



InterPlanetary File System (IPFS)

Akash Trehan

@cod3maxx | www.akashtrehan.com



Blueprint

- Goal
- Why do we need IPFS?
- High level overview / Usage
- The IPFS Stack
- Problems with IPFS
- Cool Apps
- Some cool latest updates
- Demo



Blueprint

- **Goal**
- Why do we need IPFS?
- High level overview / Usage
- The IPFS Stack
- Problems with IPFS
- Cool Apps
- Some cool latest updates
- Demo



Blueprint

- Goal
- **Why do we need IPFS?**
- High level overview / Usage
- The IPFS Stack
- Problems with IPFS
- Cool Apps
- Some cool latest updates
- Demo

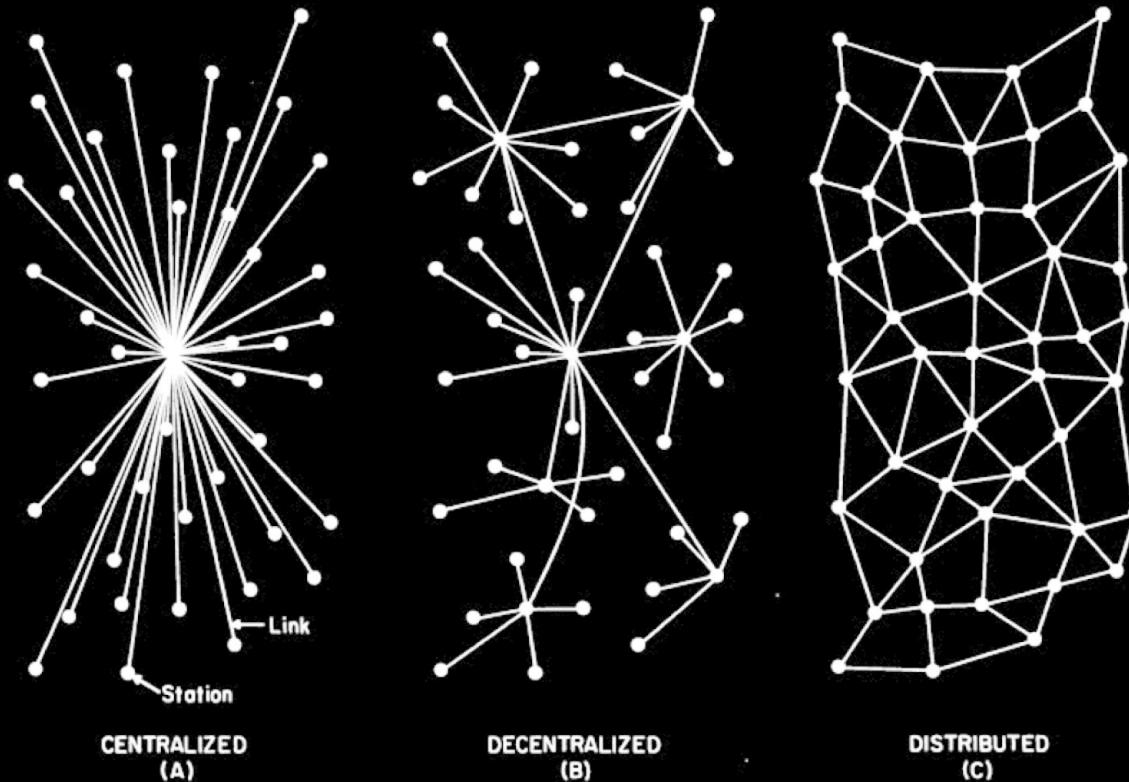
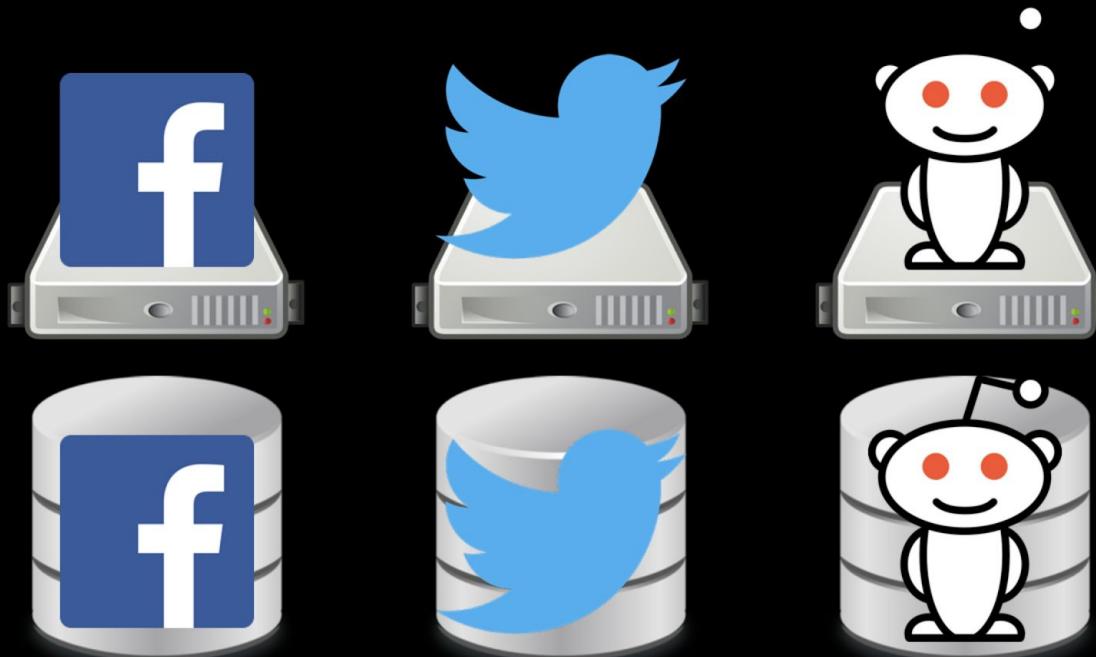
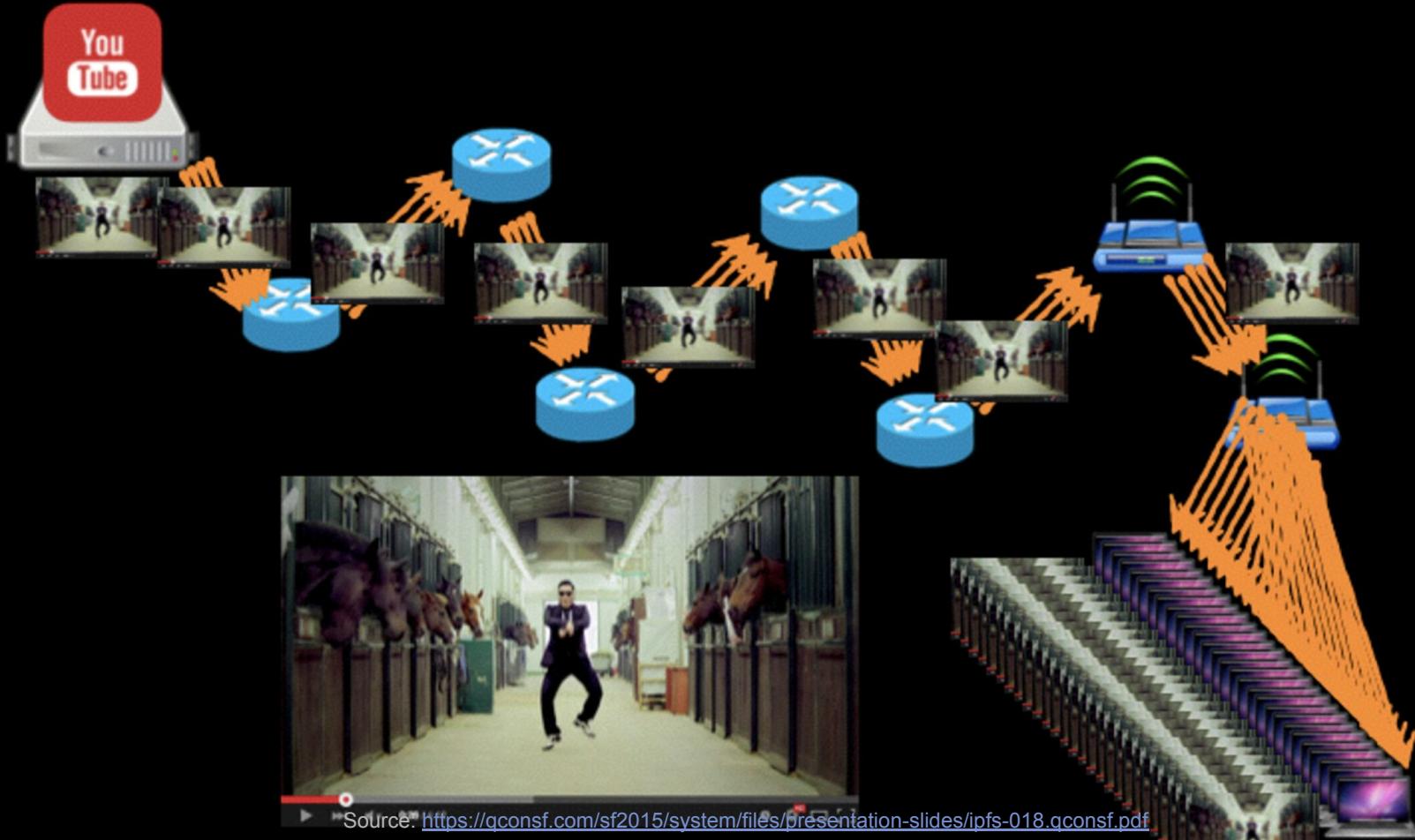


FIG. 1 – Centralized, Decentralized and Distributed Networks

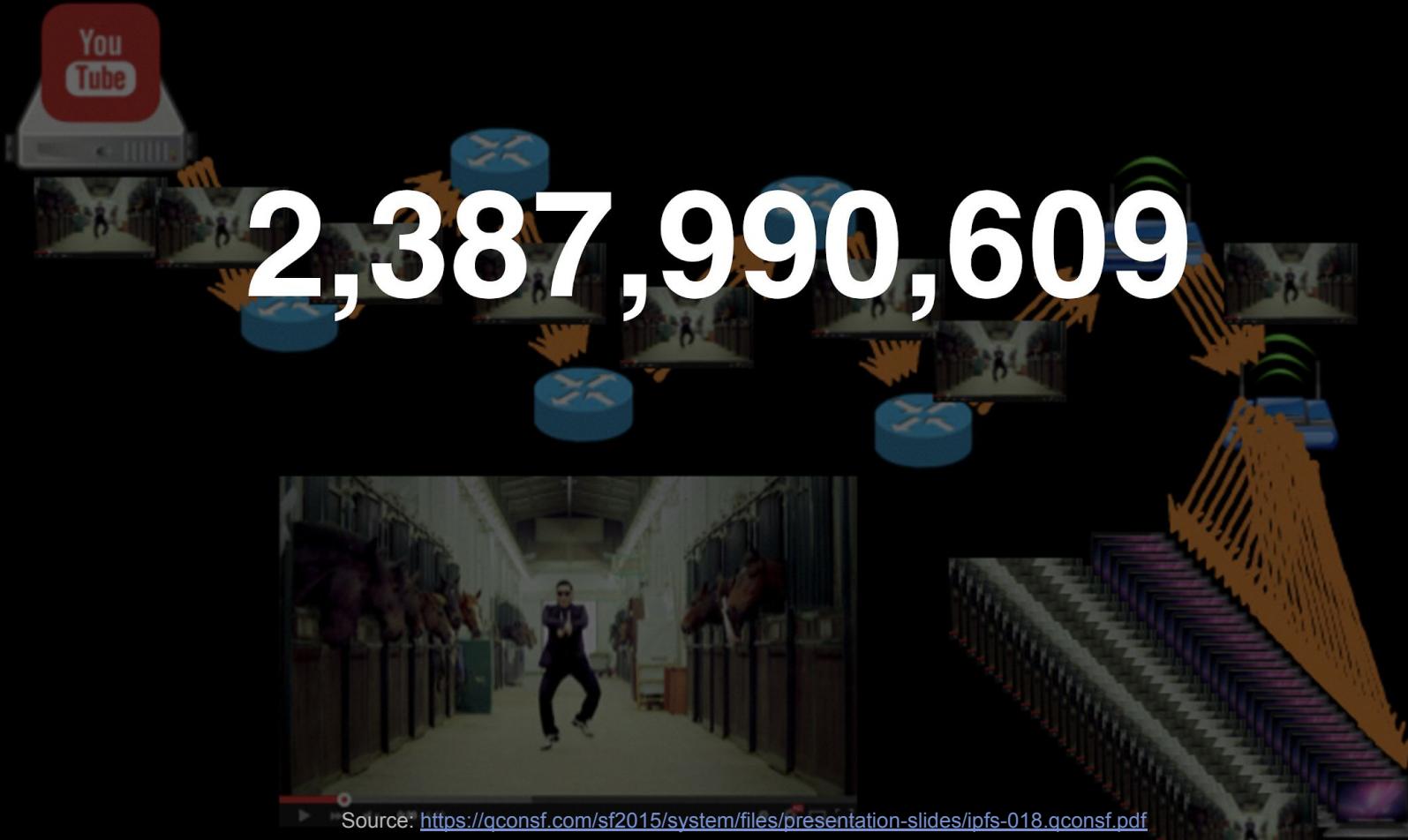
Data Control



$$200 \text{ MB} \times 30 \times 8 = 48 \text{ GB}$$



$$200 \text{ MB} \times 30 \times 8 = 48 \text{ GB}$$



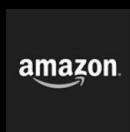
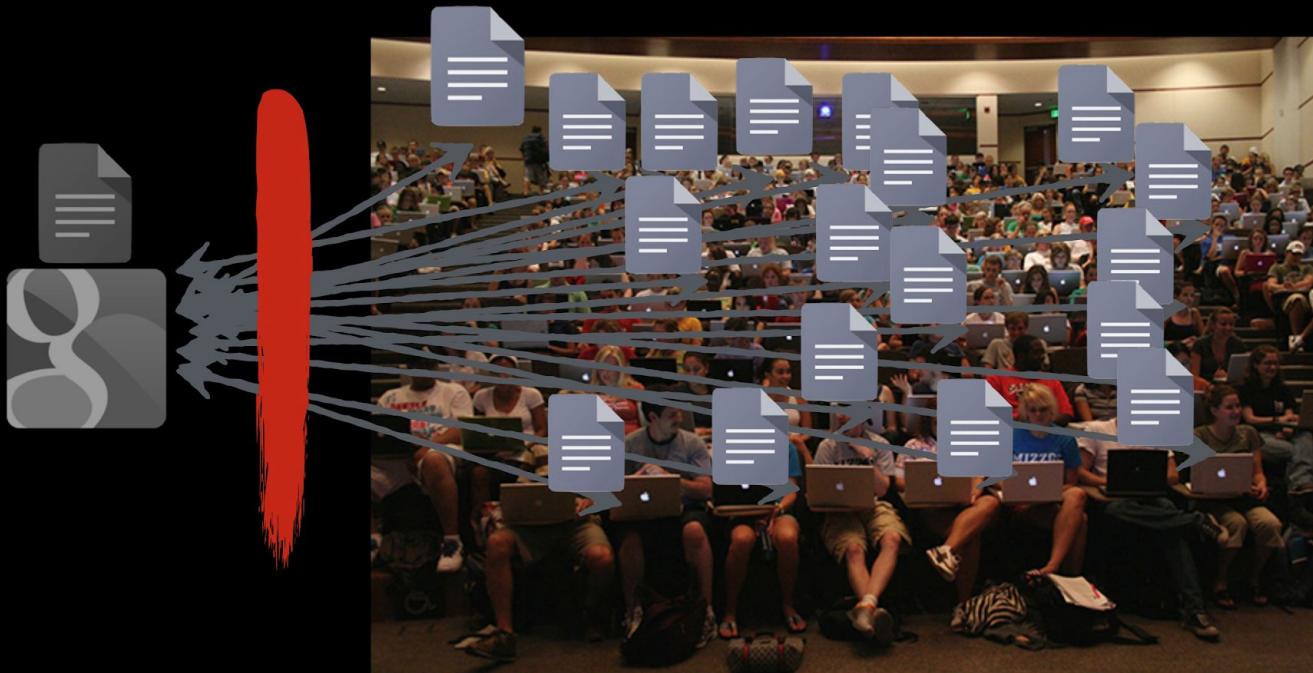
$200 \text{ MB} \times 30 \times 8 = 48 \text{ GB}$



2,387,990,609

477.6 PB

Disconnected / Offline

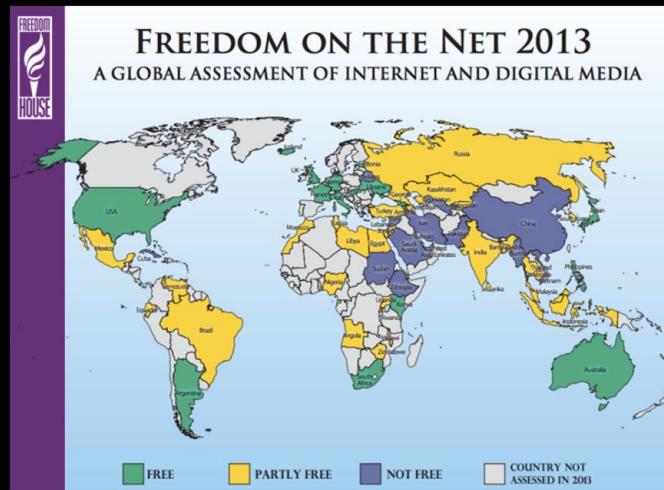


Natural Disasters



Source: <https://qconsf.com/sf2015/system/files/presentation-slides/ipsf-018.qconsf.pdf>

Human Disasters



surprise oppression

censorship





Offline



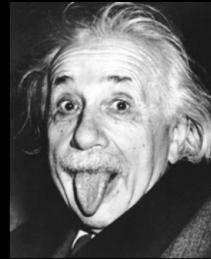
Distributed



Permanent



Safer



Smarter

IPFS



Faster



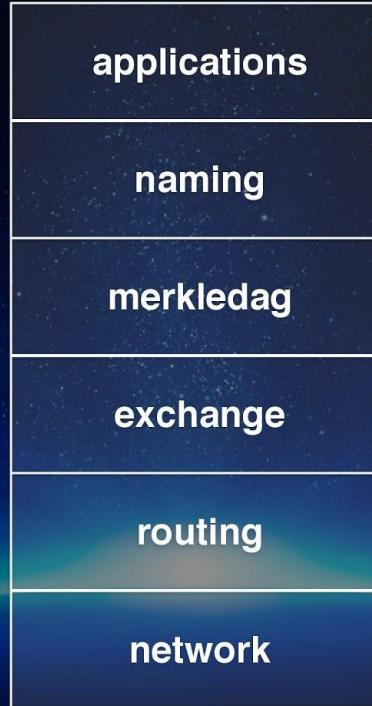
Blueprint

- Goal
- Why do we need IPFS?
- **High level overview / Usage**
- The IPFS Stack
- Problems with IPFS
- Cool Apps
- Some cool latest updates
- Demo



Blueprint

- Goal
- Why do we need IPFS?
- High level overview / Usage
- **The IPFS Stack**
- Problems with IPFS
- Cool Apps
- Some cool latest updates
- Demo



The IPFS Stack

applications

Etherpad

VLC

Git

Chat

Bitcoin

naming

IPNS

IPLD

exchange

Bitswap

routing

Kad DHT

mDNS

network

CJDNS

UDT

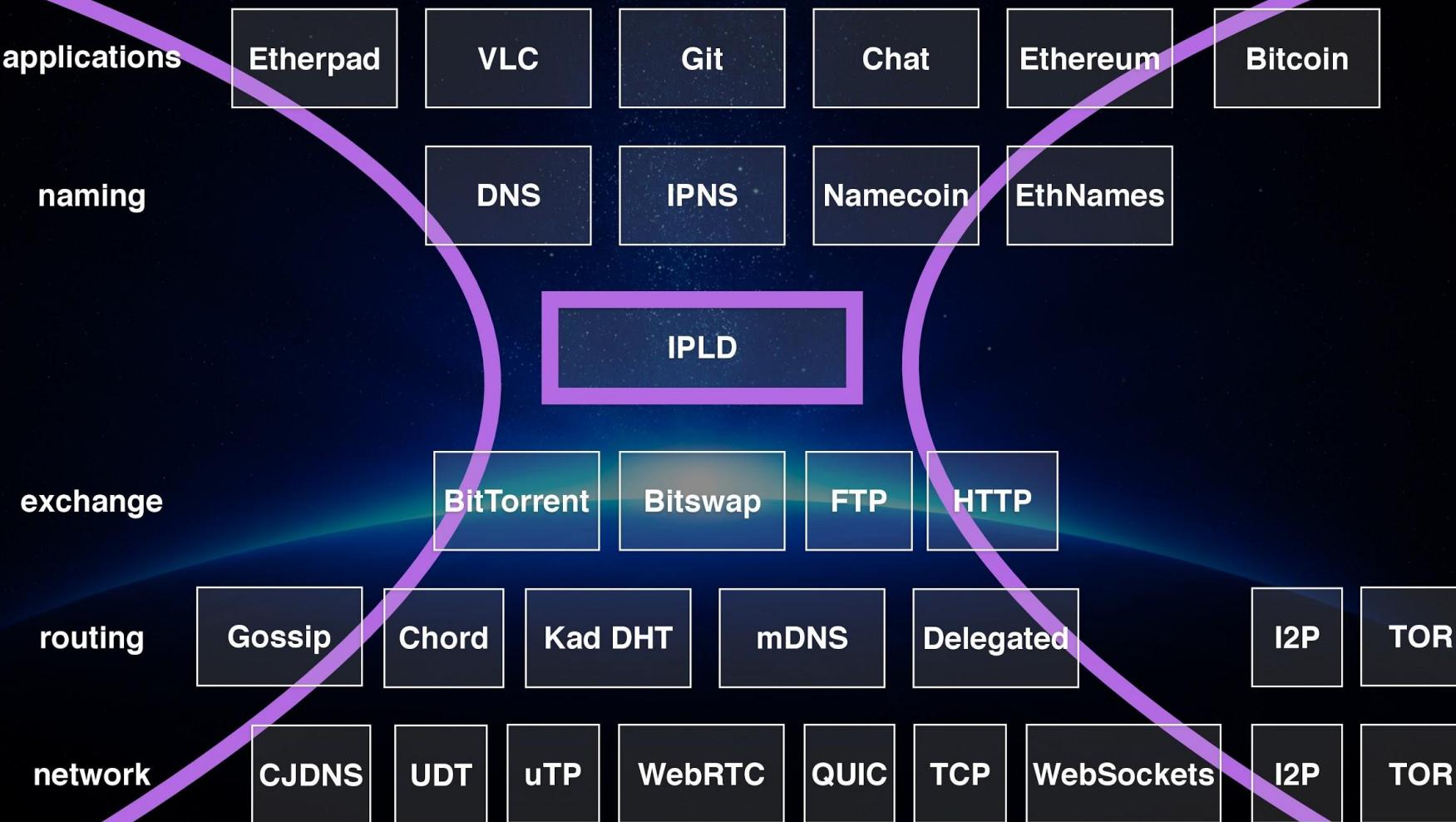
uTP

WebRTC

QUIC

TCP

WebSockets



The IPFS Stack

IPNS
IPLD

applications

Using the Data

naming
merkldag

Defining the Data

exchange

libp2p

routing
network

Moving the Data

The IPFS Stack

IPNS
IPLD

libp2p

applications

naming
merkldag

Using the Data

Defining the Data

exchange
routing
network

Moving the Data

libp2p - a collection of peer-to-peer protocols

Content Routing

mDNS

pub
sub

Kad
DHT

Kad
ICE

ICE

NAT
Traversal

Peer Routing

mDNS

DNS

DVs

Kad
DHT

STUN

TURN

Discovery

mDNS

boot
strap

DNS

Kad
DHT

PEX

PKI

Transports

TCP

uTP

QUIC

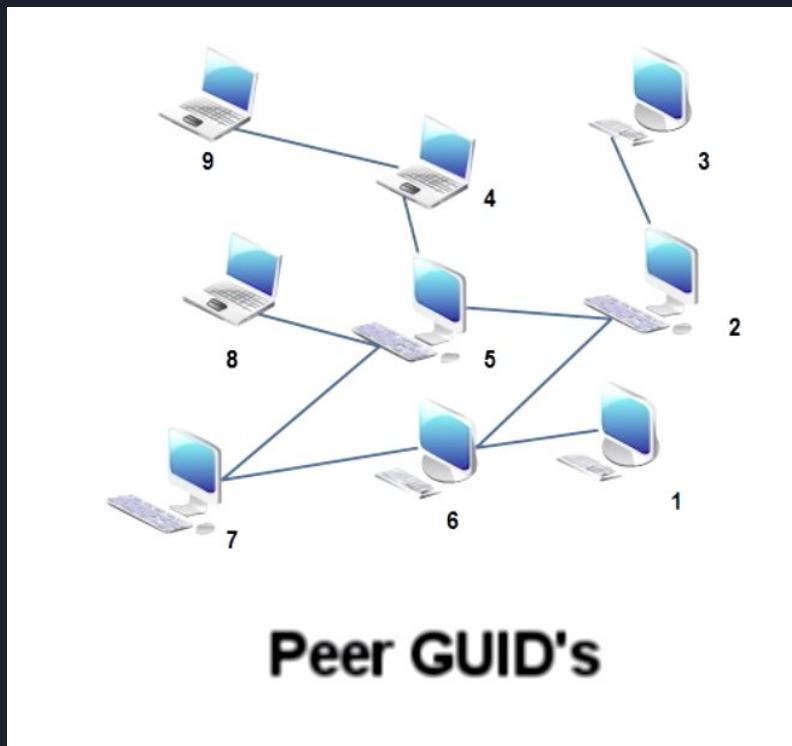
SCTP

BLE

TOR

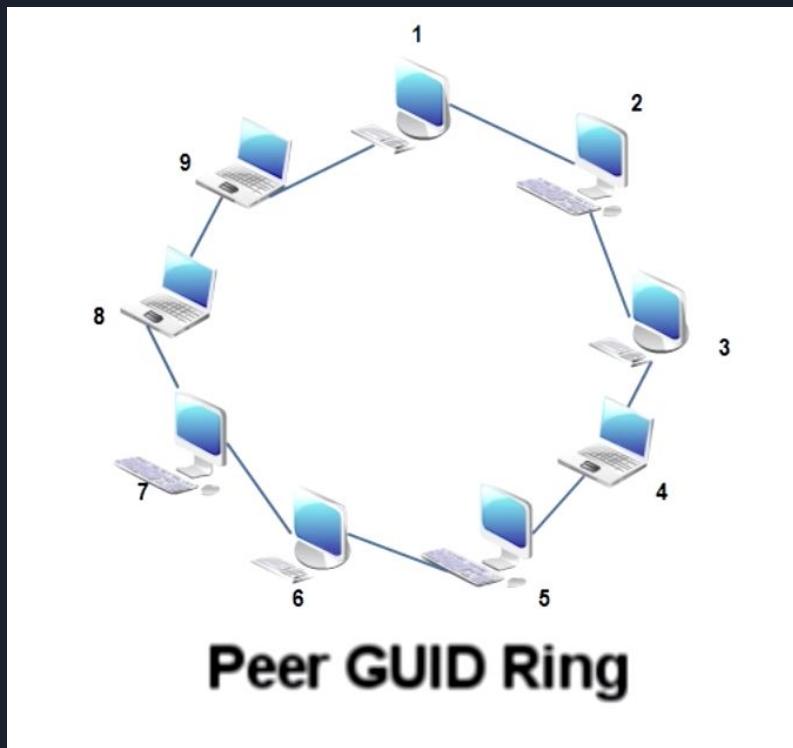
I2P

P2P networks and Distributed Hash Tables

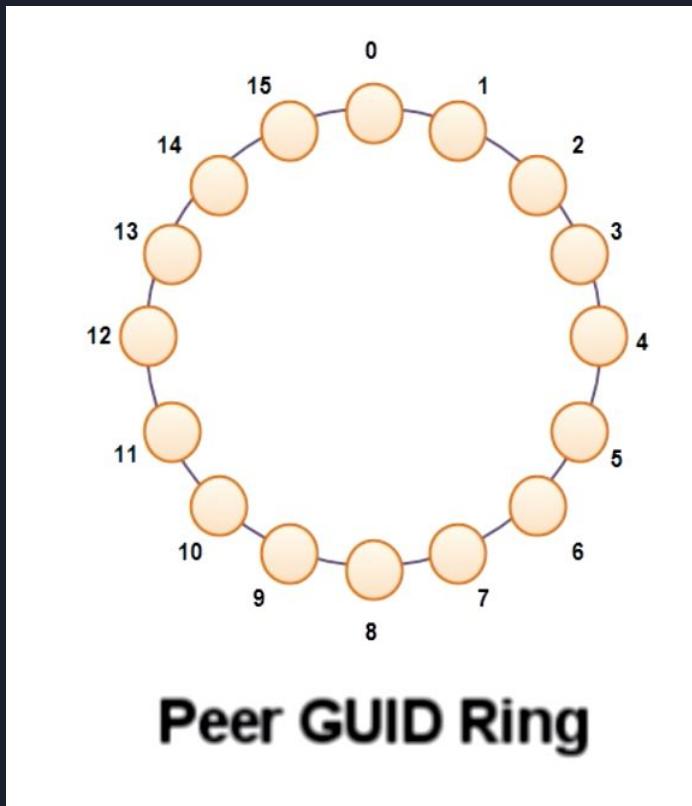


Source: <https://www.youtube.com/watch?v=kXyVqk3EbxE>

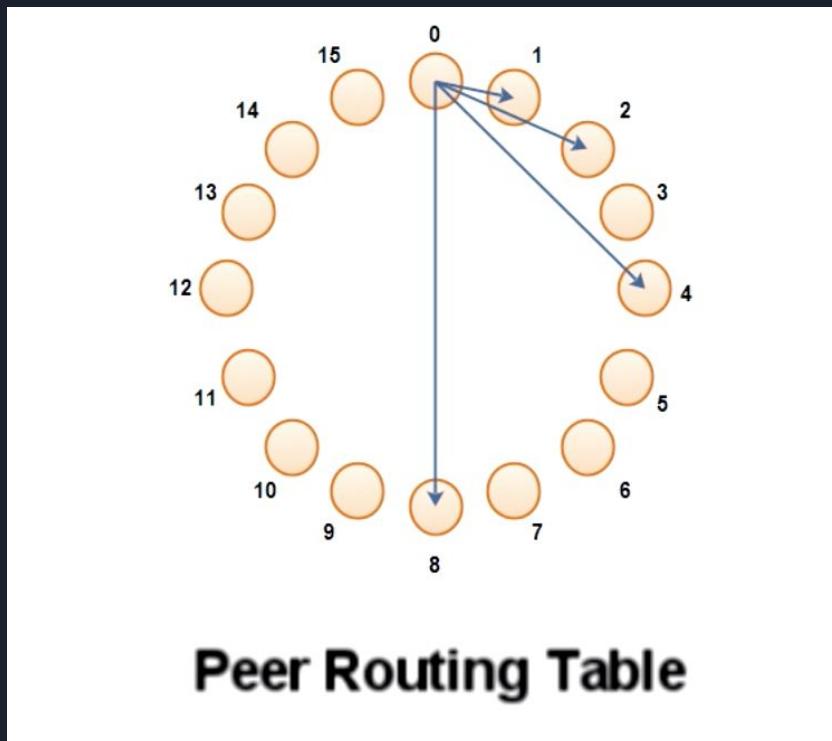
P2P networks and Distributed Hash Tables



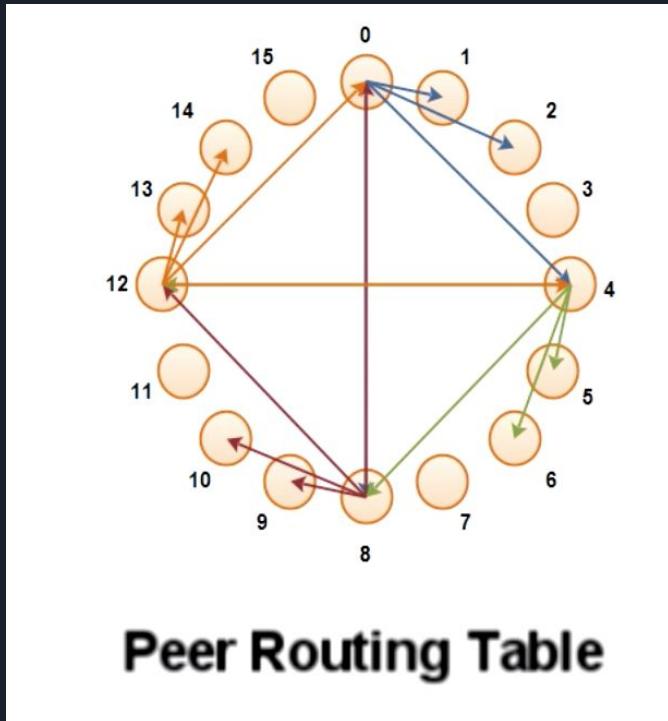
P2P networks and Distributed Hash Tables



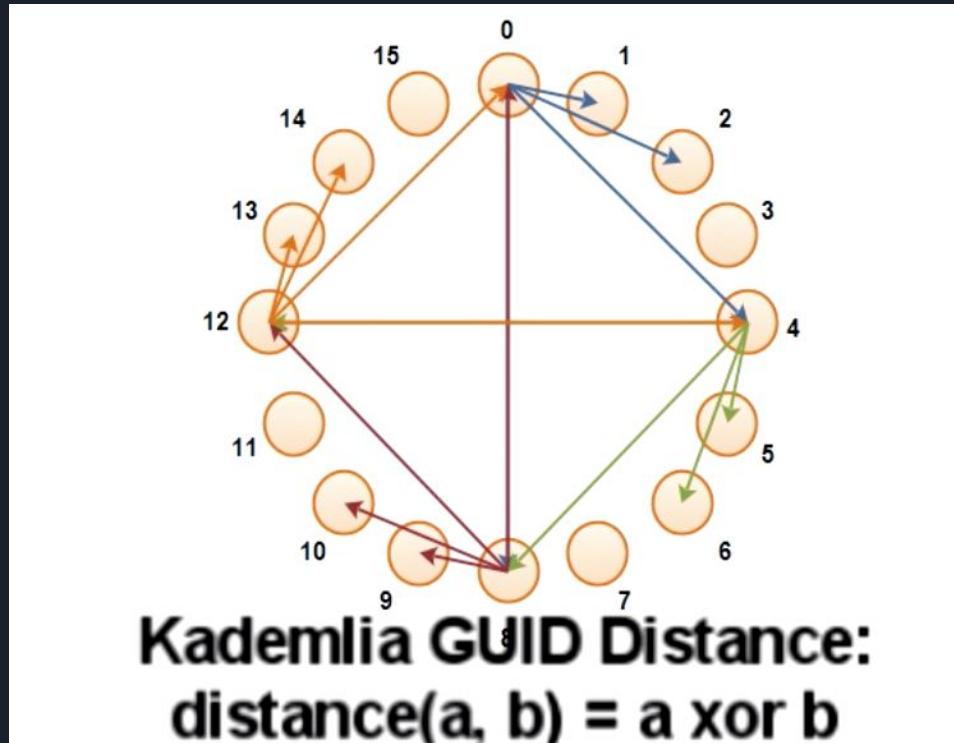
P2P networks and Distributed Hash Tables



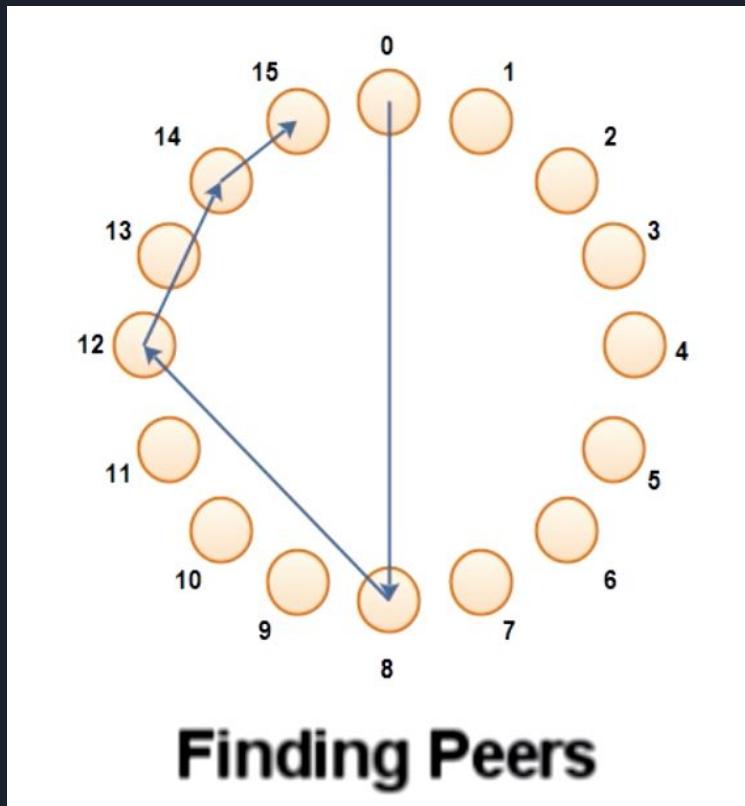
P2P networks and Distributed Hash Tables



P2P networks and Distributed Hash Tables



P2P networks and Distributed Hash Tables





IPFS Routing Interface

- FindPeer (node NodeID)
- SetValue (key []bytes, value []bytes)
- GetValue (key []bytes)
- ProvideValue(key Multihash)
- FindValuePeers (key Multihash, min int)



Incentivization

- Must incentivize nodes to seed when they do not need anything in particular, as they might have the blocks others want
- Moreover, leeches must be protected against
- The technique be effective with and resistant to other, unknown strategies
- Leniency to trusted peers
- IPFS uses a simple credit system called BitSwap



The BitSwap Protocol

Let the *debt ratio* r between a node and its peer be:

$$r = \frac{\text{bytes_sent}}{\text{bytes_recv} + 1}$$

Given r , let the probability of sending to a debtor be:

$$P(\text{send} | r) = 1 - \frac{1}{1 + \exp(6 - 3r)}$$



The BitSwap Protocol

- provides resistance to attackers who would create lots of new nodes (Sybil attacks)
- protects previously successful trade relationships, even if one of the nodes is temporarily unable to provide value
- eventually chokes relationships that have deteriorated until they improve.



The BitSwap Protocol Interface

- Peer.open (Nodeld, Ledger).
- Peer.send_want_list (WantList)
- Peer.send_block (Block)
- Peer.close (Bool)

The IPFS Stack

IPNS

IPLD

libp2p

applications

naming

merkldag

exchange

routing

network

Using the Data

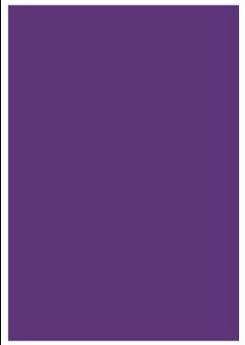
Defining the Data

Moving the Data



a web of merkle-links





`http://example.com/foo/bar/baz.png`

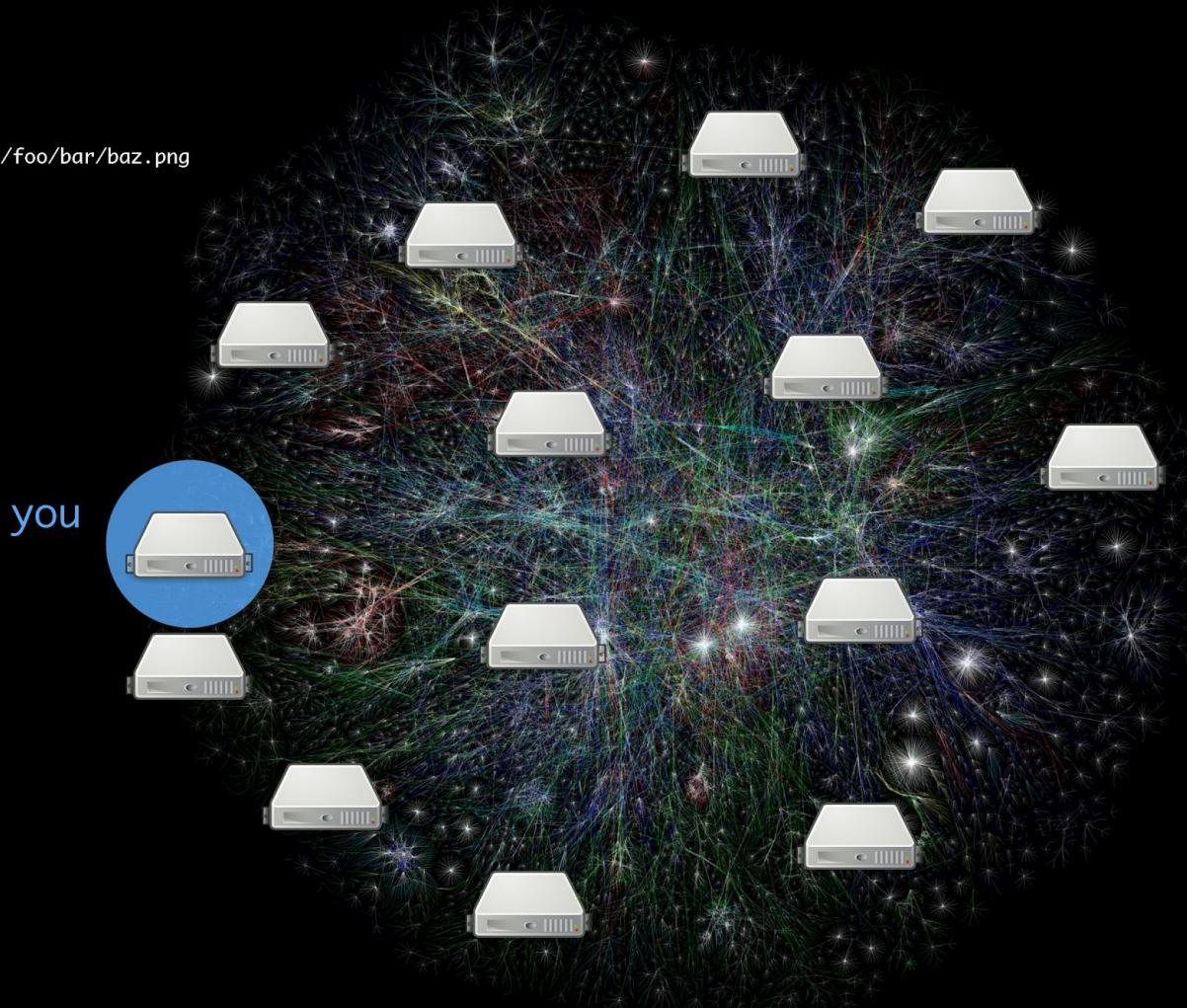


`http://10.20.30.40/foo/bar/baz.png`

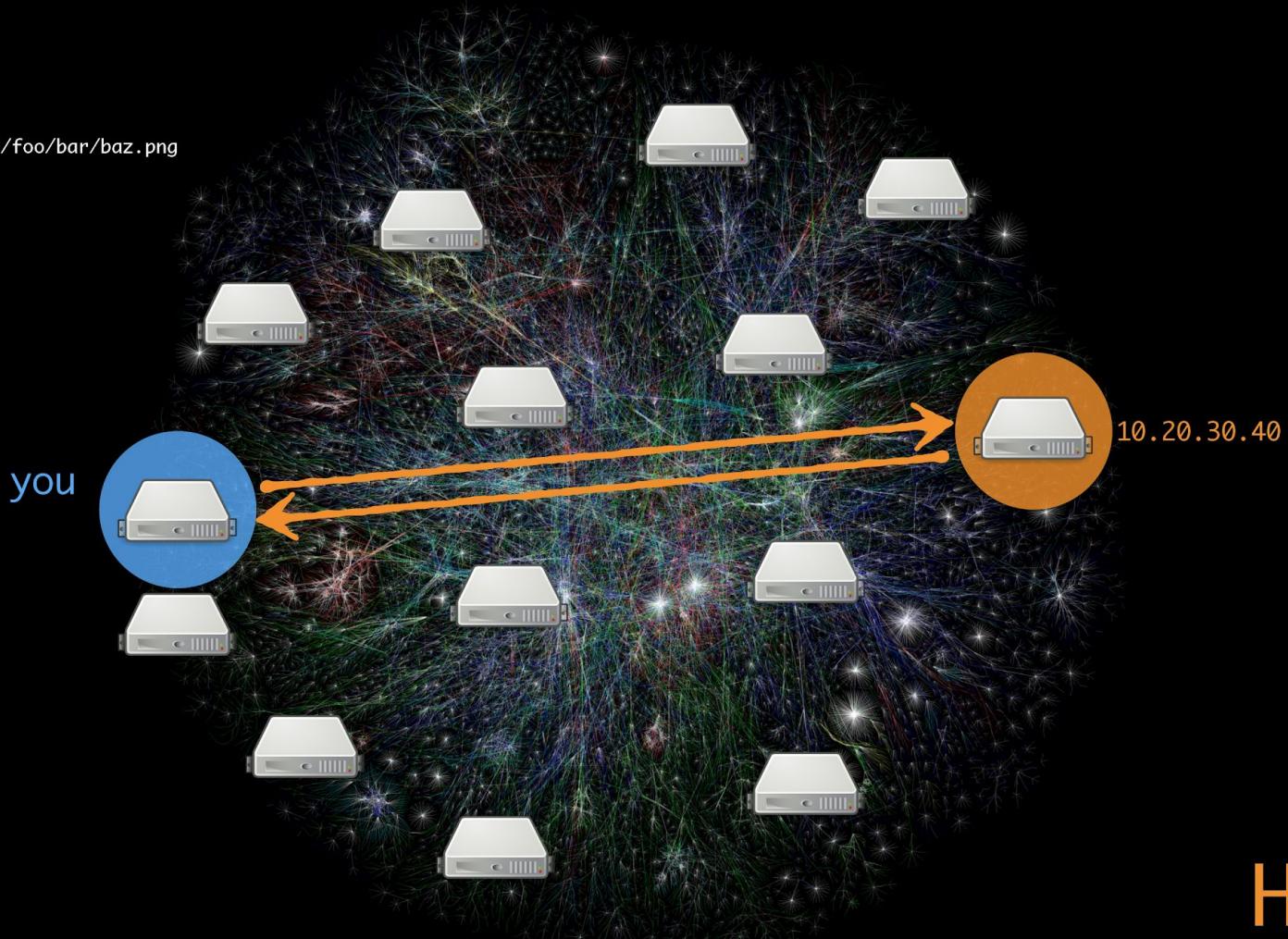
location

path

<http://10.20.30.40/foo/bar/baz.png>



<http://10.20.30.40/foo/bar/baz.png>



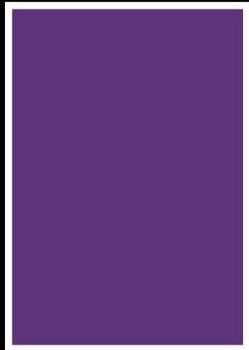
`http://example.com/foo/bar/baz.png`



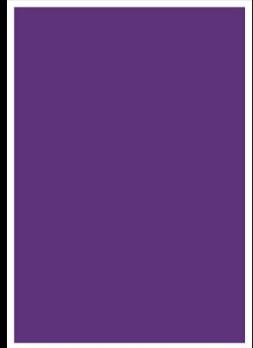
`http://10.20.30.40/foo/bar/baz.png`

location

path



hash() -> QmW98pJrc6FZ6



http://



http://

location

path

/ipns/example.com/foo/bar/baz.png



/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png

content

path

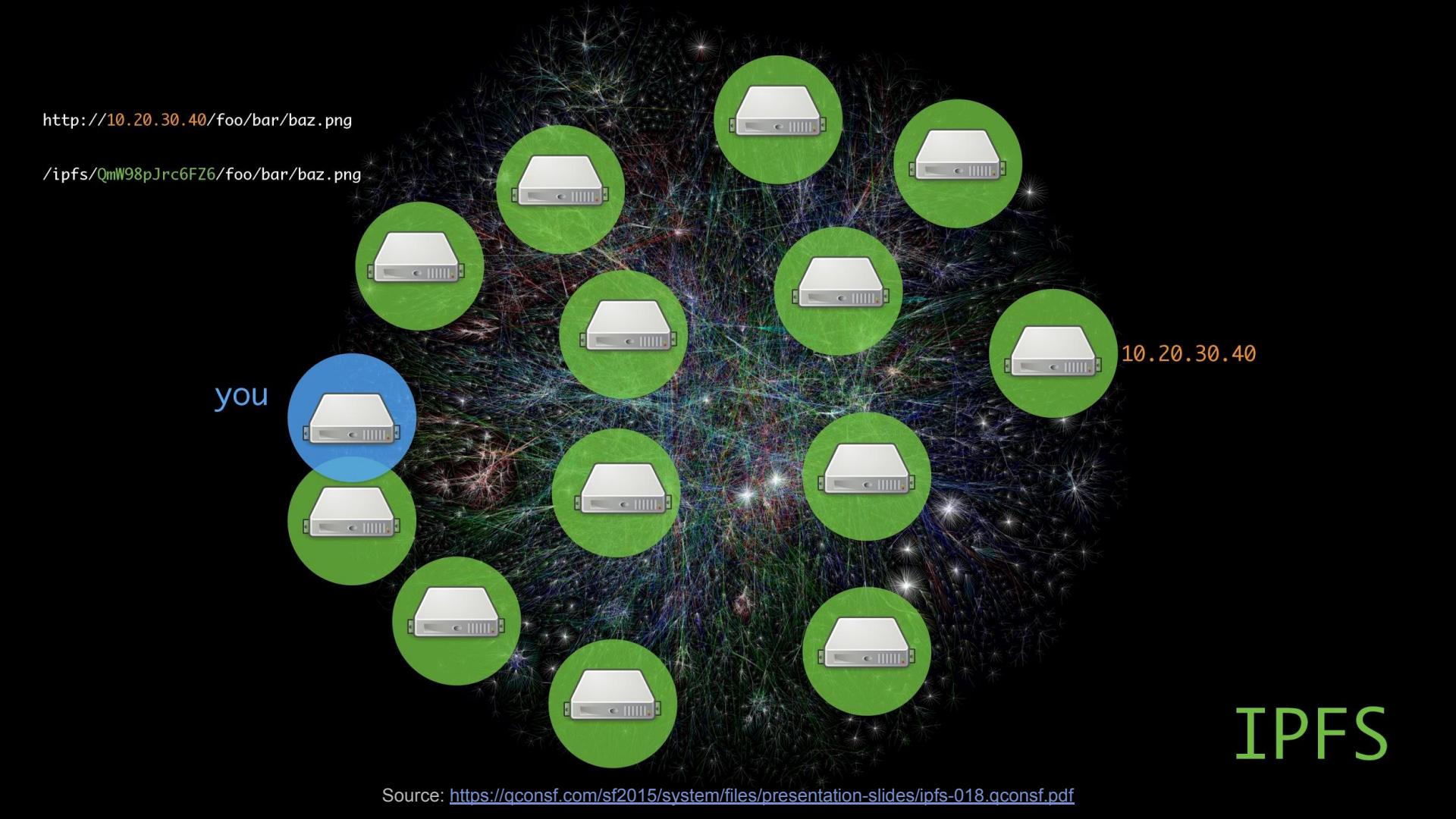
`http://10.20.30.40/foo/bar/baz.png`

`/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png`

`you`

`10.20.30.40`

HTTP



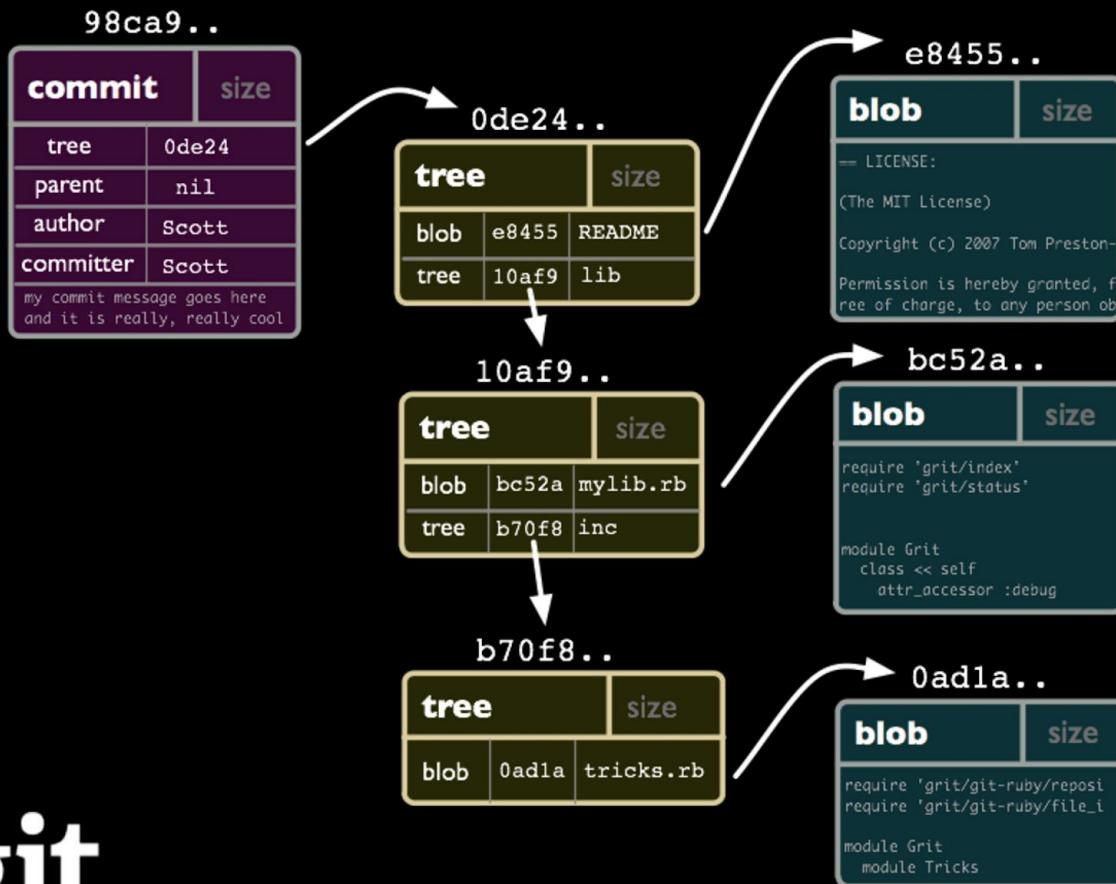
<http://10.20.30.40/foo/bar/baz.png>

</ipfs/QmW98pJrc6FZ6/foo/bar/baz.png>

you

10.20.30.40

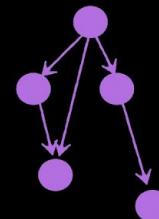
IPFS



Source: <https://qconf.com/sf2015/system/files/presentation-slides/ipfs-018.qconf.pdf>



in IPFS data forms a dag



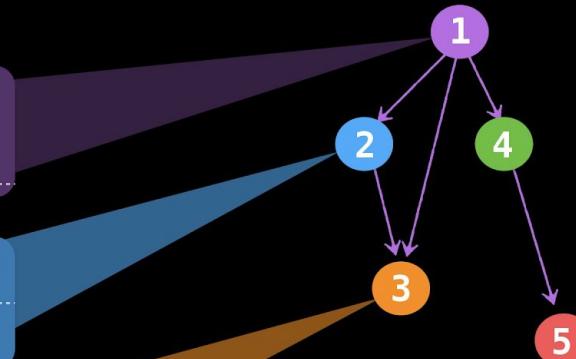
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!



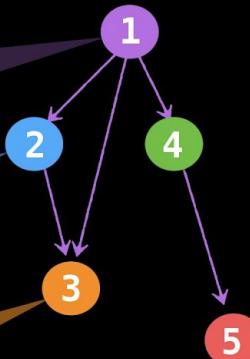
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!



suppose $H(1) = QmW98pJrc6FZ$

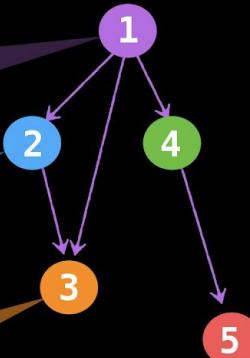
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!



suppose $H(1)$ = QmW98pJrc6FZ

We can resolve:

/ipfs/QmW98pJrc6FZ as 1

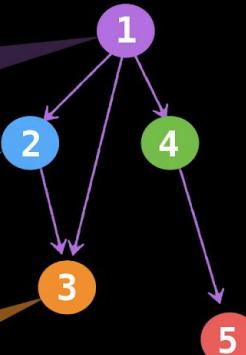
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!



suppose $H(1) = QmW98pJrc6FZ$

We can resolve:

/ipfs/QmW98pJrc6FZ as 1

/ipfs/QmW98pJrc6FZ/foo as 2

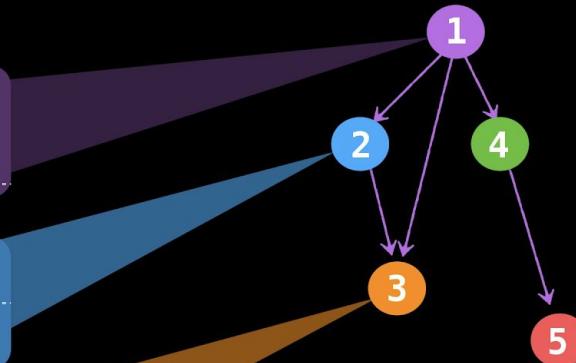
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!



suppose $H(1) = QmW98pJrc6FZ$

We can resolve:

- | | |
|------------------------|------|
| /ipfs/QmW98pJrc6FZ | as 1 |
| /ipfs/QmW98pJrc6FZ/foo | as 2 |
| /ipfs/QmW98pJrc6FZ/bar | as 3 |

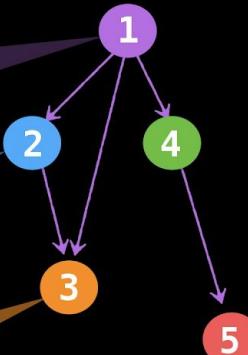
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!



suppose $H(1) = QmW98pJrc6FZ$

We can resolve:

- /ipfs/QmW98pJrc6FZ as 1
- /ipfs/QmW98pJrc6FZ/foo as 2
- /ipfs/QmW98pJrc6FZ/bar as 3
- /ipfs/QmW98pJrc6FZ/foo/biff as 3

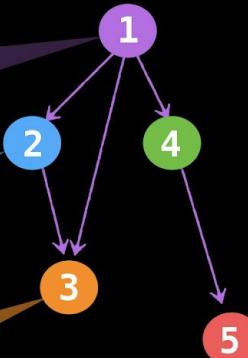
can resolve unix-style paths across dag links

suppose links:

QmbTJW4iGGBS 3987 foo
QmbTQmPMEFgh 1020 bar
QmZFJguywFUY 6787 baz

QmbTQmPMEFgh 1020 biff
this node has some data...

this node also has some data. but no links!

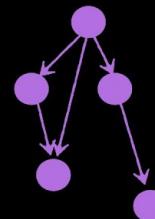


suppose $H(1) = QmW98pJrc6FZ$

We can resolve:

- /ipfs/QmW98pJrc6FZ as 1
- /ipfs/QmW98pJrc6FZ/foo as 2
- /ipfs/QmW98pJrc6FZ/bar as 3
- /ipfs/QmW98pJrc6FZ/foo/biff as 3
- /ipfs/QmW98pJrc6FZ/baz as 4

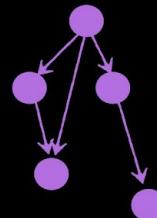
any data structures are
represented as dags



any data structures are represented as dags



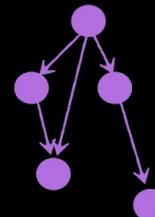
unix files
and dirs



any data structures are
represented as dags



unix files
and dirs

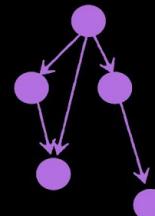


blob, tree, commit

any data structures are represented as dags



unix files
and dirs



blob, tree, commit

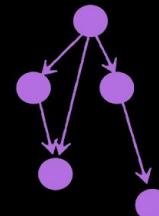


blocks, txns, wallets

any data structures are represented as dags



unix files
and dirs



kv-stores



blob, tree, commit

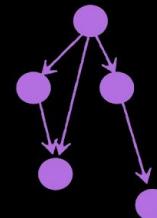


blocks, txns, wallets

any data structures are represented as dags



unix files
and dirs



kv-stores

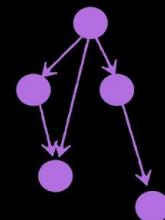


blob, tree, commit



blocks, txns, wallets

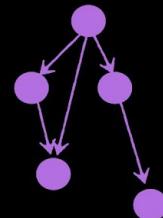
unixfs as a dag



unixfs as a dag



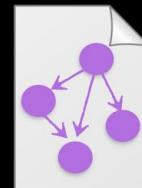
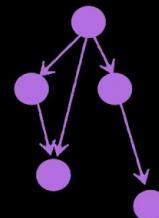
files are dag nodes



unixfs as a dag



files are dag nodes

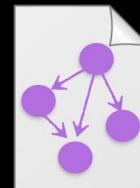
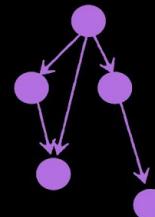


BIG files may be
split into many

unixfs as a dag



files are **dag nodes**



BIG files may be
split into many

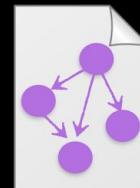
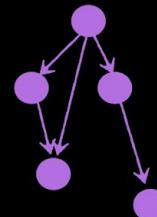


directories are
also **dag nodes**

unixfs as a dag



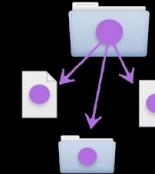
files are **dag nodes**



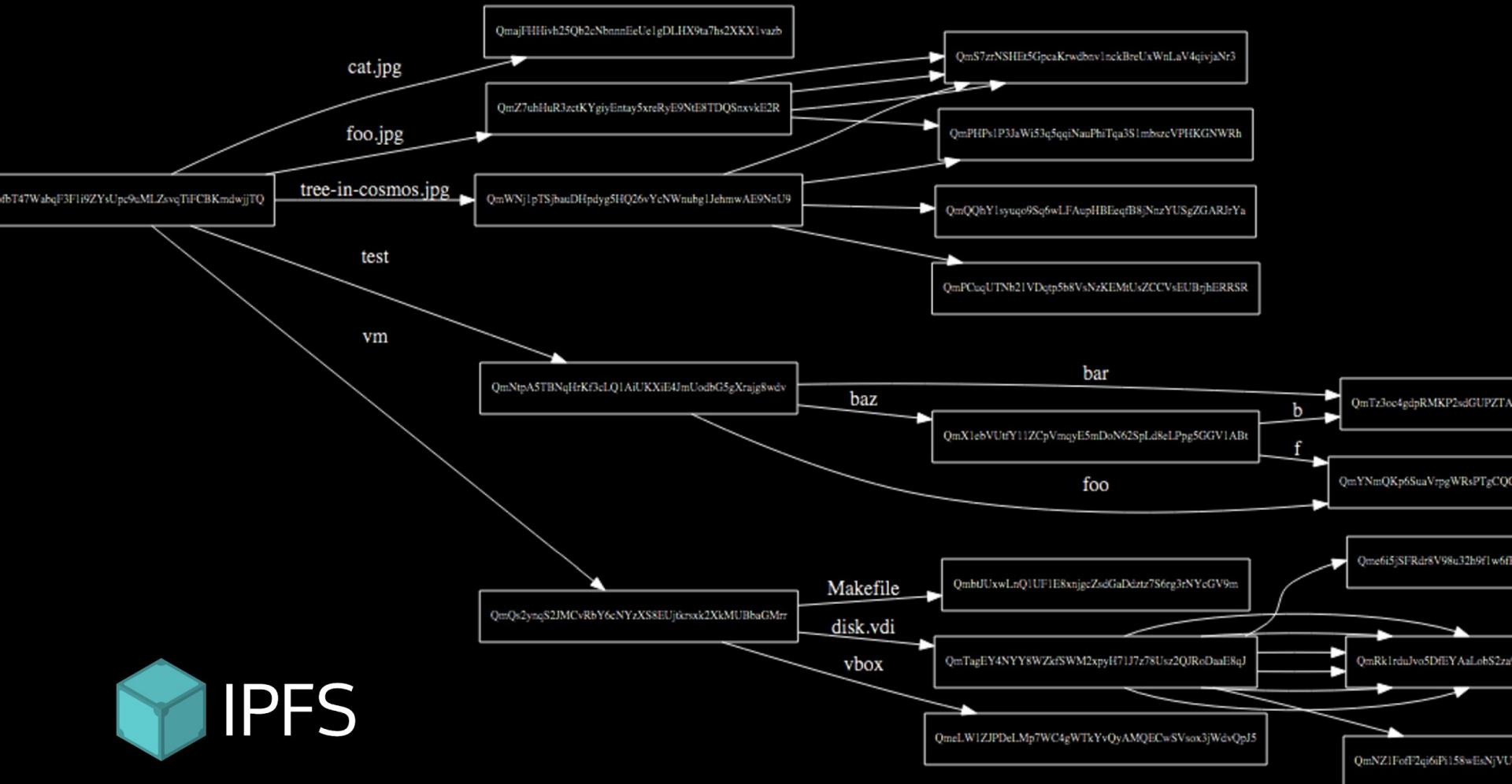
BIG files may be
split into many



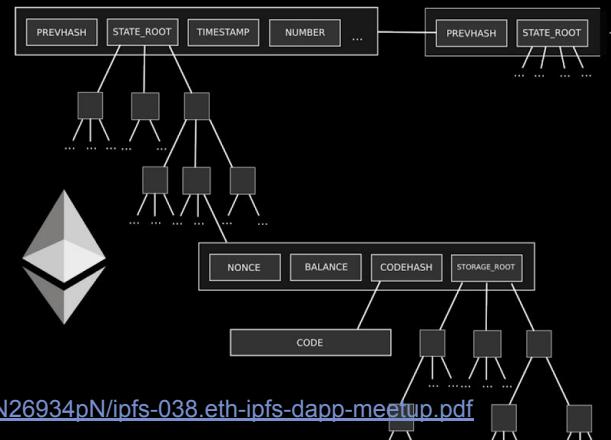
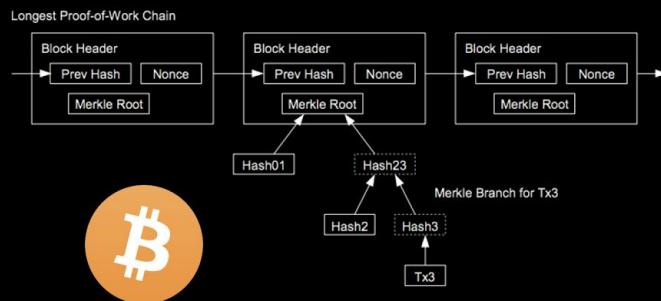
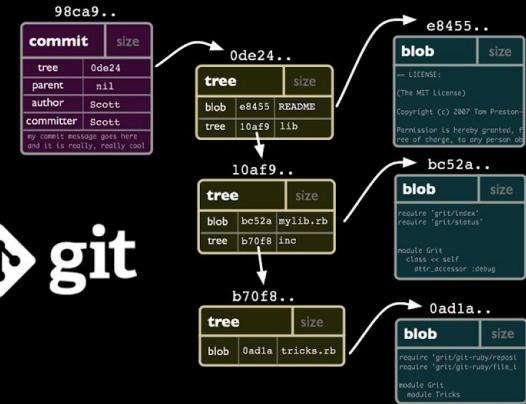
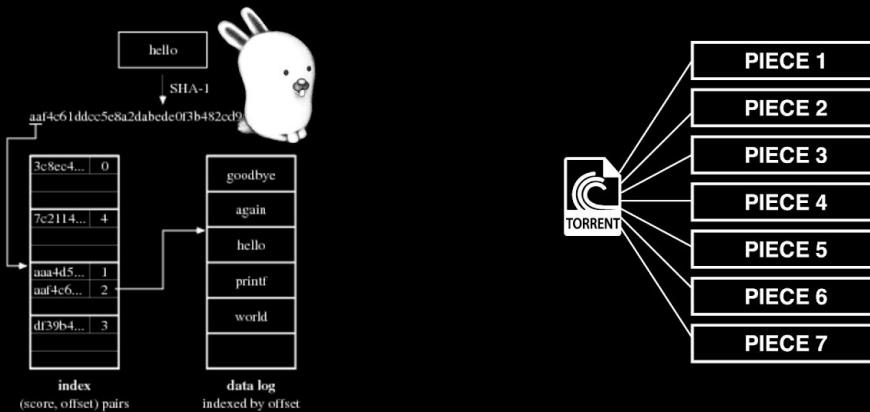
directories are
also **dag nodes**



which link
to others

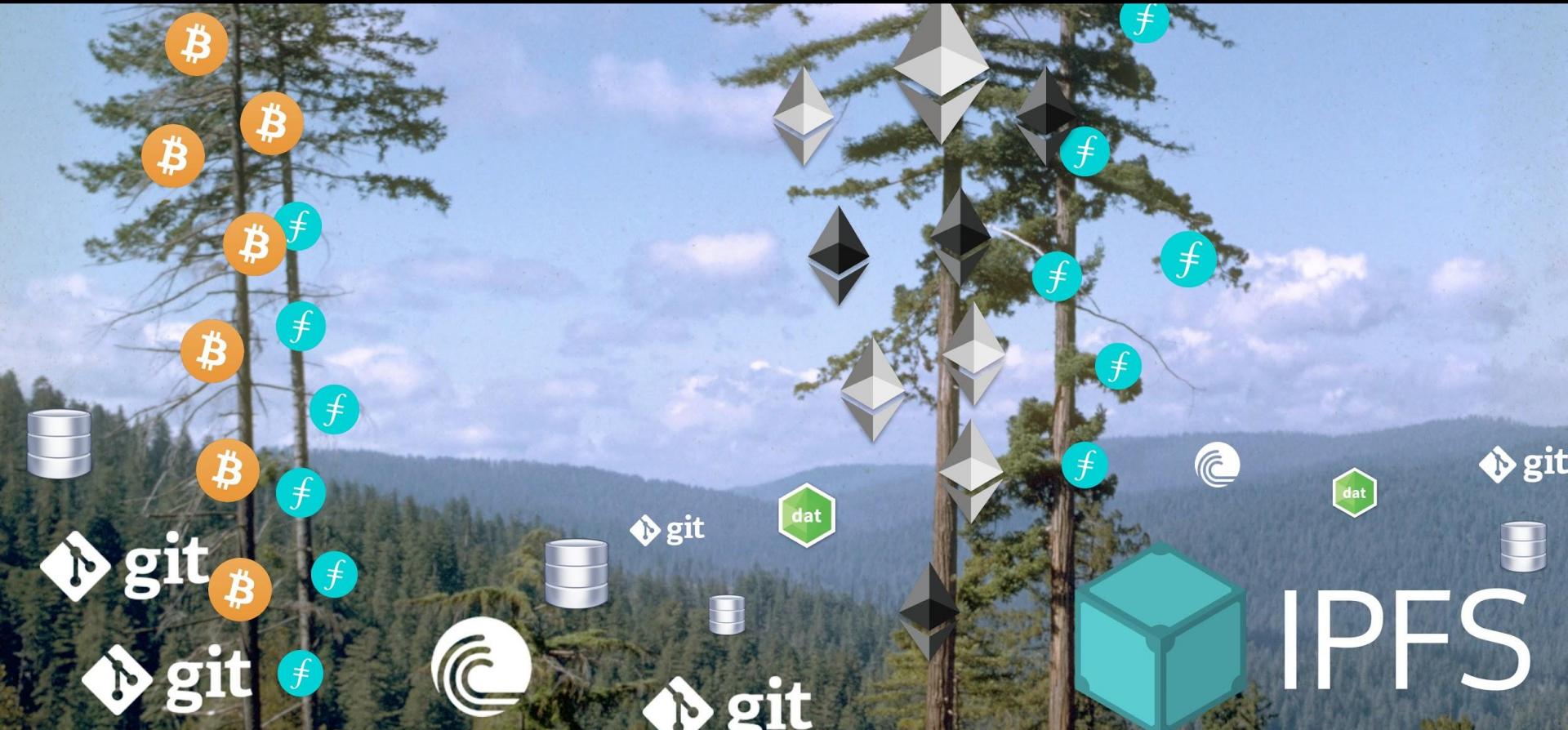


Source: <https://ipfs.io/ipfs/Qme51FQycp922BN83u8KXUYKJAxAxVAtPJbk1rxN26934pN/ipfs-038.eth-ipfs-dapp-meetup.pdf>



Source: <https://ipfs.io/ipfs/Qme51FQycp922BN83u8KXUYKJAxVAtPJbk1rxxN26934pN/ipfs-038.eth-ipfs-dapp-meetup.pdf>

IPFS is like a forest of linked merkle-trees



multiformats - self describing values

protocol agility, interop, avoid lock in

multihash - cryptographic hashes

multiaddr

multibase

multicodec

multistream

multikey

0x08e11fc41466fcda0af7dee0905605d9
e4aada4961542da952c8bb93080cc6f9

0x95a1b32bd70332e24f63f3802aae5f5e
1fa4622cc72750e0073bbbb6dcf6fce7

0xcaadb37a46daeda4e0d5e61574a9aaca
211d513806a026e6cc4461f7ba7867f9

0x08fbea061a5dea457d69fe5c12575c1d
9d30c49f575936f6e1c6d4ea0ab078df

256 0x08e11fc41466fcda0af7dee0905605d9
 e4aada4961542da952c8bb93080cc6f9

256 0x95a1b32bd70332e24f63f3802aae5f5e
 1fa4622cc72750e0073bbbb6dcf6fce7

256 0xcaadb37a46daeda4e0d5e61574a9aaca
 211d513806a026e6cc4461f7ba7867f9

256 0x08fbea061a5dea457d69fe5c12575c1d
 9d30c49f575936f6e1c6d4ea0ab078df

sha2 256 256 0x08e11fc41466fcda0af7dee0905605d9
e4aada4961542da952c8bb93080cc6f9

sha2 512 256 0x95a1b32bd70332e24f63f3802aae5f5e
1fa4622cc72750e0073bbbb6dcf6fce7

sha3 256 0xcaadb37a46daeda4e0d5e61574a9aaca
211d513806a026e6cc4461f7ba7867f9

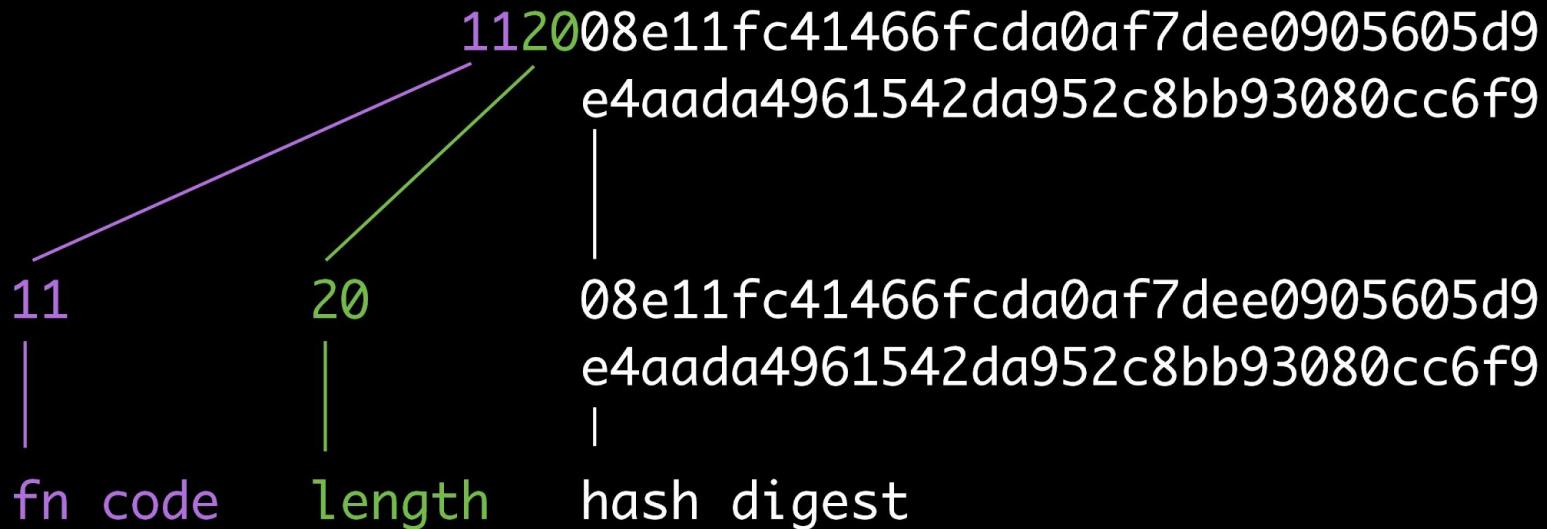
blake2b 256 0x08fbea061a5dea457d69fe5c12575c1d
9d30c49f575936f6e1c6d4ea0ab078df

sha2 256 256 112008e11fc41466fcda0af7dee0905605d9
e4aada4961542da952c8bb93080cc6f9

sha2 512 256 122095a1b32bd70332e24f63f3802aae5f5e
1fa4622cc72750e0073bbbb6dcf6fce7

sha3 256 1420caadb37a46daeda4e0d5e61574a9aca
211d513806a026e6cc4461f7ba7867f9

blake2b 256 402008fbea061a5dea457d69fe5c12575c1d
9d30c49f575936f6e1c6d4ea0ab078df



Multibase Table v1.0.0-RC (semver)

The current multibase table is [here](#):

encoding	codes	name
base1	1	unary tends to be 11111
base2	0	binary has 1 and 0
base8	7	highest char in octal
base10	9	highest char in decimal
base16	F, f	highest char in hex
base32	B, b	rfc4648 - no padding - highest letter
base32pad	C, c	rfc4648 - with padding
base32hex	V, v	rfc4648 - no padding - highest char
base32hexpad	T, t	rfc4648 - with padding
base32z	h	z-base-32 - used by Tahoe-LAFS - highest letter
base58flickr	Z	highest char
base58btc	z	highest char
base64	m	rfc4648 - no padding
base64pad	M	rfc4648 - with padding - MIME encoding
base64url	u	rfc4648 - no padding
base64urlpad	U	rfc4648 - with padding

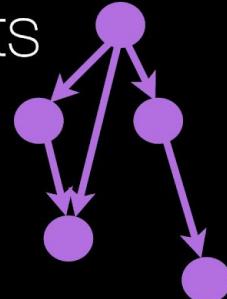
These encodings are being considered:

base128		
base256ascii	X	ascii
base-emoji	😊	base emoji
base65536	↳	base65536
utf8		
utf16		

CID: Content IDentifier



- **CID** is a format for hash-links (merkle-links)
- Uses **Multihash** for multiple hash fn support
- Uses **Multibase** for multiple encodings
- Uses **Multicodec** for multiple serialization formats





IPRS

DNS

is used for all sorts of things

A	@	104.236.176.52
A	disabled-dev0.4.0	104.236.179.241
AAAA	disabled-dev0.4.0	fc98:424c:b433:d7e2:7ee
AAAA	h	fce3:c53b:c3c5:2f54:8bb
MX	1	aspmx.l.google.com.
MX	5	alt1.aspmx.l.google.com.
TXT	@	dnslink=/ipfs/QmTgNJEg(
TXT	h	dnslink=/ipfs/QmcQBvKTI
NS	ns1.digitalocean.com.	
NS	ns2.digitalocean.com.	

DNS

is used for all sorts of things

type: A
sub: subdomain
value: 104.236.176.52
TTL: 180s

A	@	104.236.176.52
A	disabled-dev0.4.0	104.236.179.241
AAAA	disabled-dev0.4.0	fc98:424c:b433:d7e2:7ee
AAAA	h	fce3:c53b:c3c5:2f54:8bb
MX	1	aspmx.l.google.com.
MX	5	alt1.aspmx.l.google.com.
TXT	@	dnslink=/ipfs/QmTgNJEG
TXT	h	dnslink=/ipfs/QmcQBvKTI
NS	ns1.digitalocean.com.	
NS	ns2.digitalocean.com.	

IPRS

is used for all
sorts of things

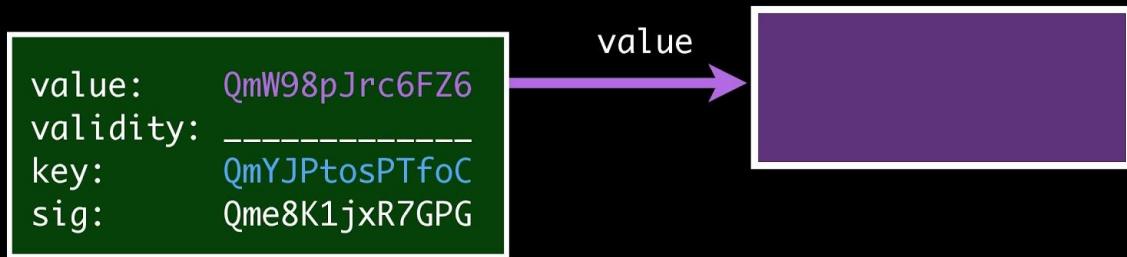
```
value:      QmW98pJrc6FZ6
validity:   _____
key:        QmYJPtosPTfoC
sig:        Qme8K1jxR7GPG
```

IPRS

is used for all
sorts of things

Records

are just IPLD objects
that link to other objects

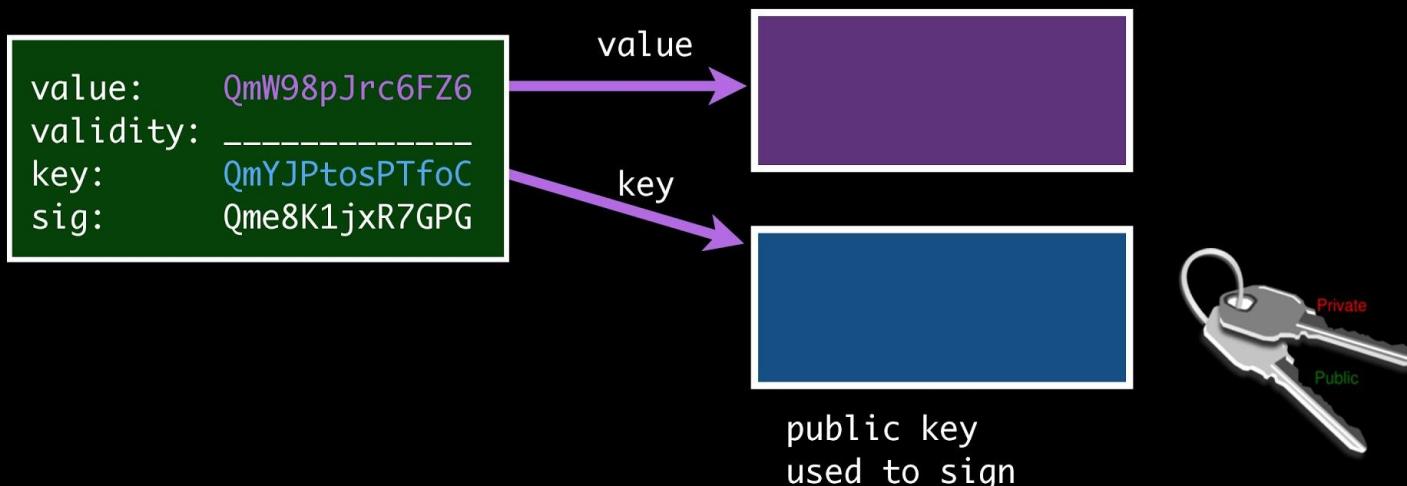


IPRS

is used for all
sorts of things

Records

are just IPLD objects
that link to other objects





IPNS

MAZIERES LINK

mutability

key name	/ipns/QmYJPtosPTfoC/foo/bar/baz.png
content addr	/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png
	fs:/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png
	ipfs:/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png



Escaping the Evils of Centralized Control with self-certifying pathnames

David Mazières and M. Frans Kaashoek

`dm@lcs.mit.edu, kaashoek@lcs.mit.edu`

MIT Laboratory for Computer Science

545 Technology Square, Cambridge MA 02139

Abstract

People have long trusted central authorities to coordinate secure collaboration on local-area networks. Unfortunately, the Internet doesn't provide the kind of administrative structures individual organizations do. As such, users risk painful consequences if global, distributed systems rely on central authorities for security. Fortunately, security need not come at the price of centralized control. To prove it, we present SFS, a secure, global, decentralized file system permitting easy cross-administrative realm collaboration. With a simple idea, self-certifying pathnames, SFS lets users escape the evils of centralized control.

Source: <https://qconf.com/sf2015/system/files/presentation-slides/ips-018.qconf.pdf>

/ipns/QmYJPtosPTfoC/foo/bar/baz.png



/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png



QmYJPtosPTfoC

QmW98pJrc6FZ6

/ipns/QmYJPtosPTfoC/foo/bar/baz.png



/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png



QmW98pJrc6FZ6



QmYJPtosPTfoC

QmW98pJrc6FZ6

/ipns/QmYJPtosPTfoC/foo/bar/baz.png



/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png

public
key

secret
key



QmW98pJrc6FZ6
signed by
QmYJPtosPTfoC



QmYJPtosPTfoC

QmW98pJrc6FZ6



/ipns/QmYJPtosPTfoC/foo/bar/baz.png



/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png

public
key

secret
key



QmW98pJrc6FZ6
signed by
QmYJPtosPTfoC

QmYJPtosPTfoC



QmW98pJrc6FZ6



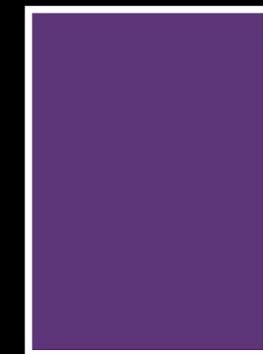
/ipns/QmYJPtosPTfoC/foo/bar/baz.png



/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png



value:	QmW98pJrc6FZ6
key:	QmYJPtosPTfoC
validity:	-----
sig:	Qme8K1jxR7GPG



QmYJPtosPTfoC

IPRS Record

QmW98pJrc6FZ6

human readability

dns name	/dns/ example.com /foo/bar/baz.png /ipns/ example.com /foo/bar/baz.png
key name	 /ipns/QmYJPtosPTfoC/foo/bar/baz.png
content addr	 /ipfs/QmW98pJrc6FZ6/foo/bar/baz.png fs:/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png ipfs:/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png

DNS Links

```
jbenet@earth: ~ > dig TXT ipfs.io

; <>> DiG 9.8.3-P1 <>> TXT ipfs.io
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37227
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ipfs.io.      IN TXT

;; ANSWER SECTION:
ipfs.io.    120  IN TXT  "dnslink=/ipfs/QmTgNJEgQaCqRht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"

;; Query time: 169 msec
;; SERVER: 209.244.0.3#53(209.244.0.3)
;; WHEN: Tue Nov 17 16:44:28 2015
;; MSG SIZE  rcvd: 98
```

DNS Links

```
jbenet @ earth : ~ > dig TXT ipfs.io

; <>> DiG 9.8.3-P1 <>> TXT ipfs.io
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37227
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ipfs.io.      IN TXT

;; ANSWER SECTION:
ipfs.io.      120  IN TXT  "dnslink=/ipfs/QmTgNJEgQaCqRht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"

;; Query time: 169 msec
;; SERVER: 209.244.0.3#53(209.244.0.3)
;; WHEN: Tue Nov 17 16:44:28 2015
;; MSG SIZE  rcvd: 98
```

DNS Links



3. jbenet@lorien ~ (zsh)

```
jbenet @ earth : ~ > dig TXT ipfs.io
```

```
ipfs.io.    120  IN TXT  "dnslink=/ipfs/QmTgNJEgQaCqRht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"
```

DNS Links

```
jbenet@earth: ~ > dig TXT ipfs.io  
ipfs.io.    120  IN TXT  "dnslink=/ipfs/QmTgNJEgQaCqRht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"
```

dns name /dns/ipfs.io/foo/bar/baz.png
 /ipns/ipfs.io/foo/bar/baz.png



content addr /ipfs/QmW98pJrc6FZ6/foo/bar/baz.png

DNS Links

```
jbenet@earth: ~ > dig TXT ipfs.io  
ipfs.io.    120  IN TXT  "dnslink=/ipfs/QmTgNJEgQaCqRht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"
```

dns name	/dns/ipfs.io/foo/bar/baz.png /ipns/ipfs.io/foo/bar/baz.png	60s-5m
----------	---	--------



content addr	/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png
--------------	-------------------------------------



DNS Links

```
jbenet@earth: ~ > dig TXT ipfs.io  
ipfs.io.    120  IN TXT  "dnslink=/ipns/QmYJPtosPTfoCht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"
```

dns name /dns/ipfs.io/foo/bar/baz.png 60s-5m
/ipns/ipfs.io/foo/bar/baz.png



key name /ipns/QmYJPtosPTfoC/foo/bar/baz.png



content addr /ipfs/QmW98pJrc6FZ6/foo/bar/baz.png



DNS Links

```
jbenet@earth: ~ > dig TXT ipfs.io  
ipfs.io.    120  IN TXT  "dnslink=/ipns/QmYJPtosPTfoCht9KSXNyZsCp2xpHZmBRms28NRMmtcERp"
```

dns name	/dns/ipfs.io/foo/bar/baz.png /ipns/ipfs.io/foo/bar/baz.png	60s-5m
-----------------	---	--------



key name	/ipns/QmYJPtosPTfoC/foo/bar/baz.png	1ms-1s
-----------------	-------------------------------------	--------



content addr	/ipfs/QmW98pJrc6FZ6/foo/bar/baz.png	∞
---------------------	-------------------------------------	---



Blueprint

- Goal
- Why do we need IPFS?
- High level overview / Usage
- The IPFS Stack
- **Problems with IPFS**
- Cool Apps
- Some cool latest updates
- Demo



Problems

- Permanence does not mean Persistent!
- Illegal/Pirated content



Blueprint

- Goal
- Why do we need IPFS?
- High level overview / Usage
- The IPFS Stack
- Problems with IPFS
- **Cool Apps**
- Some cool latest updates
- Demo



Cool Apps

- [PeerPad](#): Collaborative editing tool
- [OpenBazaar](#): Online marketplace
- [The Lab](#): Video hosting
- [IPFScloud](#): Cloud storage platform



Blueprint

- Goal
- Why do we need IPFS?
- High level overview / Usage
- The IPFS Stack
- Problems with IPFS
- Cool Apps
- **Some cool latest updates**
- Demo



Some cool latest update

- **Sweet IPFS:** InterPlanetary File System for Android
- **IPFS Pinbot:** Add content and pin things using tweets
- **ipfs.wikileaks.org:** Experimental IPFS gateway for WikiLeaks



Blueprint

- Goal
- Why do we need IPFS?
- High level overview / Usage
- The IPFS Stack
- Problems with IPFS
- Cool Apps
- Some cool latest updates
- **Demo**



References / Things I read and watched (1)

- IPFS Paper:
<https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf>
- IPFS Reddit: <https://www.reddit.com/r/ipfs/>
- IPFS Discussion Forum: <https://discuss.ipfs.io/>
- IPFS Blog: <https://blog.ipfs.io/>
- Resolution and Routing in IPFS: <https://github.com/ipfs/faq/issues/48>
- IPFS Presentation 1:
<https://speakerd.s3.amazonaws.com/presentations/ecffee895940481a8f9aa8416806a13f/ipfs-005.2.pdf>
- IPFS Presentation 2:
<https://www.w3.org/2016/04/blockchain-workshop/slides/benet-ipfs.pdf>
- IPFS Presentation 3:
<https://ipfs.io/ipfs/Qme51FQycp922BN83u8KXUYKJAxVAtPJbk1rxxN26934pN/ipfs-038.eth-ipfs-dapp-meetup.pdf>
- IPFS Presentation 4:
<https://qconsf.com/sf2015/system/files/presentation-slides/ipfs-018.qconsf.pdf>



References / Things I read and watched (2)

- IPFS talk at Stanford: <https://www.youtube.com/watch?v=HUVmypx9HGI>
- Peer-to-Peer algorithms: <https://www.youtube.com/watch?v=kXyVqk3EbwE>
- IPFS Blockchain Fun (demo): <https://www.youtube.com/watch?v=bi-4YGZXxwA>
- IPLD: The Merkle Forest: https://www.youtube.com/watch?v=Bqs_LzBiQyk
- Github MerkleDAG: <https://www.youtube.com/watch?v=ronoCeMzfJ4>
- Intro to libp2p: https://www.youtube.com/watch?v=CRe_oDtfRLw
- DHTs at GTech: <https://www.youtube.com/watch?v=nCLtfUjAVD4>
- IPFS Alpha Demo: <https://www.youtube.com/watch?v=8CMxDNuuAiQ>
- BitSwap Coffee Talk: https://www.youtube.com/watch?v=9UjqJTCg_h4