand the significance of *Multihash,* consider this situation.

You stored this image on the internet and you have its CID, which you can give to anybody who wants to get this image. Now, what if you discover in future that SHA256 is broken (this would mean that this process is NO more *tamper-proof and secure*) and you want to use SHA3(to ensure tamper-proofing and security) instead? This would mean changing the whole process of converting your photo to CID, and the previous CIDs will be useless…

This above problem may seem a small issue in this context, but you should know that these ***hash functions secure*** ***TRILLIONS of dollars***. All the Banks, National security agencies etc. use these hash functions to operate securely. ***Even the green lock that you see beside the address of every site on the browser will not work without hash functions***.

In order to solve this problem, IPFS uses *Multihash.*Multihash allows us to define hashes that are self-defining. So, we can have multiple version of CIDs, according to the hash function used. We will talk more about Multihashes in our Part 4 of the series, which dives deeper into ***Multiformat***.

Well, now we have added our photo to IPFS, but this was not the whole story. What is actually happening is something like this:



Large files are chunked, hashed, and organized into an IPLD (Merkle DAG object)

If the files are bigger than 256 kB, then they are broken down into smaller parts, so that all the part are equal or smaller than 256 kb. We can see the chunks of our photo using this command:

ipfs object get Qmd286K6pohQcTKYqnS1YhWrCiS4gz7Xi34sdwMe9USZ7u

This gives us 15 chunks, each of which is smaller than 256kb. Each of these chunks is first converted into a digest and then into CIDs.

{

"Links": [

{

"Name": "",

"Hash": "QmZ5RgT3jJhRNMEgLSEsez9uz1oDnNeAysLLxRco8jz5Be",

"Size": 262158

},

{

"Name": "",

"Hash": "QmUZvm5TertyZagJfoaw5E5DRvH6Ssu4Wsdfw69NHaNRTc",

"Size": 262158

},

{

"Name": "",

"Hash": "QmTA3tDxTZn5DGaDshGTu9XHon3kcRt17dgyoomwbJkxvJ",

"Size": 262158

},

{

"Name": "",

"Hash": "QmXRkS2AtimY2gujGJDXjSSkpt2Xmgog6FjtmEzt2PwcsA",

"Size": 262158

},

{

"Name": "",

"Hash": "QmVuqvYLEo76hJVE9c5h9KP2MbQuTxSFyntV22qdz6F1Dr",

"Size": 262158

},

{

"Name": "",

"Hash": "QmbsEhRqFwKAUoc6ivZyPa1vGUxFKBT4ciH79gVszPcFEG",

"Size": 262158

},

{

"Name": "",

"Hash": "QmegS44oDgNU2hnD3j8r1WH8xZ2RWfe3Z5eb6aJRHXwJsw",

"Size": 262158

},

{

"Name": "",

"Hash": "QmbC1ZyGUoxZrmTTjgmiB3KSRRXJFkhpnyKYkiVC6PUMzf",

"Size": 262158

},

{

"Name": "",

"Hash": "QmZvpEyzP7C8BABesRvpYWPec2HGuzgnTg4VSPiTpQWGpy",

"Size": 262158

},

{

"Name": "",

"Hash": "QmZhzU2QJF4rUpRSWZxjutWz22CpFELmcNXkGAB1GVb26H",

"Size": 262158

},

{

"Name": "",

"Hash": "QmZeXvgS1NTxtVv9AeHMpA9oGCRrnVTa9bSCSDgAt52iyT",

"Size": 262158

},

{

"Name": "",

"Hash": "QmPy1wpe1mACVrXRBtyxriT2T5AffZ1SUkE7xxnAHo4Dvs",

"Size": 262158

},

{

"Name": "",

"Hash": "QmcHbhgwAVddCyFVigt2DLSg8FGaQ1GLqkyy5M3U5DvTc6",

"Size": 262158

},

{

"Name": "",

"Hash": "QmNsx32qEiEcHRL1TFcy2bPvwqjHZGp62mbcVa9FUpY9Z5",

"Size": 262158

},

{

"Name": "",

"Hash": "QmVx2NfXEvHaS8uaRTYaF4ExeLaCSGpTSDhhYBEAembdbk",

"Size": 69716

}

],

"Data": "\b\u0002\u0018Ơ�\u0001 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 ��\u0010 Ơ\u0004"

}

IPFS uses IPLD (IPLD uses [Merkle DAG](https://en.wikipedia.org/wiki/Merkle_tree), or [directed acyclic graph](https://en.wikipedia.org/wiki/Directed_acyclic_graph)) for managing all the chunks and linking it to the base CID.

IPLD (objects) consist of 2 components:

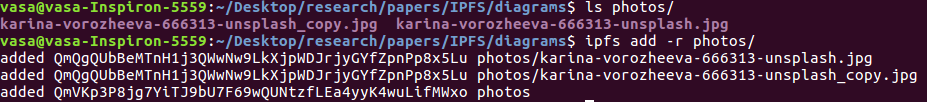
* Data — blob of unstructured binary data of size < 256 kB.
* Links — array of Link structures. These are links to other IPFS objects.

Every IPLD Link(in our case the 15 links that we got above) has 3 parts:

* Name — name of the Link
* Hash — hash of the linked IPFS object
* Size — the cumulative size of linked IPFS object, including following *its* links

IPLD is built on the ideas of Linked Data, which is actually something that folks in the decentralized web community have been talking about for quite some time. It’s something Tim Berners-Lee has been working on for ages, and his new company, [Solid](https://solid.mit.edu/), is building a business around it.

Using IPLD has also other benefits. To see this, let's create a folder named photos and add 2 photos into it (the cat pic and a copy of the same pic).



As you can see both the photos have the same hash(which proves that I haven’t changed anything in the copy of the image). This adds ***De-duplication***property to IPFS. So even if your friend adds the same cat photo to IPFS, he will not duplicate the image. This saves a lot of storage space.

Imagine, if I store this article on IPFS, and it’s each letter in chunked and has a unique CID, then this whole article can be constructed by just a combination of alphabets(capital and small), number and some special characters. ***We will only store every alphabet, number, and character only ONCE*** and rearrange it according to the Links in the data structure. This is powerful stuff…

IPFS also has a ***naming system called InterPlanetary Naming System(IPNS)***. To understand it’s significance let’s imagine that you have created a website and you hosted it on some domain. For this example, we will take my website: [***https://vaibhavsaini.com***](https://vaibhavsaini.com/)

If I want to host it on IPFS, I will simply add the website folder on IPFS. For that I have downloaded the website using wget . In case you are on Linux based OS like Ubuntu or MAC, then you can try it with me.

Download the website(or any website):

wget --mirror --convert-links --adjust-extension --page-requisites --no-parent [https://vaibhavsaini.com](https://vaibhavsaini.com/)

Now add the folder named vaibhavsaini.com to IPFS:

ipfs add -r vaibhavsaini.com

You will get something like this:



As we can see, our website is now hosted on the last CID(which is the CID for the folder):

QmYVd8qstdXtTd1quwv4nJen6XprykxQRLo67Jy7WyiLMB

We can access the site using ***http***protocol:

<https://gateway.pinata.cloud/ipfs/QmYVd8qstdXtTd1quwv4nJen6XprykxQRLo67Jy7WyiLMB/>

Let’s suppose if I want to change my profile image on the website. As we have learned above, if we change the content of the input, we get a different digest, which will mean that my final “Base CID” will be different.

This means that I will have to update the hash every time I update my site. Everybody who has the link to my previous website(the above URL) will NOT be able to see my new website 😞

This can result in a great problem.

To address this problem, IPFS uses InterPlanetary Naming System(IPNS). Using IPNS links point to CIDs. If I want to update my website CID, I will just point a new CID to the corresponding IPNS link(***this is analogous to today's DNS***). We will explore IPNS in depth in Part 3 of this series.

But for now, let’s generate an IPNS link for my website.

ipfs name publish QmYVd8qstdXtTd1quwv4nJen6XprykxQRLo67Jy7WyiLMB

This can take up to a few minutes. You will get an output like this:

Published to Qmb1VVr5xjpXHCTcVm3KF3i88GLFXSetjcxL7PQJRviXSy: /ipfs/QmYVd8qstdXtTd1quwv4nJen6XprykxQRLo67Jy7WyiLMB

Now, if I want to add an updated CID, I will just use the same command

ipfs name publish <my\_new\_CID>

Using this, you can access updated versions of my website, using this link:

<https://gateway.pinata.cloud/ipns/Qmb1VVr5xjpXHCTcVm3KF3i88GLFXSetjcxL7PQJRviXSy>

The above link is still not human-readable. We are used to names like: [***https://vaibhavsaini.com***](https://vaibhavsaini.com/)***.***In Part 3 of the series, we will see how we can link IPNS links to domain names, so you can see my IPFS hosted website on [***https://vaibhavsaini.com***](https://vaibhavsaini.com/)***.***

***IPFS is also a potential replacement to the HTTP protocol***. But why do we want to replace HTTP? It seems to work fine, right? I mean you are able to read this article and see movies on Netflix, all using HTTP protocol.

Even if it seems to work fine to us, it has a few big problems.

Let’s say you are sitting in a lecture hall, and your professor asks you to go to a specific website. Every student in the lecture makes a request to that website and are given a response. This means that the same exact data was sent individually to each student in the room. If there are 100 students, then that’s 100 requests and 100 responses. This is obviously not the most efficient way to do things. Ideally, the students will be able to leverage their physical proximity to more efficiently retrieve the information they need.

HTTP also presents a big problem if there is some problem in the networks line of communication and the client is unable to connect with the server. This can happen if an ISP has an outage, a country is blocking some content, or if the content was simply deleted or moved. These types of [broken links](https://en.wikipedia.org/wiki/HTTP_404) exist everywhere on the HTTP web.

The location-based addressing model of HTTP encourages centralization. It’s convenient to trust a handful of applications with all our data but because of this much of the data on the web becomes soiled. This leaves those providers with enormous responsibility and power over our information.

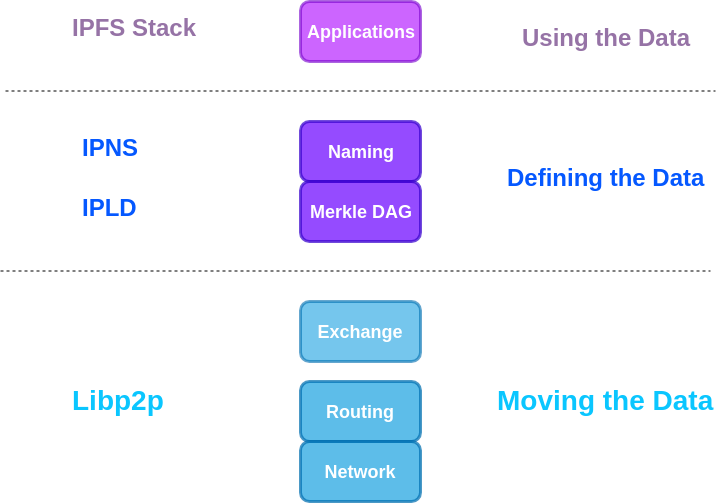
Here is where ***Libp2p*** comes in. Libp2p is used to communicate data and discovering other peers (computers and smartphones) on the IPFS network. The way this works is that if every computer and smartphone is running IPFS software then we will be part of a big BitTorrent like a network where every system can act as a client and also a server. So, if 100 students are requesting the same website, they can request the website data from each other. This kind of system, if implemented on large scale, can result in ***significant improvement in speed at which surf the Internet***.

Ok, let’s stop here. If you have made it till here then you deserve a pat on your back. Well done😎

Till now, We have learned a lot of things about IPFS. Let’s recap:

* IPFS is ***content-addressable.***The data on IPFS is identified using CIDs.
* These ***CIDs are unique***with respect to the data referenced by it.
* IPFS uses hash functions for its ***tamper-proof***property which makes IPFS a ***Self-certifying File System***.
* IPFS uses ***Multihash,*** which allows it to ***have different versions of CIDs for the same data***(*this doesn’t mean that CIDs are not unique. If we use the same hash function, then we will have same CID. We will talk more about this in Part 4 of this series*).
* IPFS uses ***IPLD to manage and link all the chunks of data*.**
* ***IPLD uses Merkle DAG***(aka [**D**irected **A**cyclic **G**raph](https://en.wikipedia.org/wiki/Directed_acyclic_graph)) data structure to link the chunks of data.
* IPLD also adds ***De-duplication*** property to IPFS.
* IPFS uses ***IPNS for linking CIDs to a fixed IPNS link***, which is analogous to today DNS of the centralized Internet.
* IPFS uses***Libp2p to communicate data and discovering other peers(computers and smartphones) on the IPFS network*** which can significantly improve the speed of your net surfing.

Below is a diagrammatic representation of IPFS stack.



IPFS Stack

I hope that you have learned a lot. If you want to jump to the next part, head here:

**[Understanding IPFS in Depth(2/6): What is InterPlanetary Linked Data(IPLD)?](https://medium.com/towardsblockchain/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b)**

[A Complete Guide to IPLD: Its Significance, How it Works and Practical Examples](https://medium.com/towardsblockchain/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b)

[medium.com](https://medium.com/towardsblockchain/understanding-ipfs-in-depth-2-6-what-is-interplanetary-linked-data-ipld-c8c01551517b)

I would like to thank [Carson Farmer](https://medium.com/u/adda52ceb6bc?source=post_page-----e937675a8c8a----------------------) for writing such an [incredible article](https://medium.com/textileio/whats-really-happening-when-you-add-a-file-to-ipfs-ae3b8b5e4b0f) on IPFS, from which this article takes a ton of Inspiration. You should also try out their [Textile Photo App](https://www.textile.photos/).

Thanks for reading ;)

**Learned something? Press and hold the 👏 to say “thanks!” and help others find this article.**

*Hold down the clap button if you liked the content! It helps me gain exposure.*



**About the Author**

[Vaibhav Saini](http://vaibhavsaini.com/) is a Co-Founder of ***[TowardsBlockchain](http://towardsblockchain.com/" \t "_blank),****an MIT Cambridge Innovation Center incubated startup.*

He works as Senior blockchain developer and has worked on several blockchain platforms including Ethereum, Quorum, EOS, Nano, Hashgraph, IOTA etc.

He is a Speaker, Writer and a drop-out from [IIT Delhi](http://www.iitd.ac.in/).