IPFS File System

Interplanetary File System

IPFS File System

Interplanetary File System

- Disclaimer -

Not responsible for any damage done to you, your friends, your accounts, your pet goldfish, etc. All information is for educational or general knowledge purposes.
Information held within may or may not be legal by your country, state or business.

If it's not legal then should you do it?

OtakuSystems LLc

Pat Baker

info@otakusystems.com

Twitter: @otakusystems

Current file access issues

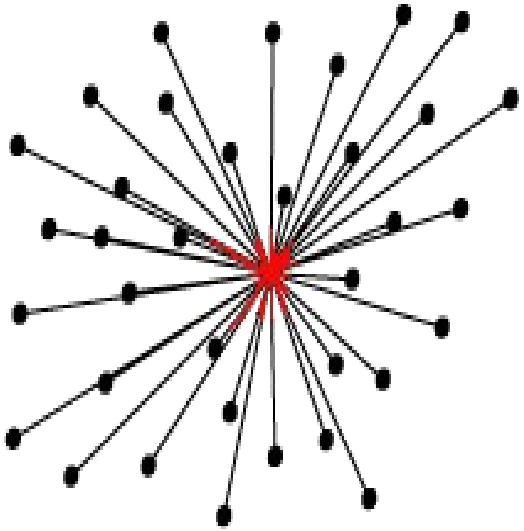
- Location based (URLs Uniform Resource Locations) access to file, think https://www.puppies.com/beagle.jpg.
- If site is off line, data may not be access able.
- No easy way to truly verify if data has been altered or tampered with.

- Country's can censor your access to the sitelinks or sites themselves.

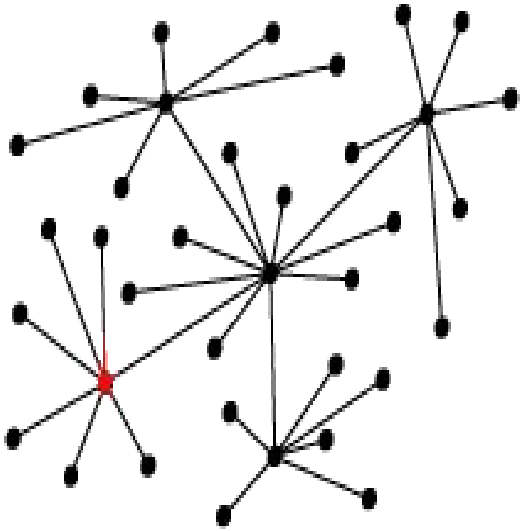
- If site vanishes the data that the site kept may be lost forever, unless stored on sites like ‘Internet Archives, etc.’

- Can be expensive to replicate the sites, may need multiple virtual or physical machines.

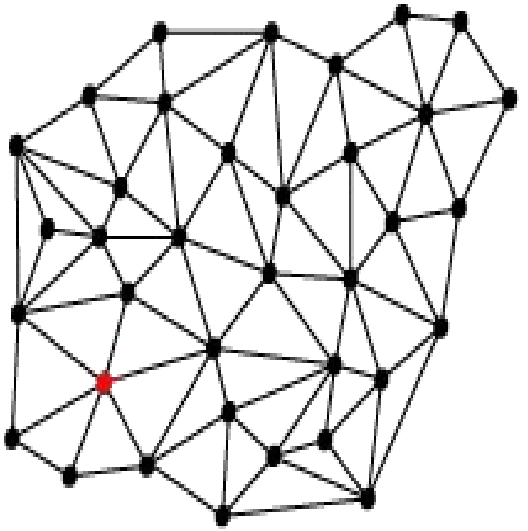
Central vs Decentralized vs distributed



centralised



decentralised



distributed

IPFS addresses the centralized issue.

What is IPFS

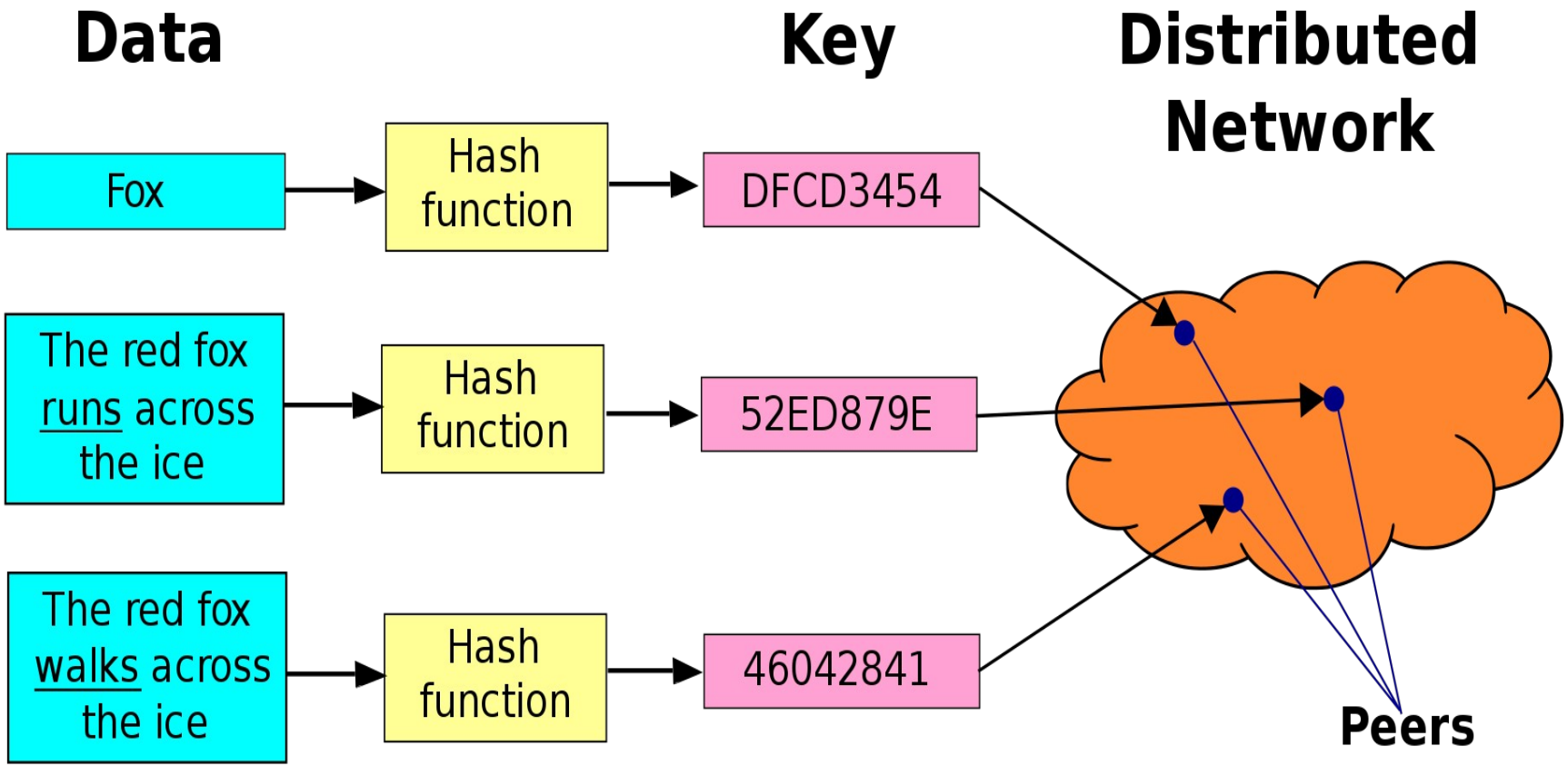
- IPFS is a decentralized storage and delivery network based on fundamental principles of P2P networking and content-based addressing.

- Content based addressing – use of based ‘Cryptographic hashing’ based link for file </ipfs/3EGYtrfgbfrgrvdFDtge564wefVww4gsfggg45gbnteFgtedRRwfvAwrg

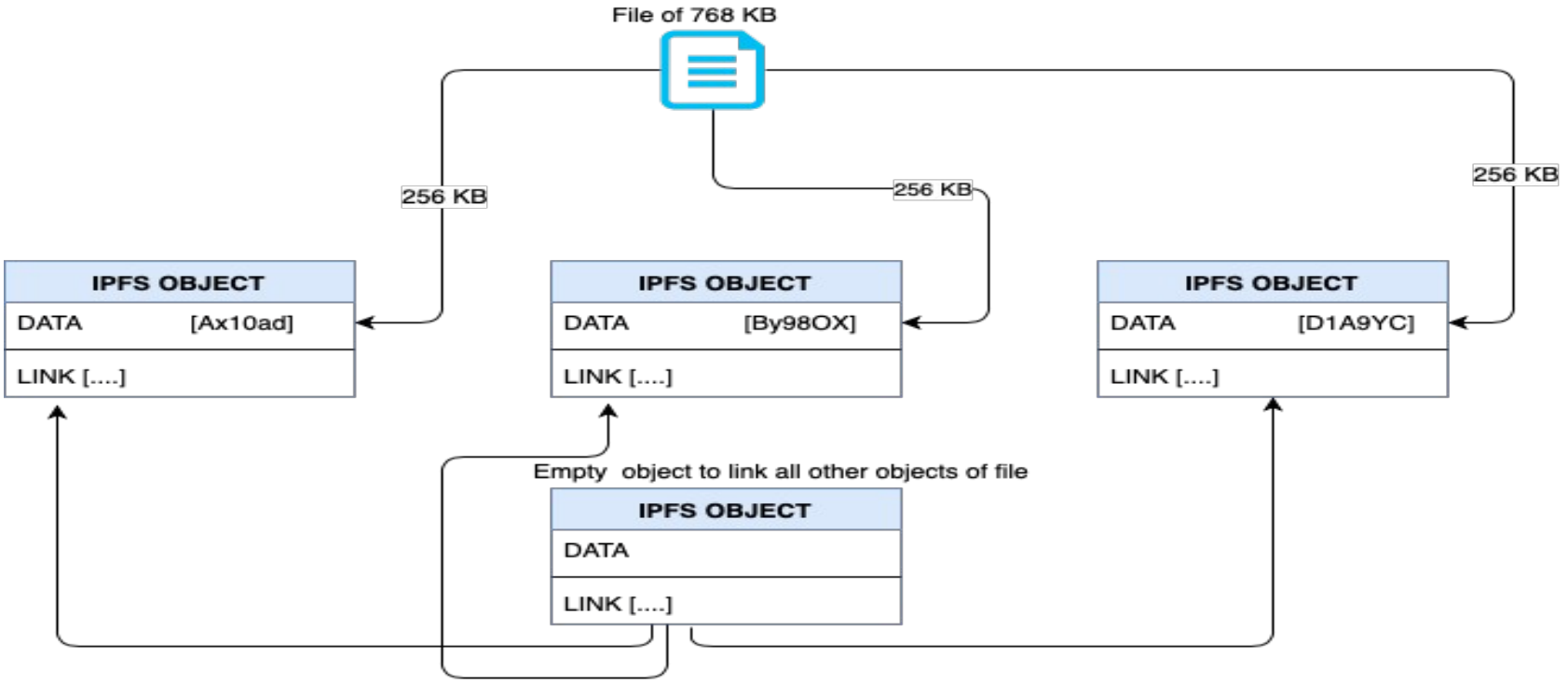
- Cryptographic hashes can be derived from the content of the data itself, meaning that anyone using the same algorithm on the same data will arrive at the same hash. If Ada and Grace are both using the same decentralized web protocol, such as IPFS, to share the exact same photo of a kitten, both images will have exactly the same hash. By comparing those hashes and confirming that they're the same, we can guarantee that every single pixel of those two photos is identical.

- Cryptographic hashes are unique. If Grace uses Photoshop to remove a single whisker from that kitty, the updated image will have a new hash. Simply by looking at that hash, even without access to the file itself, it will be easy to tell that the file now contains different data.

- Decoding data structures - A CID (Content Identifier) is a particular form of content addressing used on the decentralized web. It was developed for IPFS (a decentralized web protocol which we'll discuss in later tutorials), but has very broad implications. A CID is a single identifier that contains both a cryptographic hash and a codec, which holds information about how to interpret that data. Codecs encode and decode data in certain formats. Many formats and protocols use content addressing already. Tools like Git and protocols like Ethereum and Bitcoin are among them, but they differ in how to interpret the data and in what cryptographic function they use for hashing. CID allows us to create a universal identifier for any of these systems.


Data
Key
Distributed Network
Fox
Hash function
DFCD3454
The red fox runs across the ice
Hash function
52ED879E
The red fox walks across the ice
Hash function
46042841
Peers

File Sizes (IPFS objects) are broken into 256 Kb chunks that are linked to other chunks of the objects.



What IPFS can help solve

- IPFS allows users to host and receive content in a manner similar to BitTorrent. ... Any user in the network can serve a file by its content address, and other peers in the network can find and request that content from any node who has it using a distributed hash table (DHT).

- Having servers physically closer to the users ensures lower latency, enabling load-balancing, and allows you to scale the availability of content with demand. The IPFS network is a CDN by design because each node will cache what they consume and serve that data to its peers.

- IPFS helps to resolve congestion and overly controlling governments by distribution. Instead of locations, IPFS addresses point directly to the resources and it makes sure that this data comes from the closest sources.

- IPFS is a file sharing system that can be leveraged to more efficiently store and share large files. It relies on cryptographic hashes that can easily be stored on a blockchain.

- The system is a peer-to-peer file-sharing network.

- Files are broken into 256KB chunks for speed of retrieval and storage. Each chunk is linked to each other chunk using reference object links.

- IPFS objects can be linked to BlockChain objects and depending on the size of the file, be stored on the blockchain directly.

Parts of IPFS

- IPFS Daemon
- Merkle-dag tree -Distributed Hash Tables
- Nodes
- CID – Content Identifier
- Pinning of content
- IPNS (interplanetary naming system) - ipfs.io/ipns/QmeQe5FTgMs8PNspzTQ3LRz1iMhdq9K34TQnsCP2jqt8wV/
- IPFS Gateway

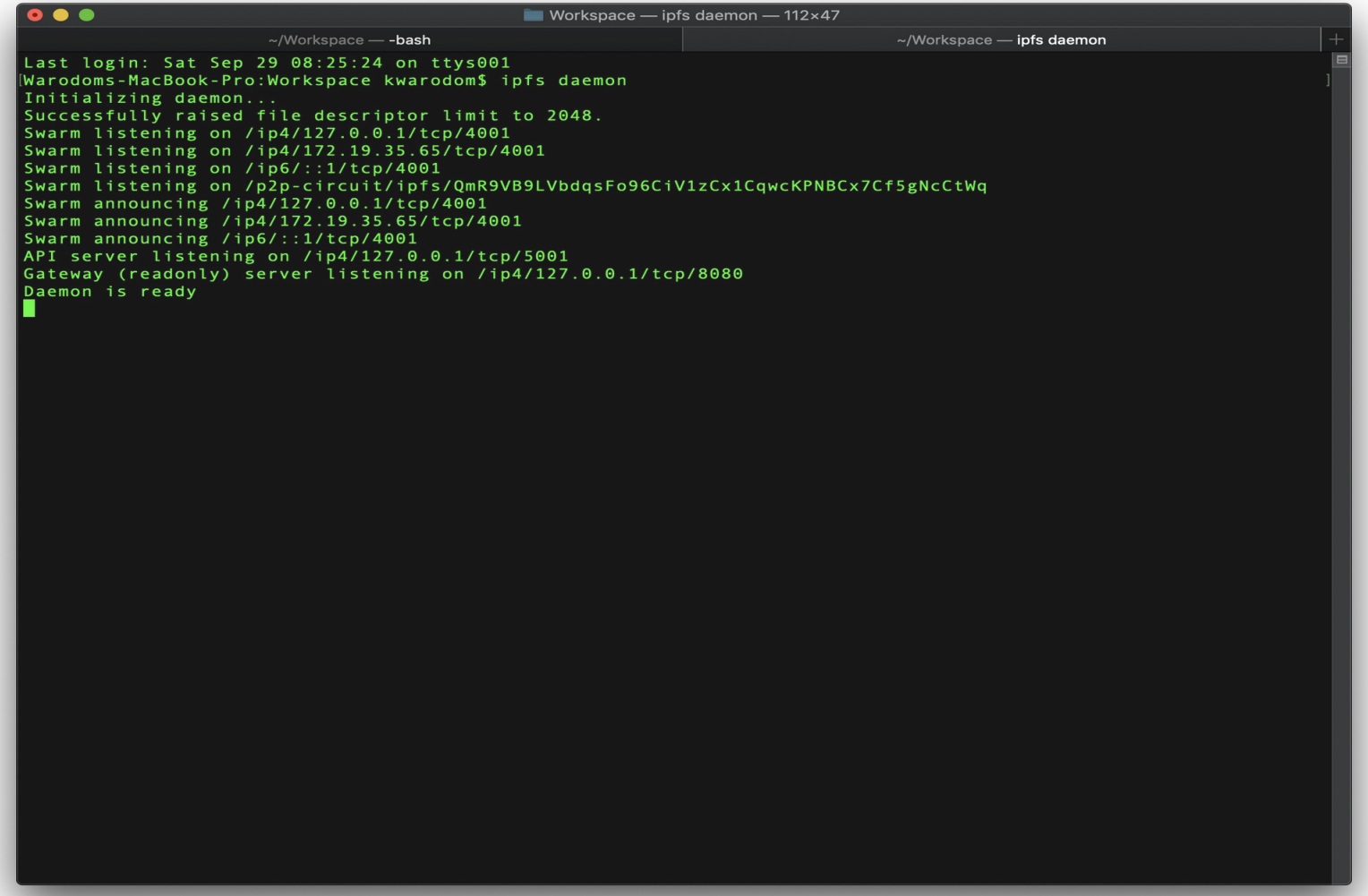
IPFS Daemon

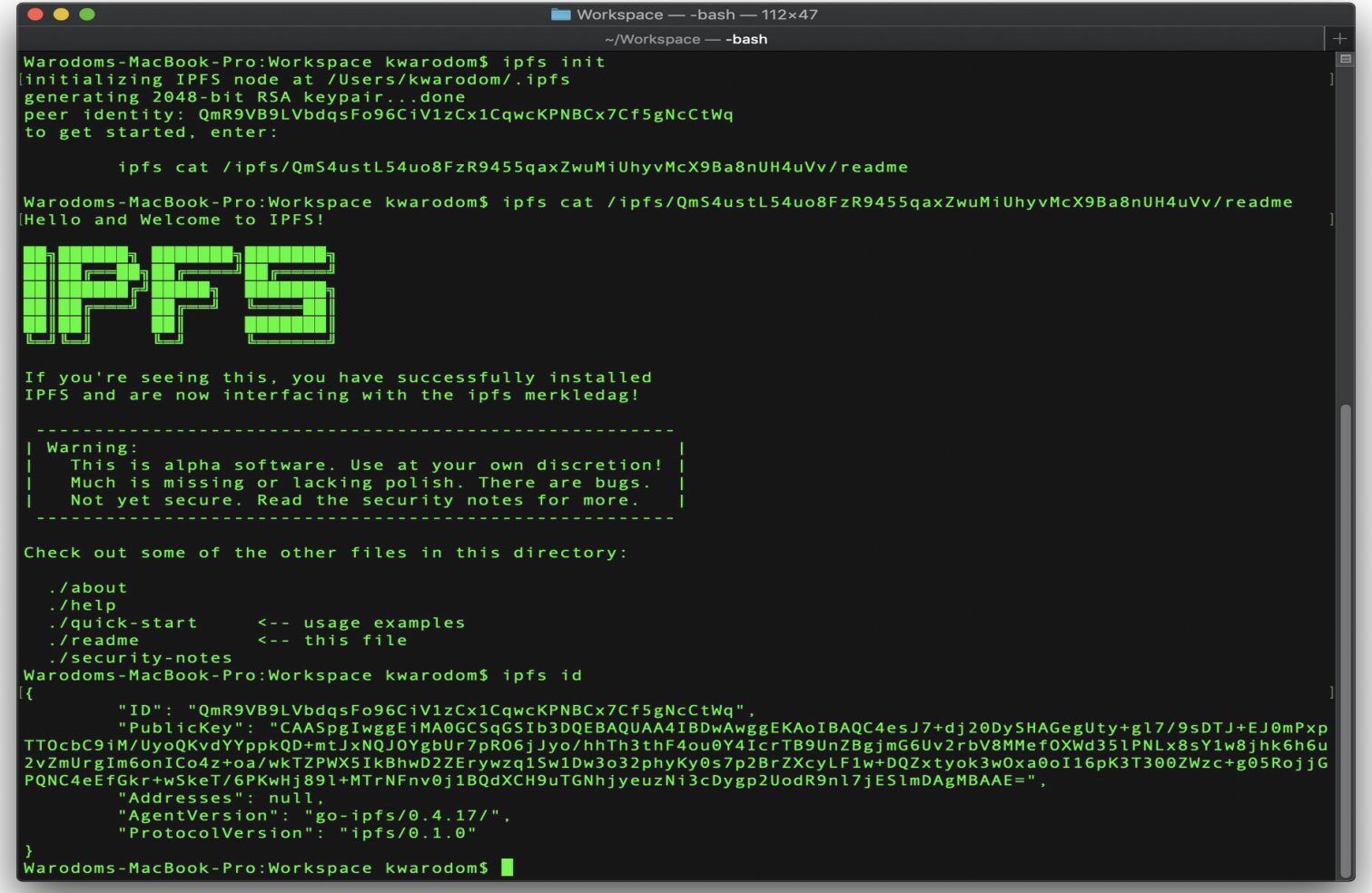
- Ipfs daemon – command to start IPFS system.
- Ipfs command by self will display commands available to you.
- You can run many web GUI commands displays at the command line.

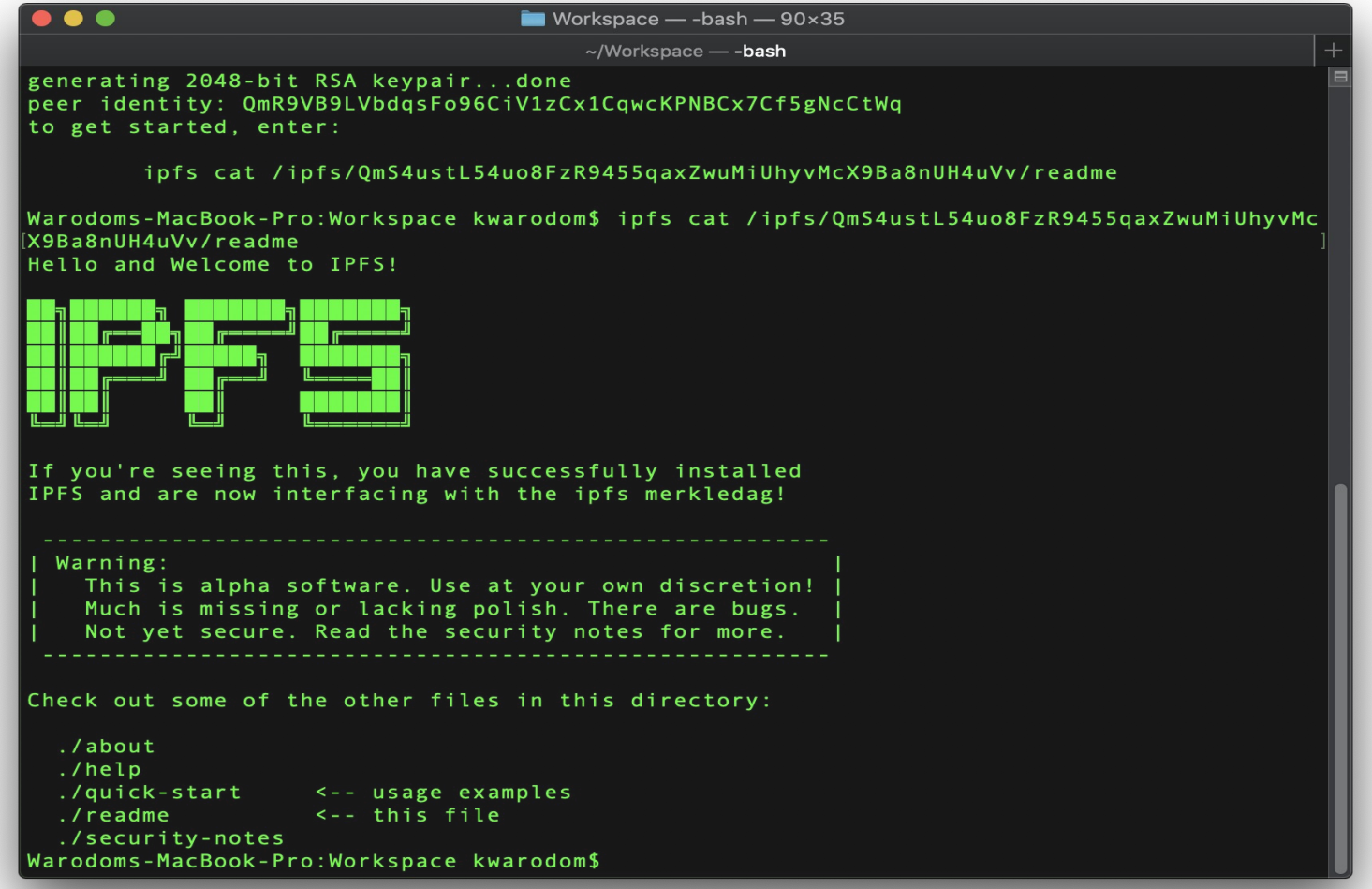
Some IPFS commands

- Ipfs – by self will show help list of commands.
- Ipfs init – Initialize local IPFS configuration.
- Ipfs daemon – starts the ipfs daemon on node.
- Ipfs add <file’s> - Adds a file or files to IPFS.
- Ipfs ls <hash> - Lists a link or links usins hash.

- Ipfs stats – Show various operational stats.
- Ipfs swarm – manage and show connections.
- Ipfs config – show and manage configurations.
- Ipfs update – inline update of go-ipfs client.
- Ipfs log – show logs.


Workspace — ipfs daemon — 112×47
~/Workspace — -bash
~/Workspace — ipfs daemon
Last login: Sat Sep 29 08:25:24 on ttys001
Warodoms-MacBook-Pro:Workspace kwarodom$ ipfs daemon
Initializing daemon...
Successfully raised file descriptor limit to 2048.
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/172.19.35.65/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit/ipfs/QmR9VB9LVbdqsFo96CiV1zCx1CqwcKPNBCx7Cf5gNcCtWq
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/172.19.35.65/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready


Workspace — -bash — 112×47
~/Workspace — -bash
Warodoms-MacBook-Pro:Workspace kwarodom$ ipfs init
initializing IPFS node at /Users/kwarodom/.ipfs
generating 2048-bit RSA keypair...done
peer identity: QmR9VB9LVbdqsFo96CiV1zCx1CqwcKPNBCx7Cf5gNcCtWq
to get started, enter:
ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv/readme
Warodoms-MacBook-Pro:Workspace kwarodom$ ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv/readme
Hello and Welcome to IPFS!
If you're seeing this, you have successfully installed
IPFS and are now interfacing with the ipfs merkledag!
Warning:
This is alpha software. Use at your own discretion!
Much is missing or lacking polish. There are bugs.
Not yet secure. Read the security notes for more.
Check out some of the other files in this directory:
./about
./help
./quick-start <-- usage examples
./readme <-- this file
./security-notes
Warodoms-MacBook-Pro:Workspace kwarodom$ ipfs id
{
"ID": "QmR9VB9LVbdqsFo96CiV1zCx1CqwcKPNBCx7Cf5gNcCtWq",
"PublicKey": "CAASpgIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC4esJ7+dj20DySHAGegUty+gl7/9sDTJ+EJ0mPxp
TTOcbC9iM/UyoQKvdYYppkQD+mtJxNQJOYgbUr7pRO6jJyo/hhTh3thF4ou0Y4IcrTB9UnZBgjmG6Uv2rbV8MMefOXWd35lPNLx8sY1w8jhk6h6u
2vZmUrgIm6onICo4z+oa/wkTZPWX5IkBhwD2ZErywzq1Sw1Dw3o32phyKy0s7p2BrZXcyLF1w+DQZxtyok3wOxa0oI16pK3T300ZWzc+g05RojjG
PQNC4eEfGkr+wSkeT/6PKwHj89l+MTrNFnv0j1BQdXCH9uTGNhjyeuzNi3cDygp2UodR9nl7jESlmDAgMBAAE=",
"Addresses": null,
"AgentVersion": "go-ipfs/0.4.17/",
"ProtocolVersion": "ipfs/0.1.0"
}
Warodoms-MacBook-Pro:Workspace kwarodom$


Workspace — -bash — 90×35
~/Workspace — -bash
generating 2048-bit RSA keypair...done
peer identity: QmR9VB9LVbdqsFo96CiV1zCx1CqwcKPNBCx7Cf5gNcCtWq
to get started, enter:
ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuMiUhyvMcX9Ba8nUH4uVv/readme
Warodoms-MacBook-Pro:Workspace kwarodom$ ipfs cat /ipfs/QmS4ustL54uo8FzR9455qaxZwuMiUhyvMc
X9Ba8nUH4uVv/readme
Hello and Welcome to IPFS!
IPFS
If you're seeing this, you have successfully installed
IPFS and are now interfacing with the ipfs merkledag!
Warning:
This is alpha software. Use at your own discretion!
Much is missing or lacking polish. There are bugs.
Not yet secure. Read the security notes for more.
Check out some of the other files in this directory:
./about
./help
./quick-start <-- usage examples
./readme <-- this file
./security-notes
Warodoms-MacBook-Pro:Workspace kwarodom$

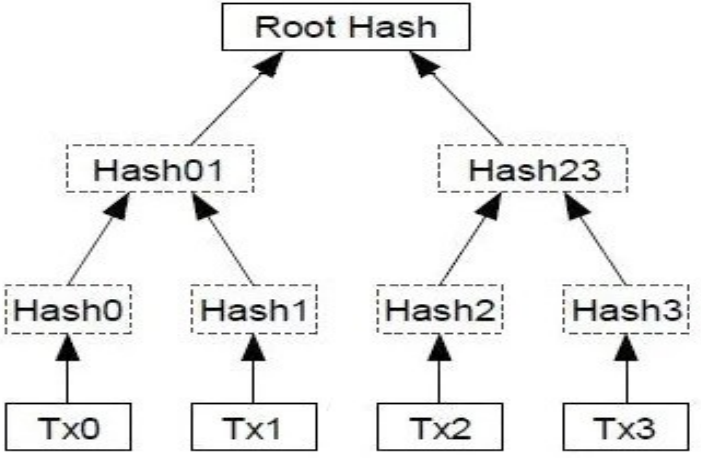
Merkle-dag tree -Distributed Hash Tables

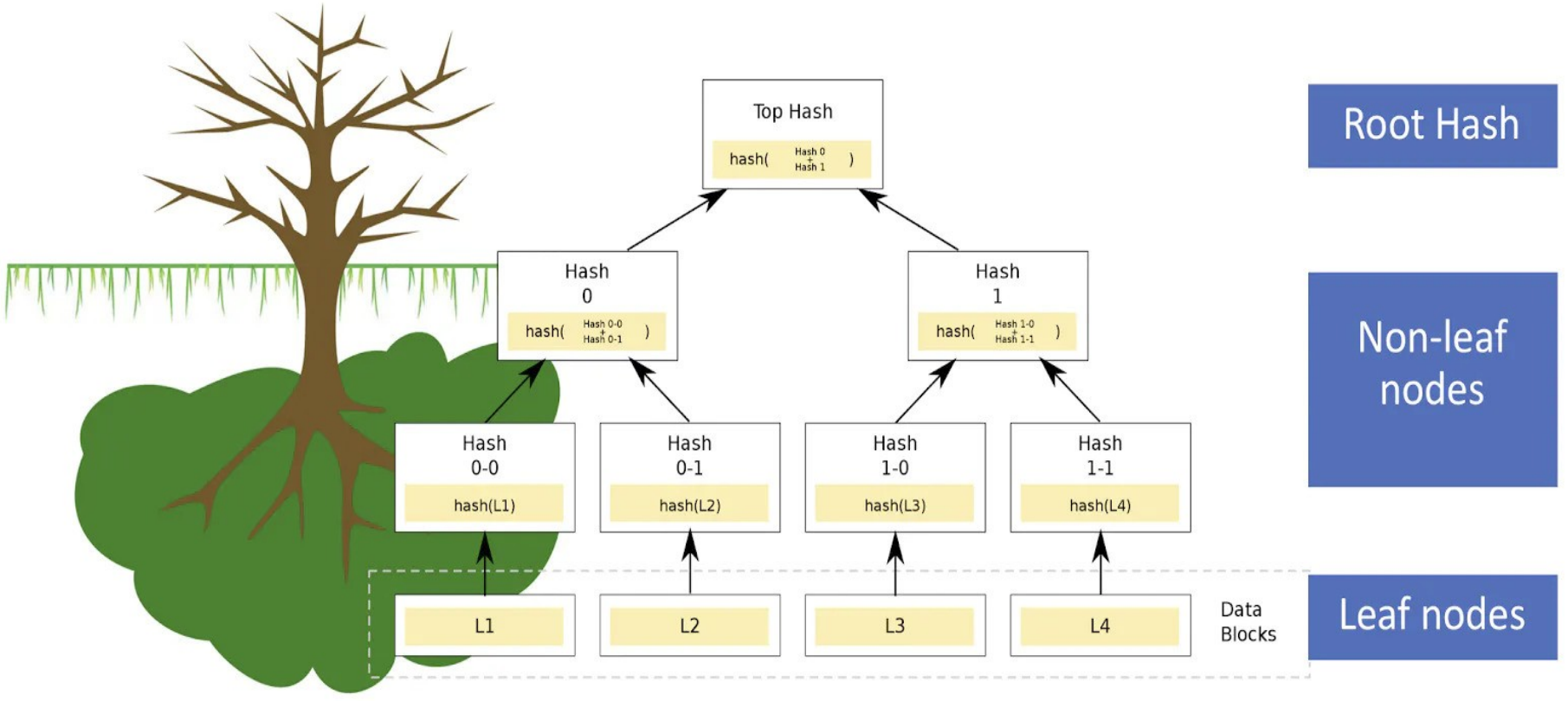
- A Merkle DAG is a DAG where each node has an identifier, and this is the result of hashing the node's contents — any opaque payload carried by the node and the list of identifiers of its children — using a cryptographic hash function like SHA256. This brings some important considerations: (Directed Acyclic Graphs)

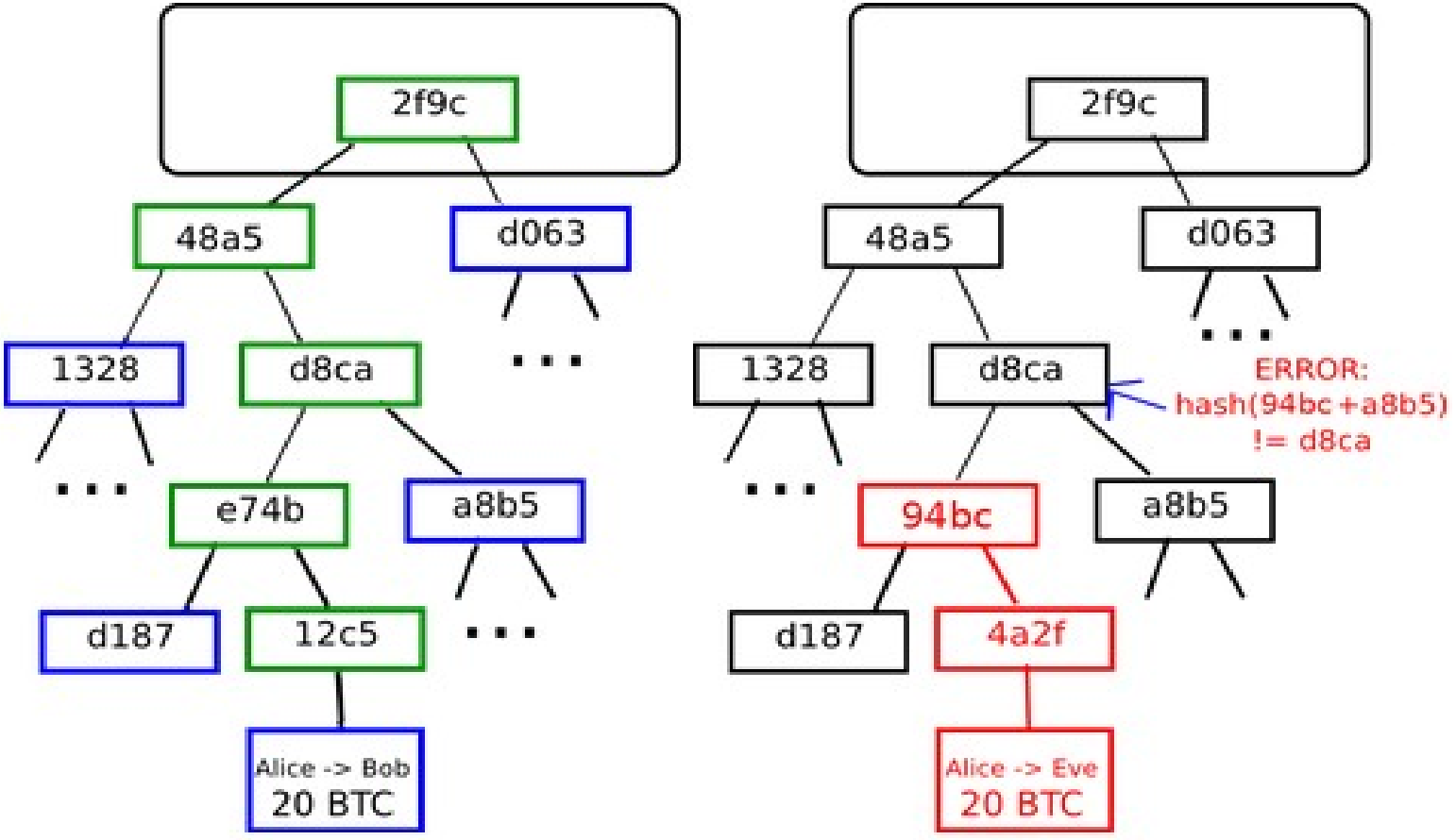
  - Merkle DAGs can only be constructed from the leaves, that is, from nodes without children. Parents are added after children because the children's identifiers must be computed in advance to be able to link them.
  - Every node in a Merkle DAG is the root of a (sub)Merkle DAG itself, and this subgraph is contained in the parent DAG.
  - Merkle DAG nodes are immutable. Any change in a node would alter its identifier and thus affect all the ascendants in the DAG, essentially creating a different DAG.

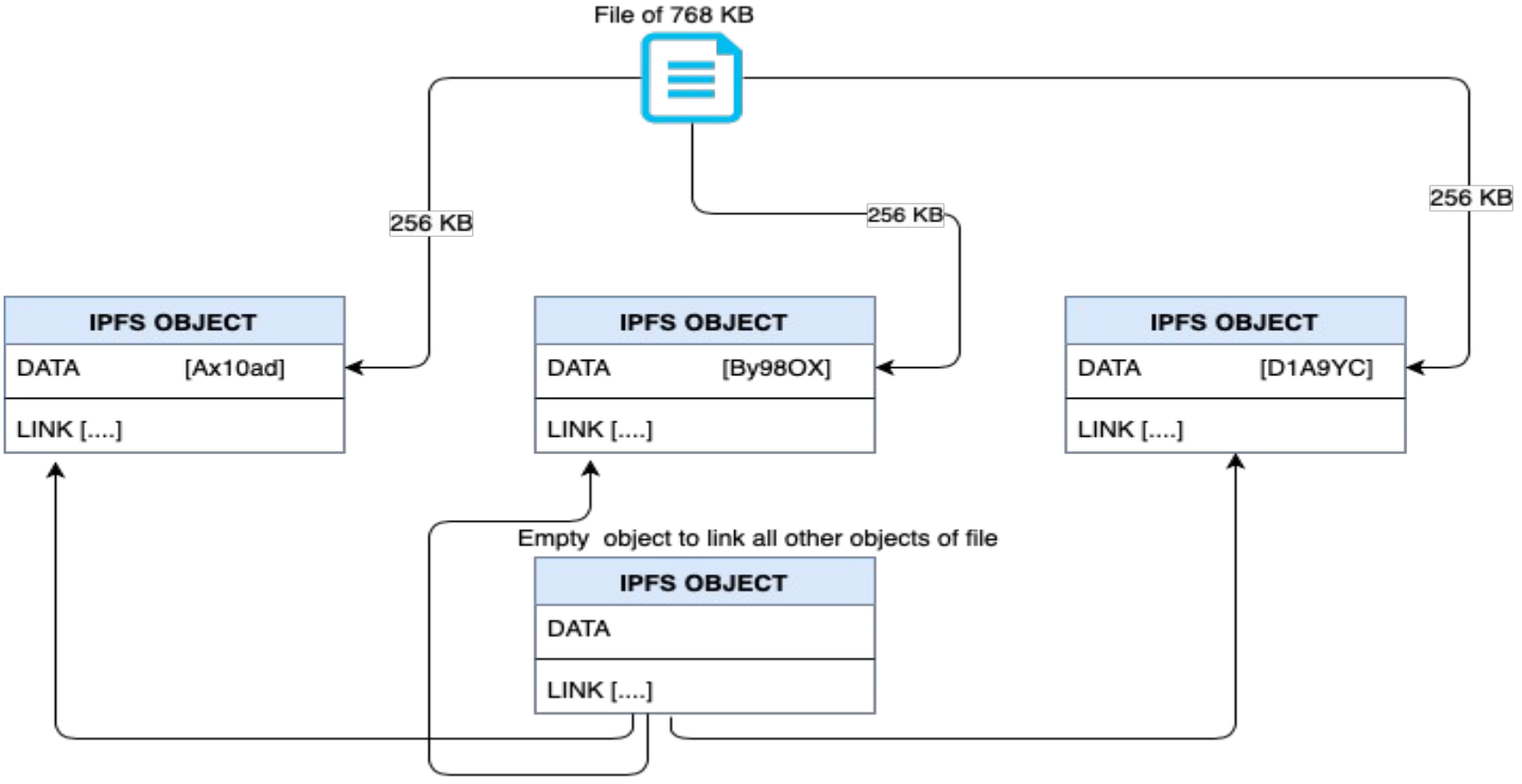

Merkel's Tree

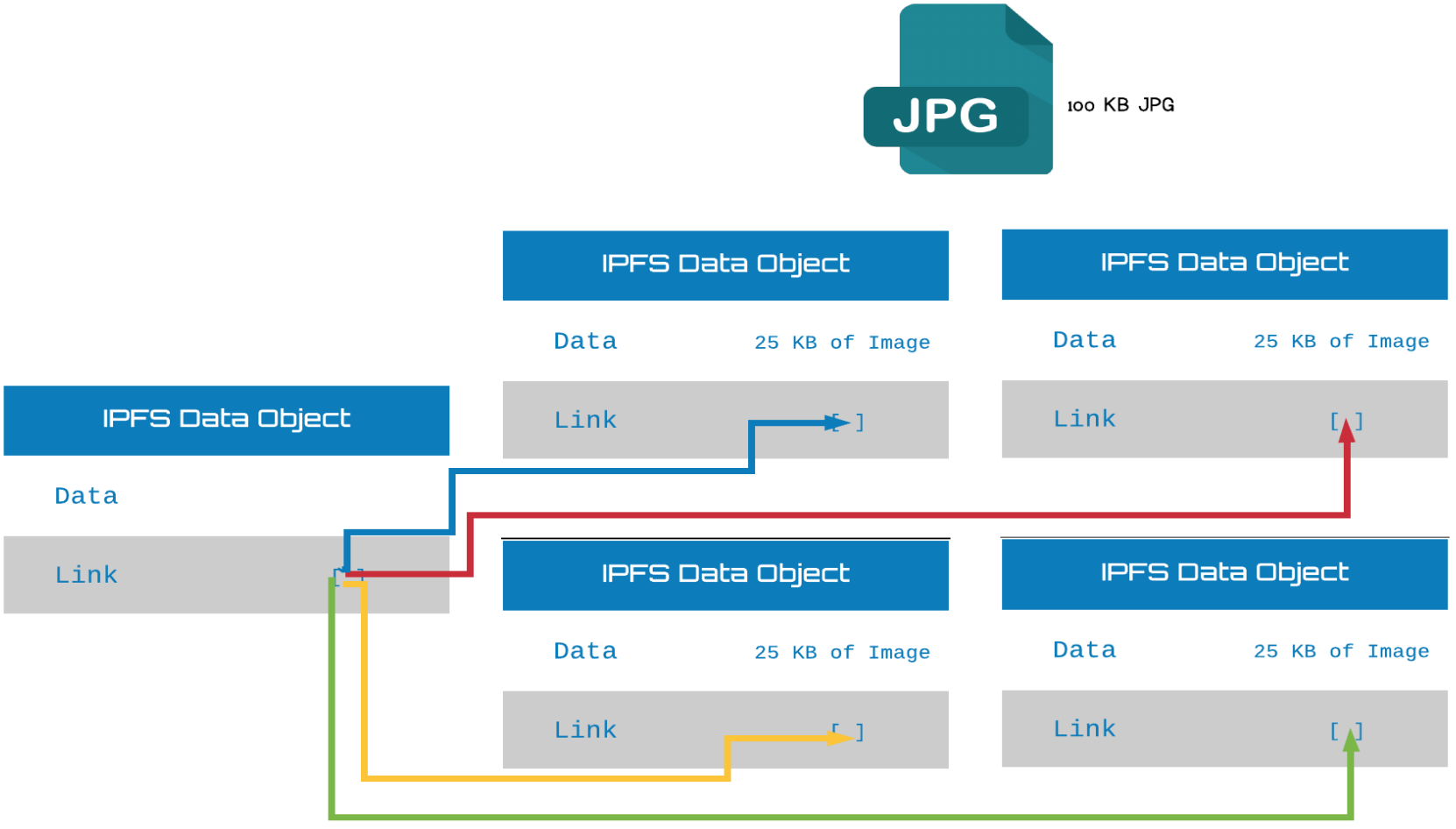
Merkle Tree


Root Hash
Hash01
Hash23
Hash0
Hash1
Hash2
Hash3
Tx0
Tx1
Tx2
Tx3


Top Hash
hash(Hash 0 + Hash 1)
Hash 0
hash(Hash 0-0 + Hash 0-1)
Hash 1
hash(Hash 1-0 + Hash 1-1)
Hash 0-0
hash(L1)
Hash 0-1
hash(L2)
Hash 1-0
hash(L3)
Hash 1-1
hash(L4)
L1
L2
L3
L4
Data Blocks
Root Hash
Non-leaf nodes
Leaf nodes


2f9c
48a5
d063
1328
d8ca
e74b
a8b5
d187
12c5
Alice -> Bob
20 BTC
2f9c
48a5
d063
1328
d8ca
ERROR:
hash(94bc +a8b5)
!= d8ca
94bc
a8b5
d187
4a2f
Alice -> Eve
20 BTC


File of 768 KB
256 KB
256 KB
256 KB
IPFS OBJECT
DATA [Ax10ad]
LINK [....]
IPFS OBJECT
DATA [By98OX]
LINK [....]
IPFS OBJECT
DATA [D1A9YC]
LINK [....]
Empty object to link all other objects of file
IPFS OBJECT
DATA
LINK [....]


JPG
100 KB JPG
IPFS Data Object
Data
25 KB of Image
Link
IPFS Data Object
Data
25 KB of Image
Link
IPFS Data Object
Data
Link
IPFS Data Object
Data
25 KB of Image
Link
IPFS Data Object
Data
25 KB of Image
Link

Node

- Participants in the IPFS network are called nodes. Nodes are the most crucial aspect of IPFS - without nodes running the IPFS daemon, there would be no IPFS Network.

- Go-IPFS - The Go implementation is designed to run on servers and user machines with the full capabilities of IPFS. New IPFS features are usually created on Go-IPFS before any other implementation.

- JS-IPFS - The Javascript implementation is designed to run in the browser with a limited set of capabilities.

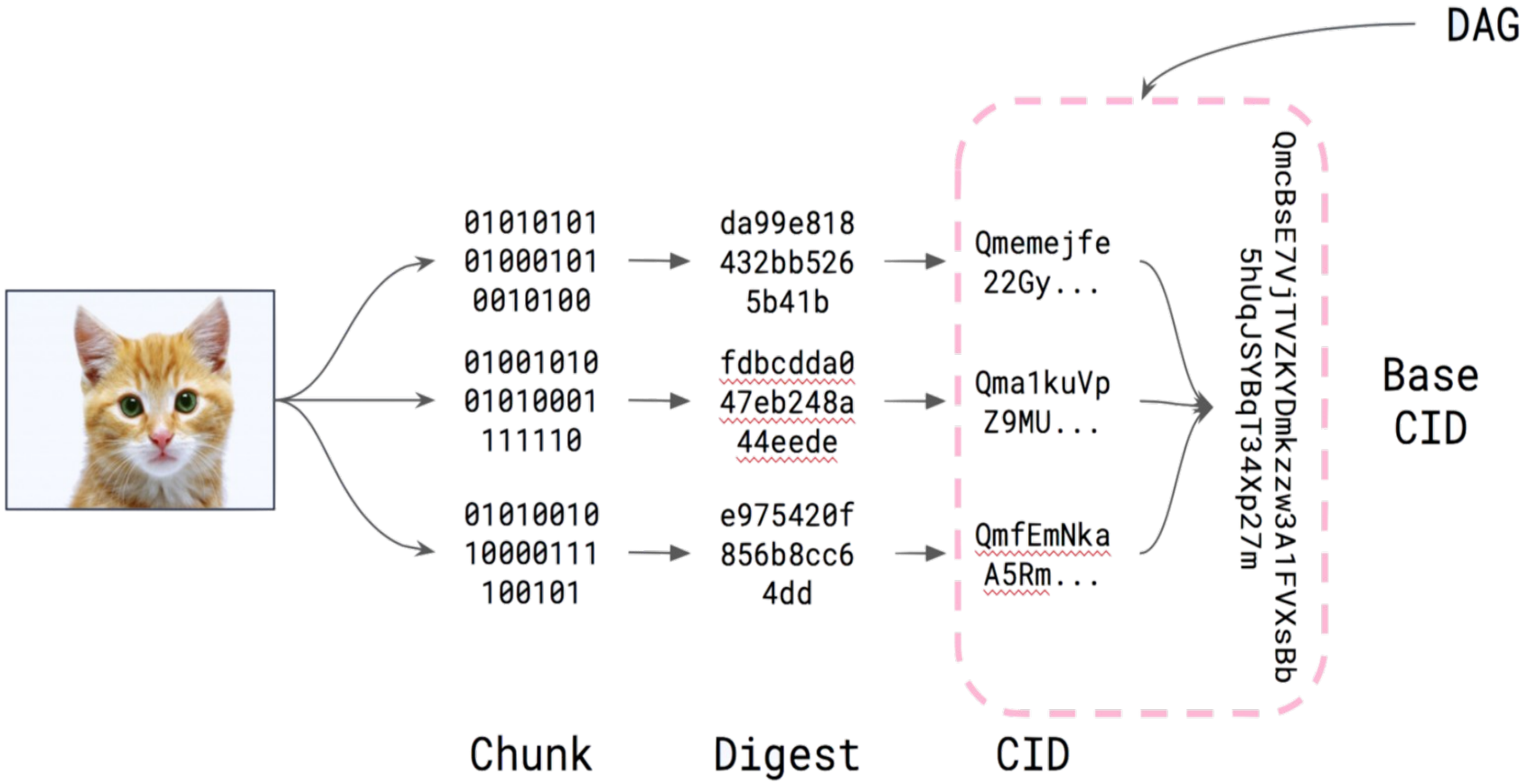
CID

- Content Identifiers (CIDs): is a label used to point to material in IPFS. It doesn’t indicate where the content is stored, but it forms a kind of address based on the content itself. CIDs are short, regardless of the size of their underlying content.

- CIDs are based on the content’s cryptographic hash (SHA-1 (used by Git), SHA-256, or BLAKE2).

- CID – Content Identifier – Immutable, Verifiable, Trust-less, Permanent

  - Multicodec - Multicodec is an identifier indicating the format of the target content. It helps people and software know how to interpret that content after it has been fetched.
  - Multihash - Multihash is a protocol for differentiating outputs from various well-established hash functions, addressing size and encoding considerations.
  - Multiformats -The Multiformats project is a collection of protocols that aim to future-proof systems today. A key element is enhancing format values with self-description. This allows for interoperability, protocol agility, and promotes extensibility.


DAG
01010101
01000101
0010100
01001010
01010001
111110
01010010
10000111
100101
da99e818
432bb526
5b41b
fdbcdda0
47eb248a
44eede
e975420f
856b8cc6
4dd
Qmemejfe
22Gy...
Qma1kuVp
Z9MU...
QmfEmNka
A5Rm...
QmcBsE7VjTVZKYDmkzzw3A1FVXsBb
5hUqJSYBqT34Xp27m
Base
CID
Chunk
Digest
CID

Pinning

- Pinning is the method of telling an IPFS node that particular data is important and so it will never be removed from that node's cache.

- A variant of pinning that uses a third-party service to ensure that data persists on IPFS, even when your local node goes offline or your local copy of data is deleted during garbage collection.

- Pinning is normally done at the local node (your node holding the data) level rather then global. Some nodes will allow local to them, some will not.

- Pinning of data can make it faster to retrieve being it may be on multiple nodes at once.

IPNS (interplanetary naming system)

- IPFS uses content-based addressing; it creates an address of a file or folder based on data contained within the file. If you were to share an IPFS address such as **/ipfs/QmbezGequPwcsWo8UL4wDF6a8hYwM1hmbzYv2mnKkEWa** with someone, you would need to give the person a new link every time you update the content.

- A name in IPNS is the hash of a public key. It is associated with a record containing information about the hash it links to that is signed by the corresponding private key. New records can be signed and published at any time.

IPFS Gateway

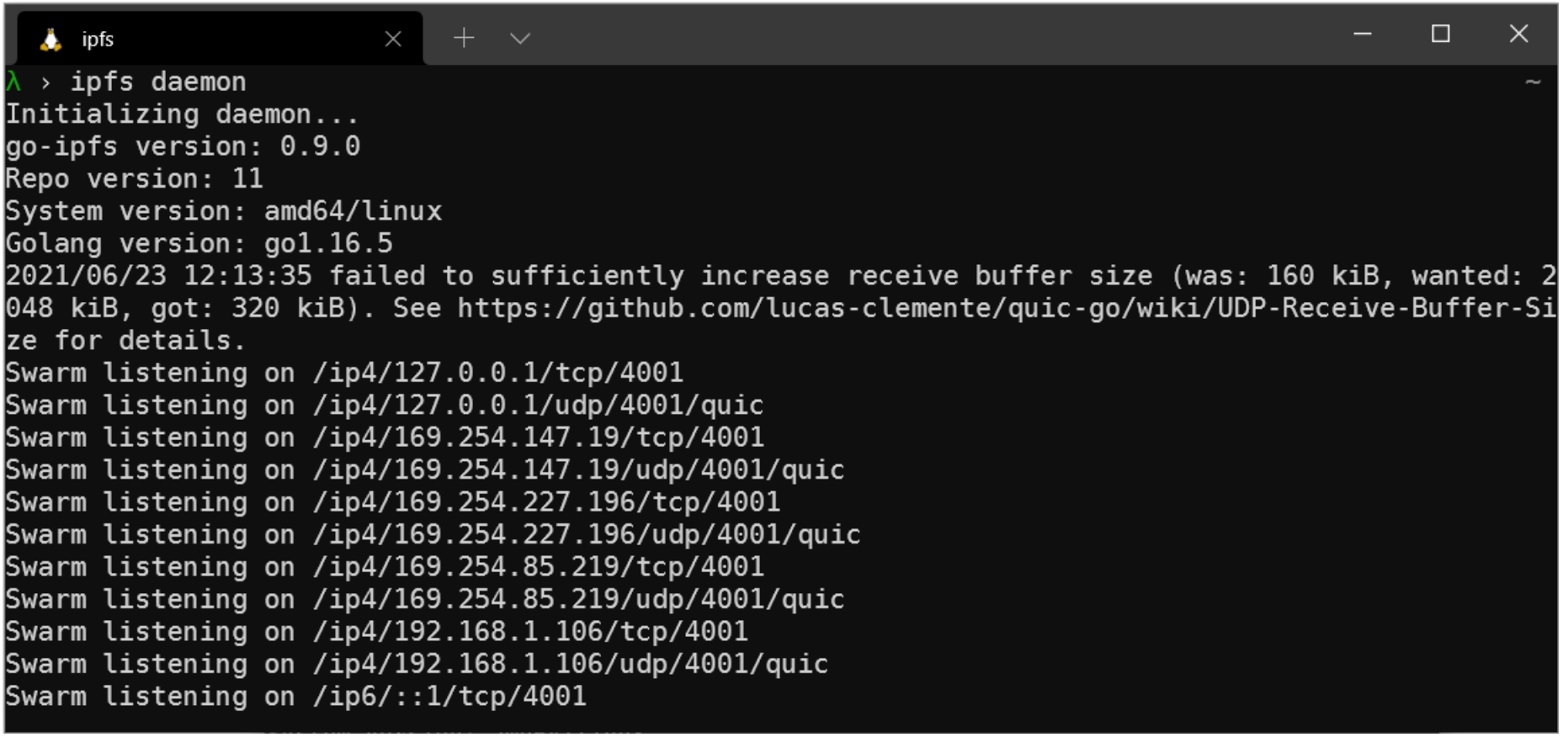
- An IPFS Gateway acts as a bridge between traditional web browsers and IPFS. Through the gateway, users can browse files and websites stored in IPFS as if they were stored on a traditional web server.
  - By default, go-ipfs nodes run a gateway at http://127.0.0.1:8080/.
  - We also provide a public gateway at https://ipfs.io. If you've ever seen a link in the form https://ipfs.io/ipfs/Qm..., that's being served from our gateway.
  - https://ipfs.io/ipfs/QmfM2r8seH2GiRaC4esTjeraXEachRt8ZsSeGaWTPLyMoG?filename=hello_world.txt

How to install / Use

- https://docs.ipfs.io/install/

- It will depend on your OS, command line based is the fastest, most activities are accessed by web interface.

- Ipfs daemon
- http://[my:ip:fd:ad:dr:es:s]:5001/webui


ipfs
λ › ipfs daemon
~
Initializing daemon...
go-ipfs version: 0.9.0
Repo version: 11
System version: amd64/linux
Golang version: go1.16.5
2021/06/23 12:13:35 failed to sufficiently increase receive buffer size (was: 160 kiB, wanted: 2
048 kiB, got: 320 kiB). See https://github.com/lucas-clemente/quic-go/wiki/UDP-Receive-Buffer-Si
ze for details.
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/127.0.0.1/udp/4001/quic
Swarm listening on /ip4/169.254.147.19/tcp/4001
Swarm listening on /ip4/169.254.147.19/udp/4001/quic
Swarm listening on /ip4/169.254.227.196/tcp/4001
Swarm listening on /ip4/169.254.227.196/udp/4001/quic
Swarm listening on /ip4/169.254.85.219/tcp/4001
Swarm listening on /ip4/169.254.85.219/udp/4001/quic
Swarm listening on /ip4/192.168.1.106/tcp/4001
Swarm listening on /ip4/192.168.1.106/udp/4001/quic
Swarm listening on /ip6/::1/tcp/4001

IPFS

STATUS

FILES

EXPLORE

PEERS

SETTINGS

UI v2.11.4
Revision 69cd414
See the code
Report a bug

QmHash/bafyHash

Browse

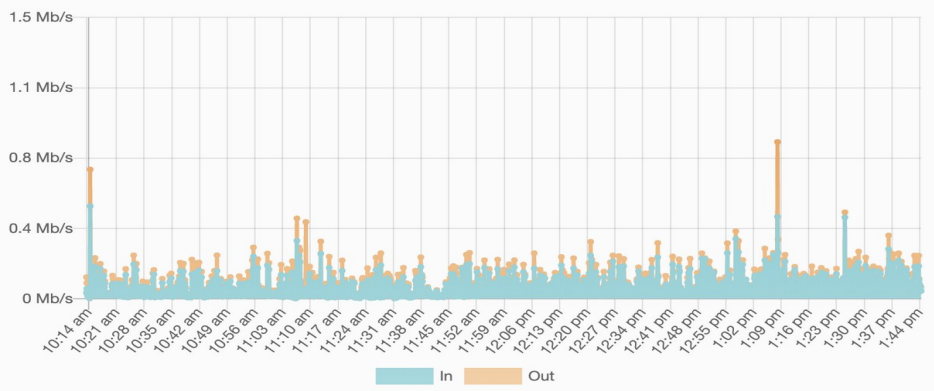
Connected to IPFS

Hosting 13.9 GB of files — **Discovered** 848 peers

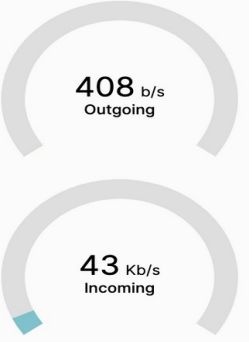
PEER ID	QmLaShMVN9CM2TShGMSwGfYG3P1P5bfhpRGgYJ3igEjgap
AGENT	go-ipfs v0.7.0
UI	v2.11.4

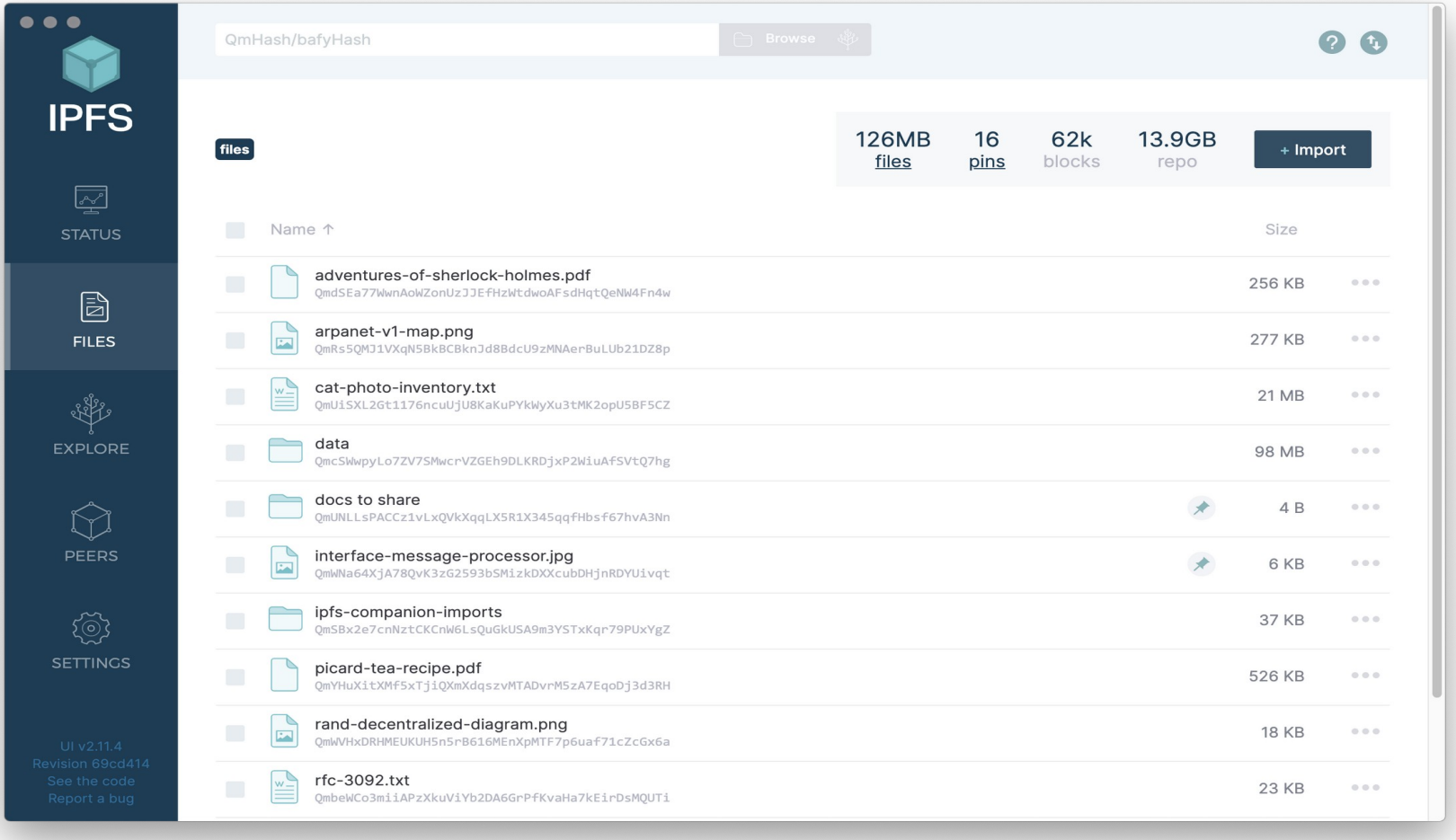
▶ Advanced

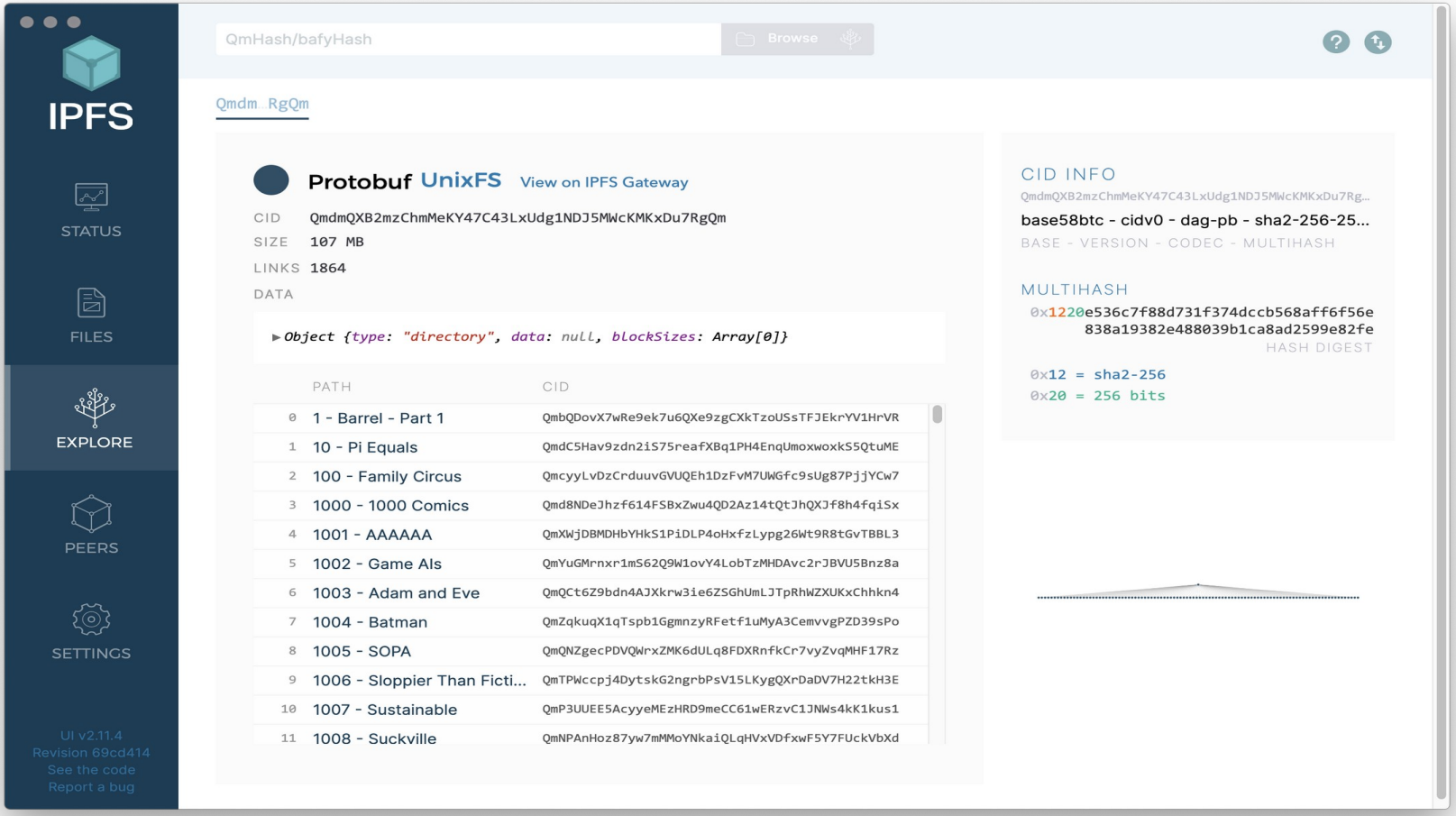
BANDWIDTH OVER TIME

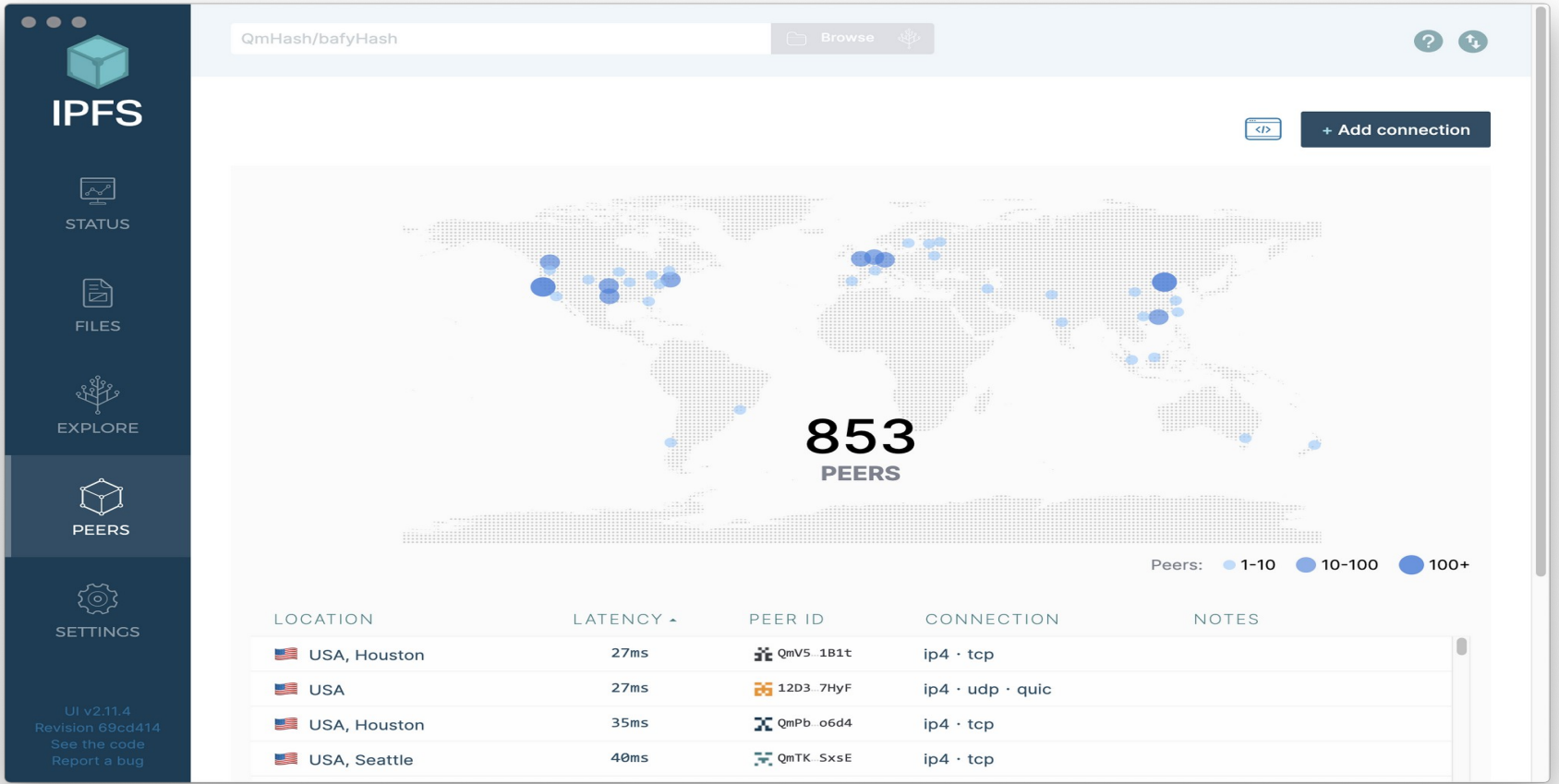


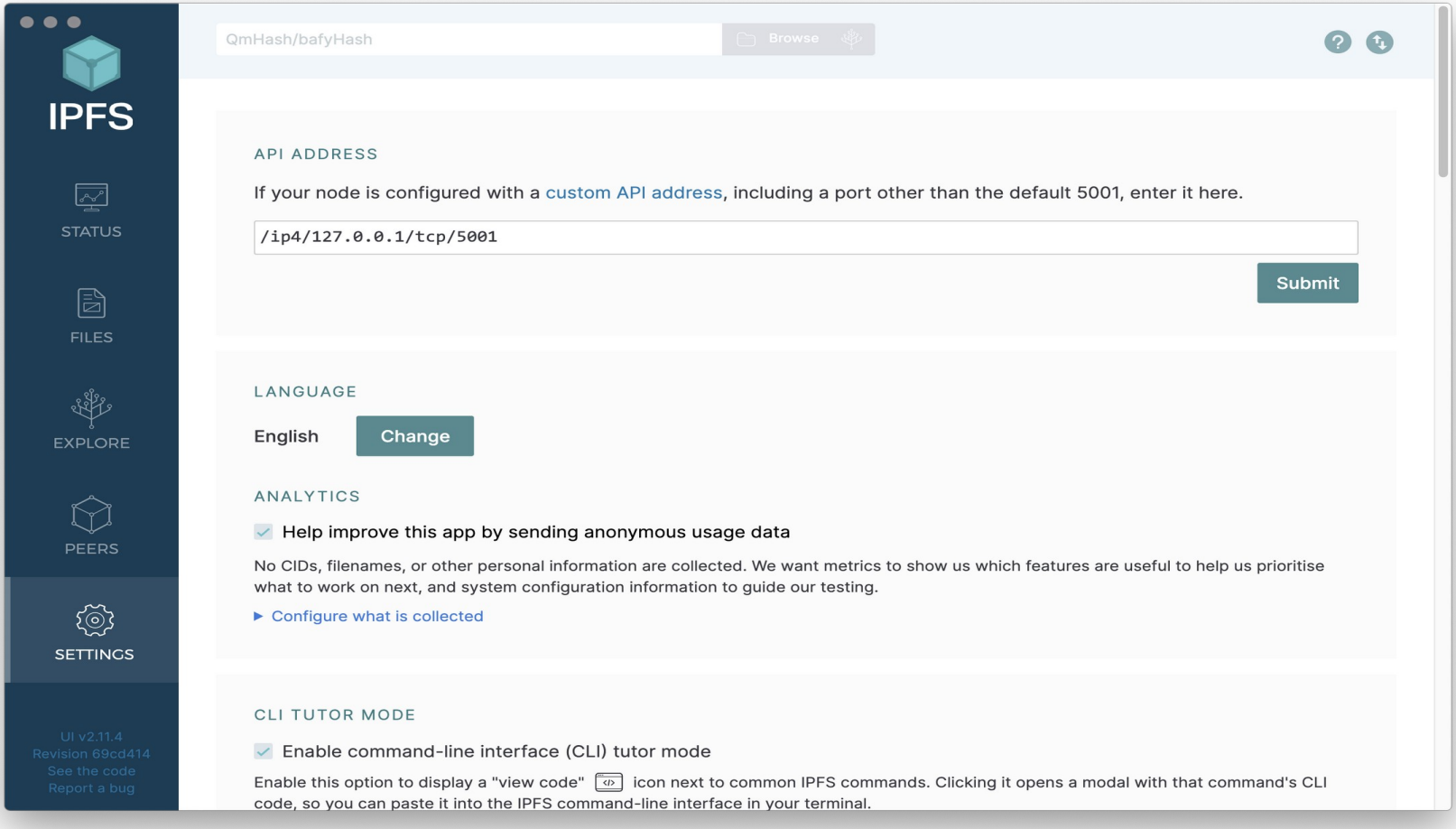
NETWORK TRAFFIC




IPFS
STATUS
FILES
EXPLORE
PEERS
SETTINGS
UI v2.11.4
Revision 69cd414
See the code
Report a bug
QmHash/bafyHash
Browse
files
126MB
files
16
pins
62k
blocks
13.9GB
repo
+ Import
Name ↑
Size
adventures-of-sherlock-holmes.pdf
QmdSEa77WwnAoWZonUzJJEfHzWtdwoAFsdHqtQeNW4Fn4w
256 KB
arpanet-v1-map.png
QmRs5QMJ1VXqN5BkBCBknJd8BdcU9zMNAerBuLUb21DZ8p
277 KB
cat-photo-inventory.txt
QmUiSXL2Gt1176ncuUjU8KaKuPYkWyXu3tMK2opU5BF5CZ
21 MB
data
QmcSWwpyLo7ZV7SMwcrVZGEh9DLKRDjxP2WiuAfSVtQ7hg
98 MB
docs to share
QmUNLLsPACCz1vLxQVkXqqLX5R1X345qqfHbsf67hvA3Nn
4 B
interface-message-processor.jpg
QmWNa64XjA78QvK3zG2593bSMizkDXXcubDHjnRDYUivqt
6 KB
ipfs-companion-imports
QmSBx2e7cnNztCKCnW6LsQuGkUSA9m3YSTxKqr79PUxYgZ
37 KB
picard-tea-recipe.pdf
QmYHuXitXMf5xTjiQXmXdqszvMTADvrM5zA7EqoDj3d3RH
526 KB
rand-decentralized-diagram.png
QmWVHxDRHMEUKUH5n5rB616MEnXpMTF7p6uaf71cZcGx6a
18 KB
rfc-3092.txt
QmbeWCo3miiAPzXkuViYb2DA6GrPfKvaHa7kEirDsMQUTi
23 KB


IPFS
STATUS
FILES
EXPLORE
PEERS
SETTINGS
UI v2.11.4
Revision 69cd414
See the code
Report a bug
QmHash/bafyHash
Browse
Qmdm RgQm
Protobuf UnixFS
View on IPFS Gateway
CID QmdmQXB2mzChmMeKY47C43LxUdg1NDJ5MWcKMKxDu7RgQm
SIZE 107 MB
LINKS 1864
DATA
▶ Object {type: "directory", data: null, blockSizes: Array[0]}
PATH
CID
0 1 - Barrel - Part 1 QmbQDovX7wRe9ek7u6QXe9zgCXkTzoUSsTFJEkrYV1HrVR
1 10 - Pi Equals QmdC5Hav9zdn2iS75reafXBq1PH4EnqUmoxwoxkS5QtuME
2 100 - Family Circus QmcyyLvDzCrduuvGVUQEh1DzFvM7UWGfc9sUg87PjjYCw7
3 1000 - 1000 Comics Qmd8NDeJhzf614FSBxZwu4QD2Az14tQtJhQXJf8h4fqiSx
4 1001 - AAAAAA QmXWjDBMDHbYHkS1PiDLP4oHxfzLypg26Wt9R8tGvTBBL3
5 1002 - Game AIs QmYuGMrnxr1mS62Q9W1ovY4LobTzMHDAvc2rJBVU5Bnz8a
6 1003 - Adam and Eve QmQCt6Z9bdn4AJXkrw3ie6ZSGhUmLJTpRhWZXUKxChhkn4
7 1004 - Batman QmZqkuqX1qTspb1GgmnzyRFetf1uMyA3CemvvgPZD39sPo
8 1005 - SOPA QmQNZgecPDVQWrxZMK6dULq8FDXRnfkCr7vyZvqMHF17Rz
9 1006 - Sloppier Than Ficti... QmTPWccpj4DytskG2ngrbPsV15LKygQXrDaDV7H22tkH3E
10 1007 - Sustainable QmP3UUEE5AcyyeMEzHRD9meCC61wERzvC1JNWs4kK1kus1
11 1008 - Suckville QmNPAnHoz87yw7mMMoYNkaiQLqHVxVDfxwF5Y7FUckVbXd
CID INFO
QmdmQXB2mzChmMeKY47C43LxUdg1NDJ5MWcKMKxDu7Rg...
base58btc - cidv0 - dag-pb - sha2-256-25...
BASE - VERSION - CODEC - MULTIHASH
MULTIHASH
0x1220e536c7f88d731f374dccb568aff6f56e838a19382e488039b1ca8ad2599e82fe
HASH DIGEST
0x12 = sha2-256
0x20 = 256 bits


IPFS
STATUS
FILES
EXPLORE
PEERS
SETTINGS
UI v2.11.4
Revision 69cd414
See the code
Report a bug
QmHash/bafyHash
Browse
+ Add connection
853
PEERS
Peers: 1-10 10-100 100+
LOCATION
LATENCY
PEER ID
CONNECTION
NOTES
USA, Houston 27ms QmV5..1B1t ip4 · tcp
USA 27ms 12D3..7HyF ip4 · udp · quic
USA, Houston 35ms QmPb..o6d4 ip4 · tcp
USA, Seattle 40ms QmTK..SxsE ip4 · tcp


IPFS
STATUS
FILES
EXPLORE
PEERS
SETTINGS
UI v2.11.4
Revision 69cd414
See the code
Report a bug
QmHash/bafyHash
Browse
API ADDRESS
If your node is configured with a custom API address, including a port other than the default 5001, enter it here.
/ip4/127.0.0.1/tcp/5001
Submit
LANGUAGE
English
Change
ANALYTICS
Help improve this app by sending anonymous usage data
No CIDs, filenames, or other personal information are collected. We want metrics to show us which features are useful to help us prioritise what to work on next, and system configuration information to guide our testing.
Configure what is collected
CLI TUTOR MODE
Enable command-line interface (CLI) tutor mode
Enable this option to display a "view code" icon next to common IPFS commands. Clicking it opens a modal with that command's CLI code, so you can paste it into the IPFS command-line interface in your terminal.

Links of interest

- https://ipfs.io/
- https://docs.ipfs.io/concepts/what-is-ipfs/
- https://docs.ipfs.io/concepts/what-is-ipfs/#decentralization
- https://proto.school/content-addressing
- https://webui.ipfs.io/

DEMO

Any Questions?

Thank you

Go forth and expand