

THE LIFECYCLE OF DATA IN THE DWEB

CORE COURSE D



Oli Evans

IPFS GUI



Michelle Hertzfeld

IPFS Cluster



Kishan Sagathiya

IPFS Cluster

GOAL

Understand how
IPFS manages data

What happens after you add a file

Don't know your DAG from your Datastore?

It's all covered in Core Course A...



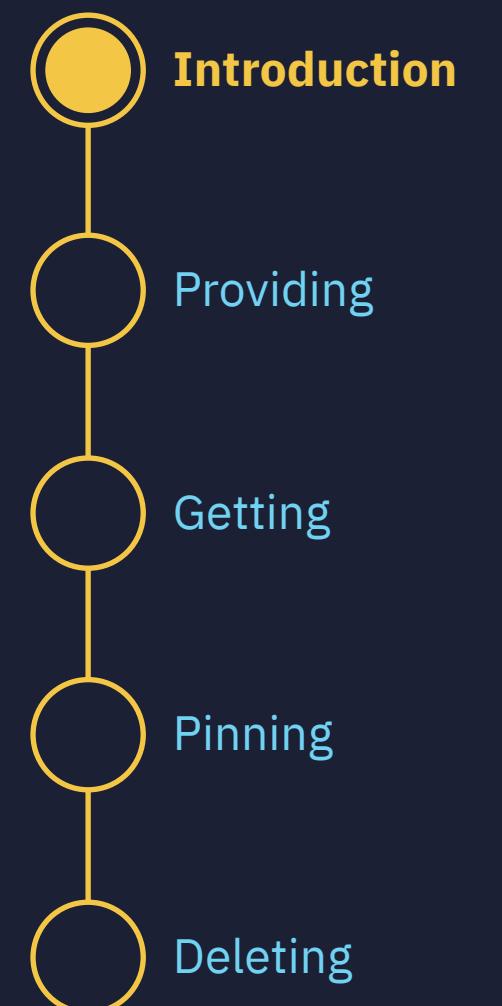
A file you add to IPFS is
stored as **blocks** in your
local repo.



Providing



I have these blocks

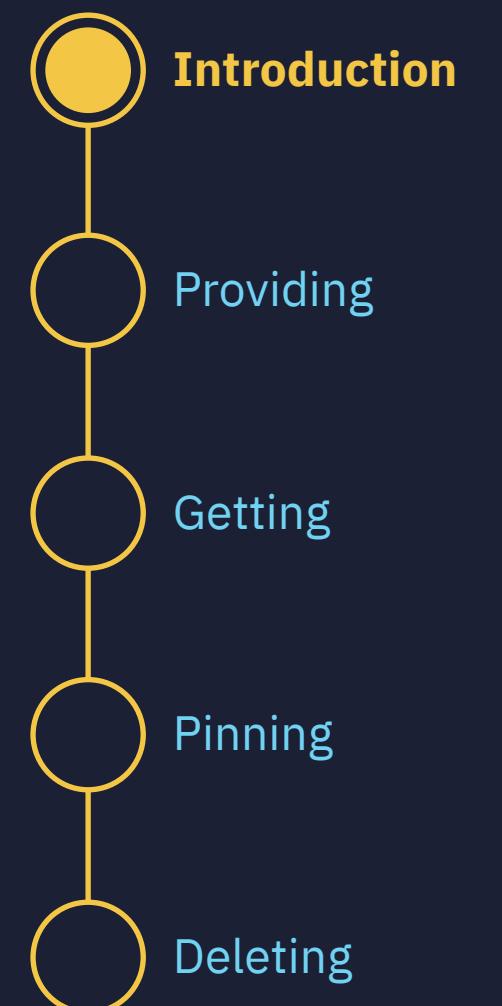




Getting

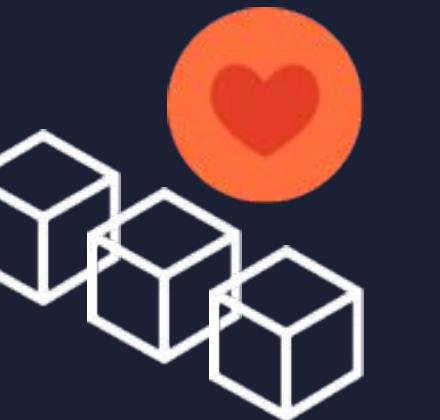


I want those blocks

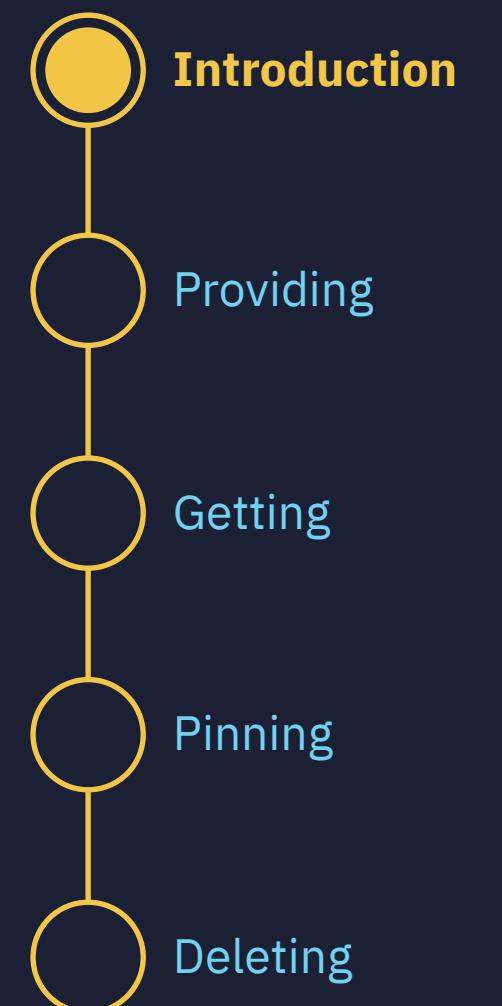




Pinning



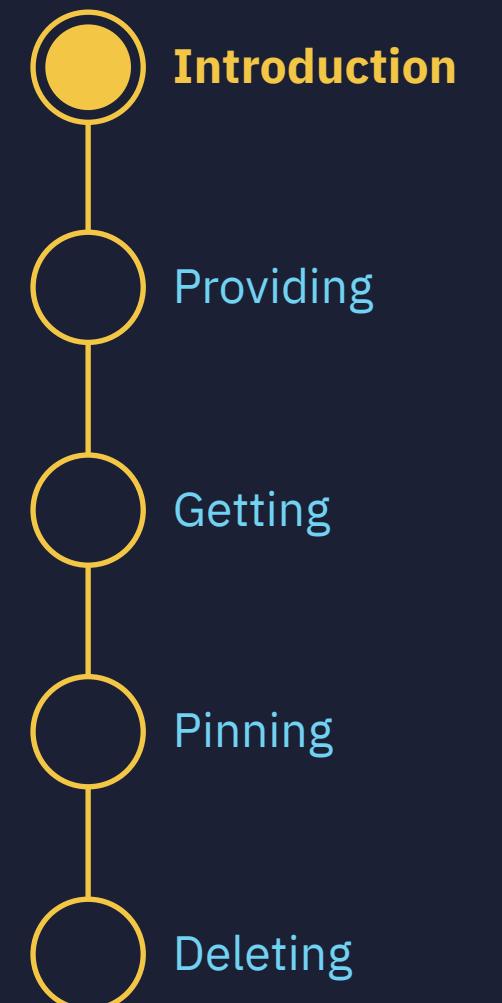
Keep these blocks





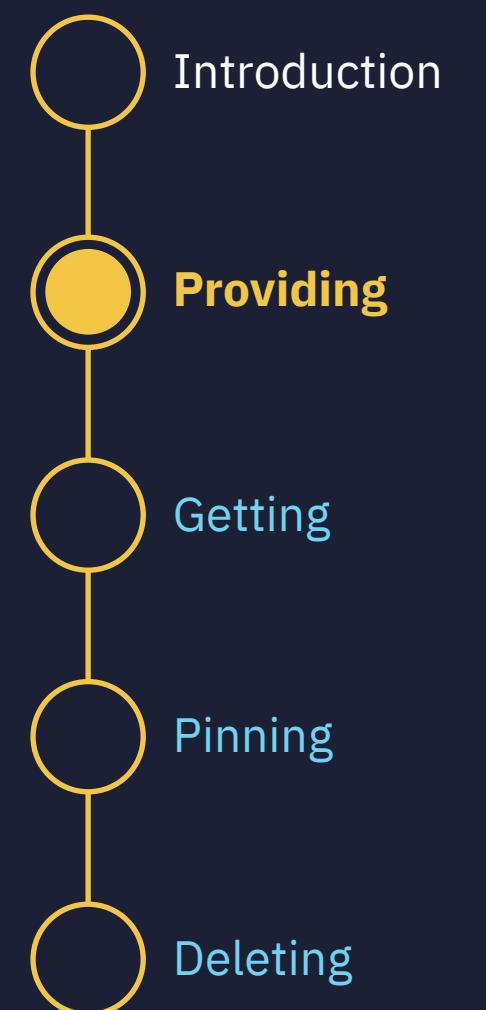
Deleting

DO NOT WANT!





Providing

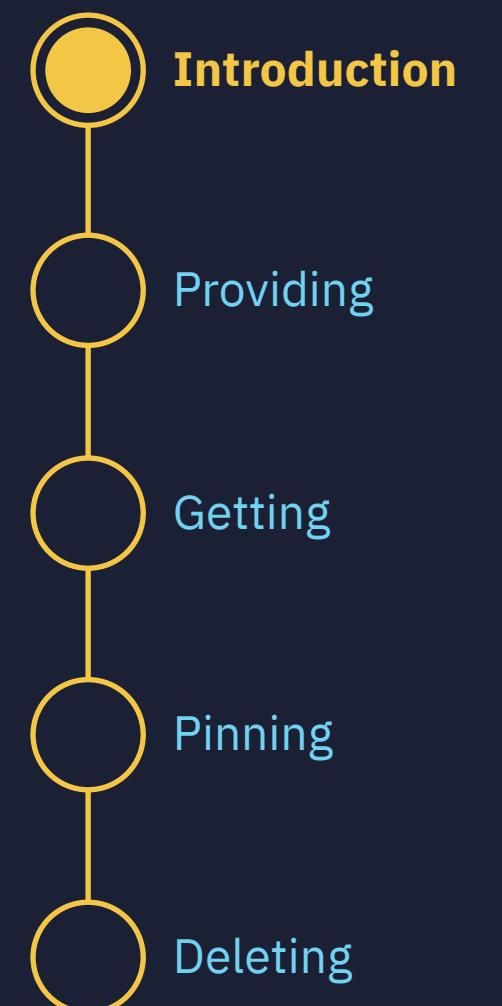




Providing

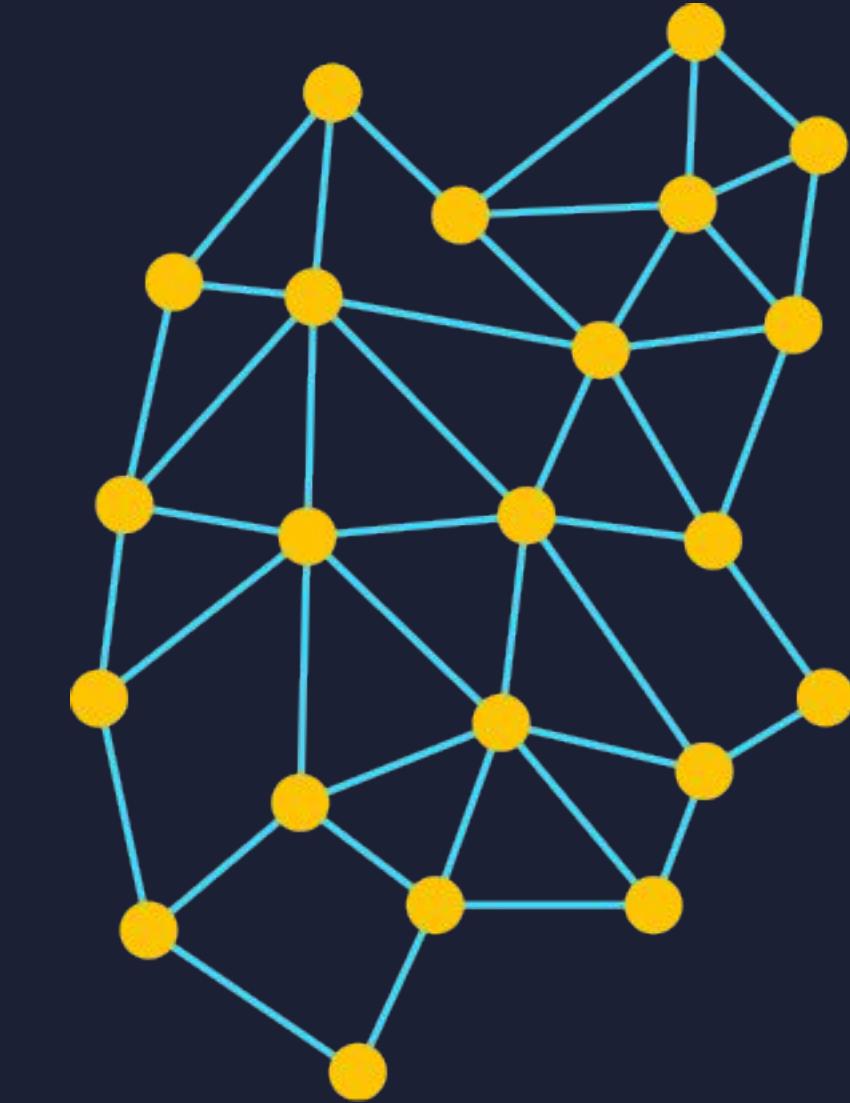


I have these blocks



Each block has a **CID**,
the content-address of
it's data.

CIDs provide Distributed Naming



The steps for creating a CID allow multiple users to agree on names for blocks without asking a central authority

Content-addressing
lets us separate

what we want

&

where we get it

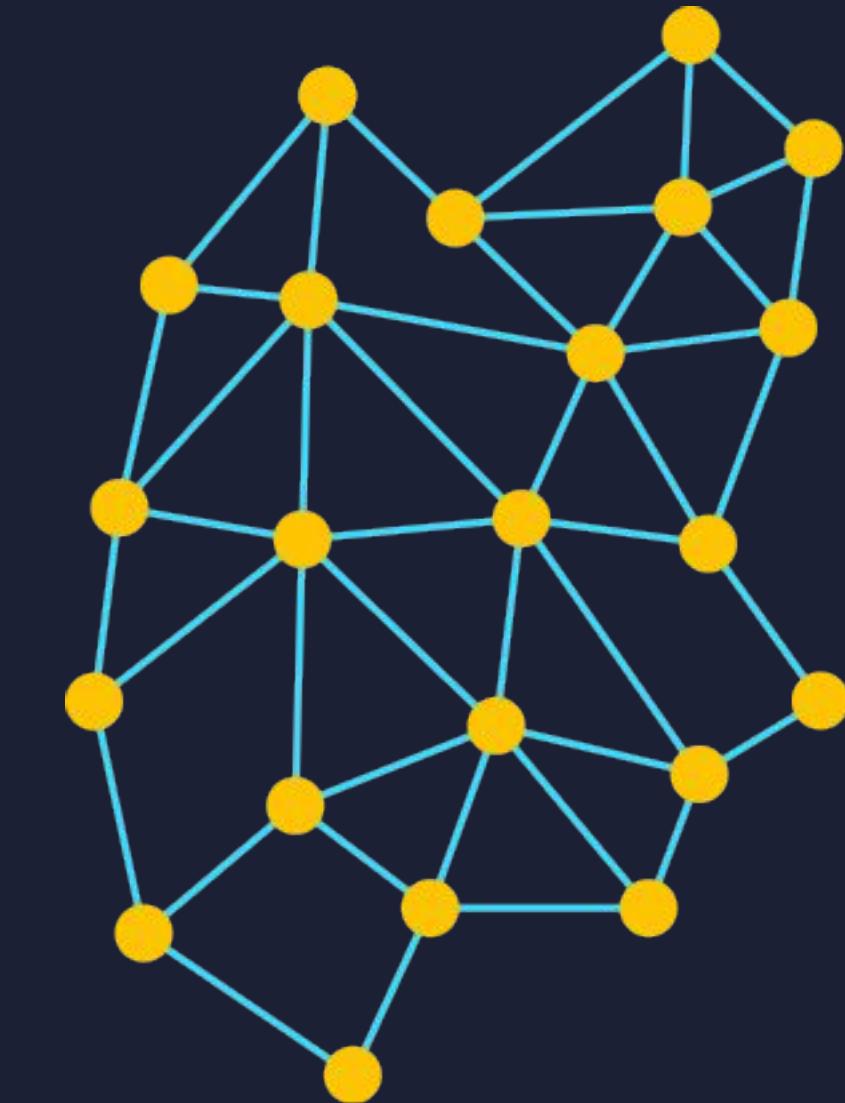
Challenge

How do we locate content-addressed data?



Kademlia

Kademlia provides Distributed Discovery

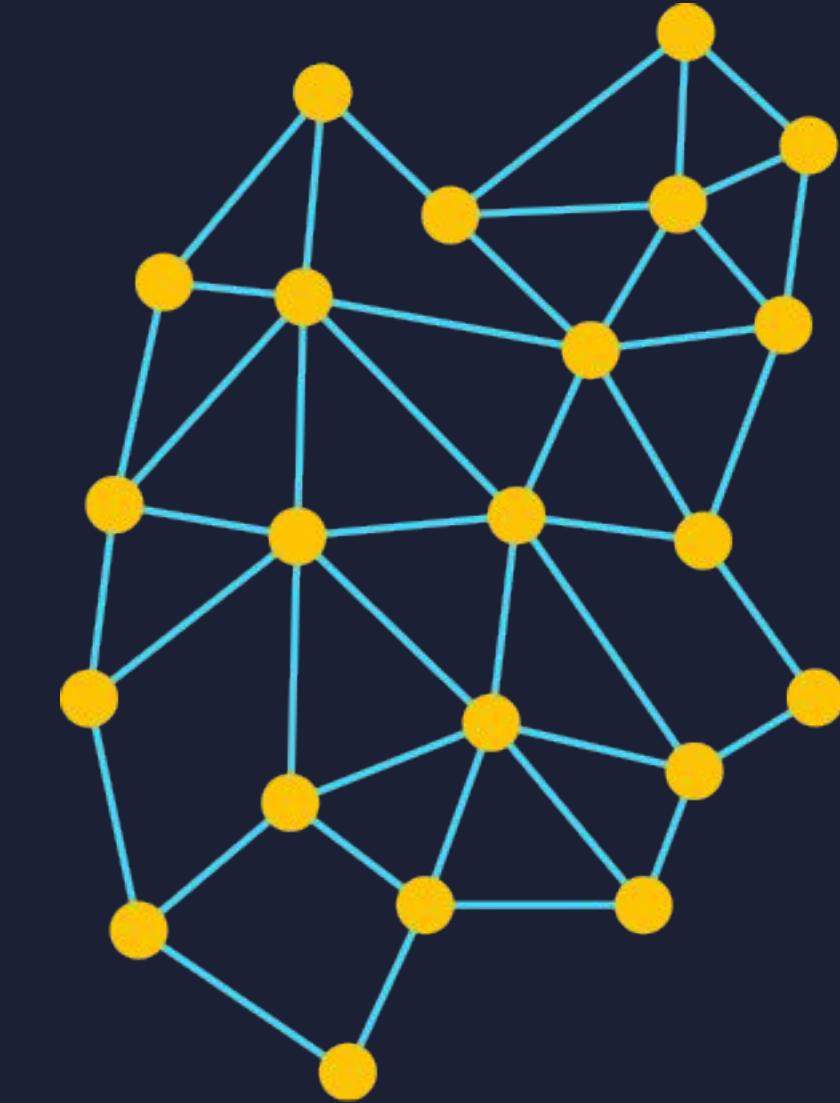


The Kademlia algorithm defines who to tell and who to ask about where to find **data**, over a changing network of peers.

Key	Value
 QmCheese	 QmZaphod

This **block**, I have it!

Kademlia is a Distributed Hash Table



A protocol for storing and retrieving key/value pairs stored across a changing network of peers

Tell the peers with PeerIDs
most similar to the CID

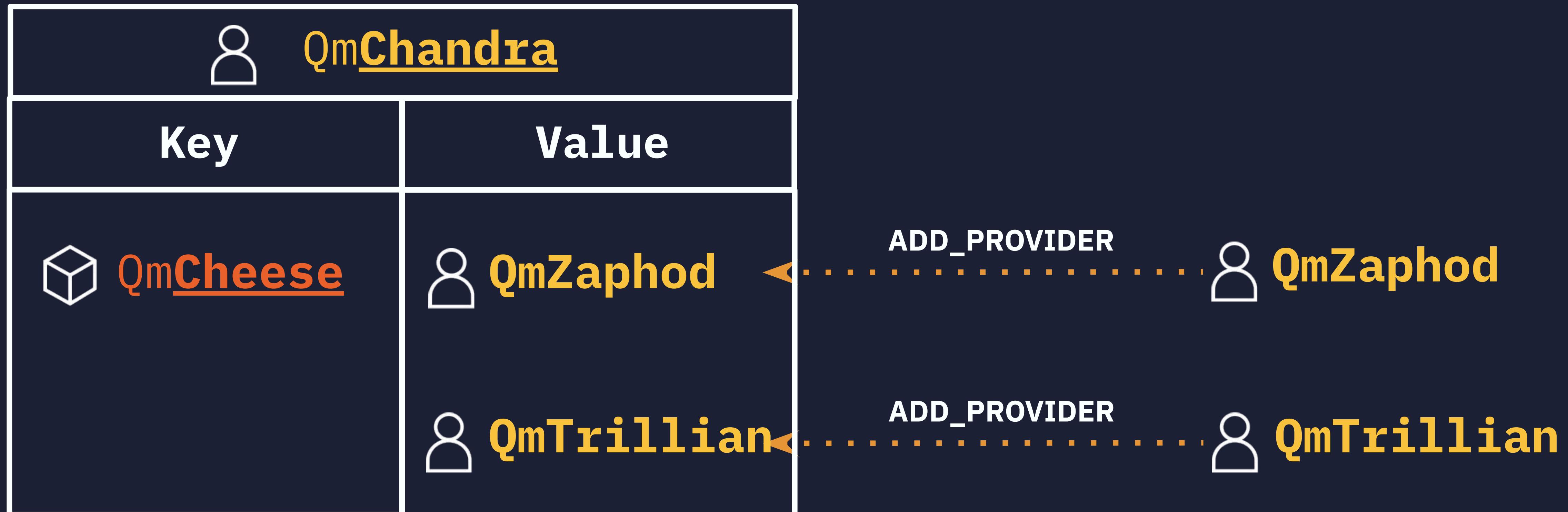




Multiple Providers



Multiple Providers





QmBrian



QmBread

Would you like to know ~~XOR~~ more?

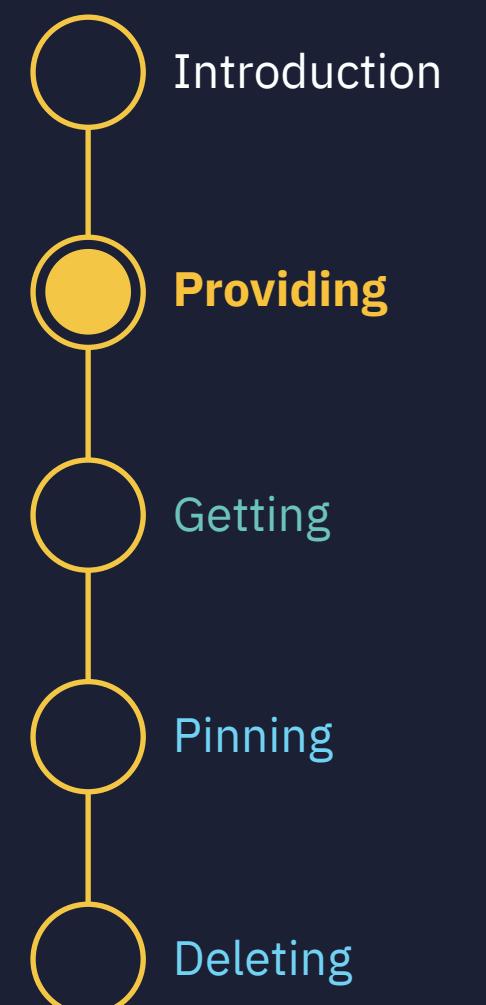
It's covered in Core Course B and the DHT Deep Dive



Reproviding 

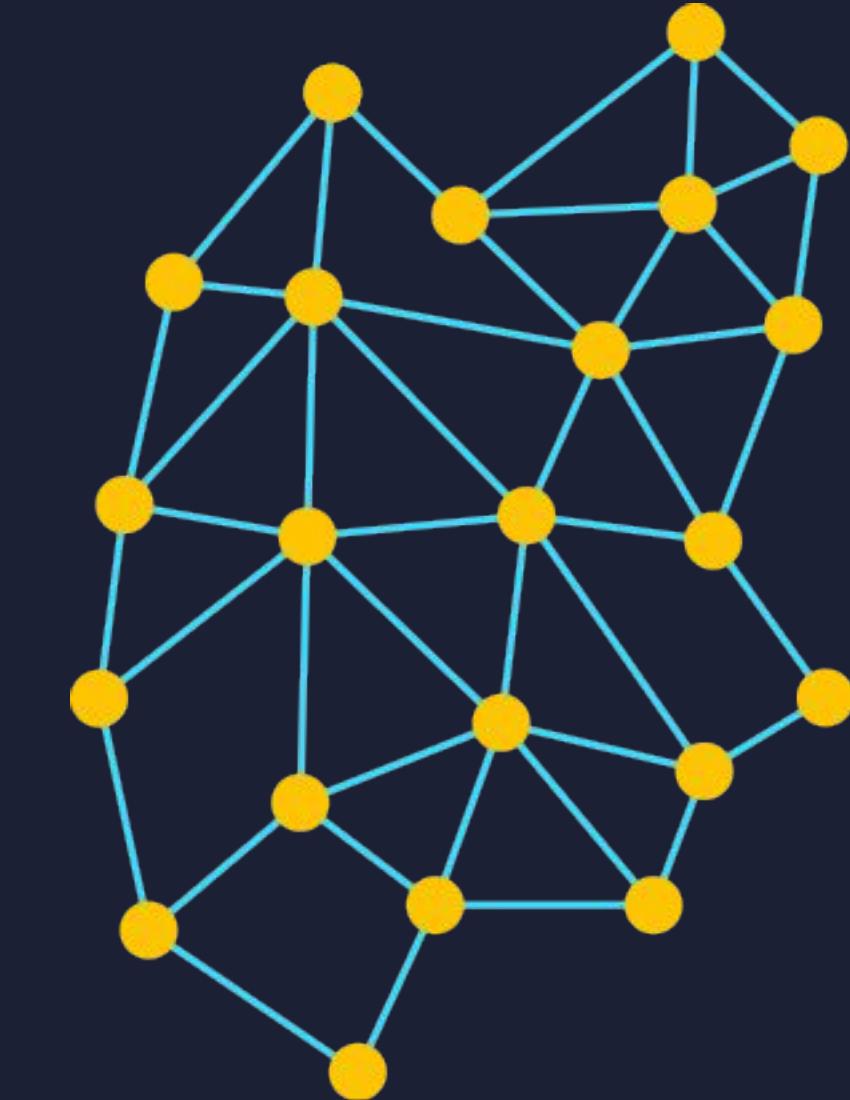


I still have these blocks



Provider strategies

Which CIDs to announce?



- all (default)
- roots (direct and recursive pins)
- pins (direct, recursive and indirect pins)

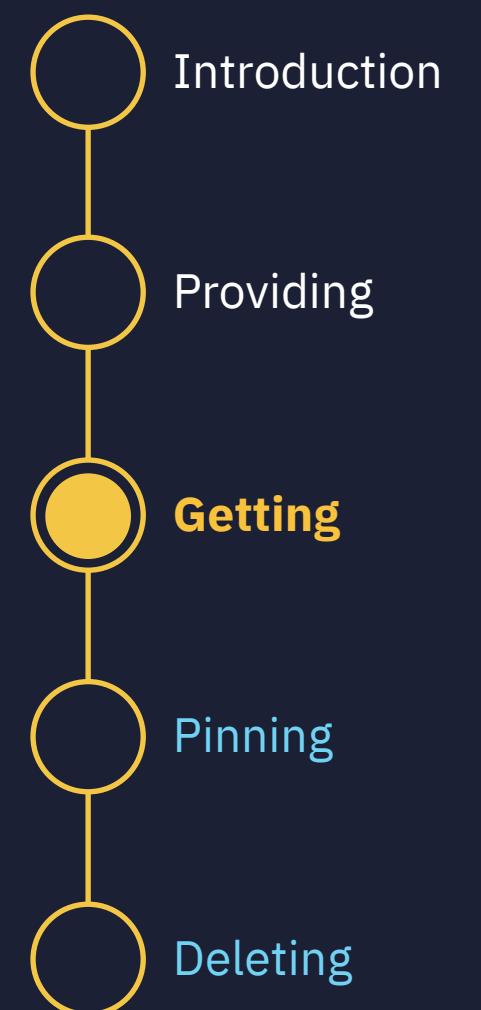


```
# cat ~/.ipfs/config | jq .Reprovider
{
  "Interval": "12h",
  "Strategy": "all"
}
```

```
$ ipfs dht findprovs QmfQkD8pBSBCBxWEwFSu4XaDVSWK6bjnNuaWZjMyQbyDub  
QmRbvzDMjRFxFk4afGryHb51nBh4Wtpz1ghrpxNT1adgC  
Qmbk4Z9E19GaStfYGYksFXyMd9S1KcagdGTJcWzuB8puni  
QmTA5k7QjmCkHtMBiTkh8qSxuPKwtYmfZeUfaji96nPJHU  
QmVNa1BHFdWmsocyB3FPwaE4ANEyu2q9NrJAf8YGo495MZ  
QmVtEhVygrxN2ZkpDY2NfukFGYPfHNY1C6KUQat1YDRTnT  
Qmbudnu9Vm8tvTMPNuqshuQiJUu48ASYBkJ67JfShc3CWj  
QmaC7Xz2pJWBYUFSQQYazmqqRLy54sDokPtQ2QShZLHuPG  
QmSTwGkyU9HADx7nucdTJa5xapAYu4sprrp96hrbTEoCmn  
QmVxcs1boYRjgKrrtx7Db1jGJtg8eXo1VvcTQjFZ7n19WG  
...
```



Getting





Getting



I want those blocks



Bitswap

Bitswap

Trading blocks

Bitswap has two jobs:

To request blocks you need from your connected peers

To send blocks you have to peers who want them



AND NOW FOR
SOMETHING

completely different

The Bitswap dance

A short physical performance

User A unwraps a cat gif and now has block QmCatGif

User B throws a ping pong ball wantlist at A and C with [QmCatGif] on it and shouts WANT!

User A throws a block for QmCatGif to User B

User C throws a ping pong ball want list for [QmCatGif] to B and C

Both A and B throw blocks at User C. (probably oli, the blocks hit him, hilarity ensues.)

User C drops 1 and keeps 1, they are identical. (hints at a overuse of bandwidth issue, see graphsync for possible optimisation)

The Wantlist

Each peer stores the list of CIDs they want to get

A Wantlist message

A list of CIDs you send as message to connected peers

A Blocks message

1 or more blocks prefixed with how to calculate their CIDs

Bitswap

Verified data

A block sent via bitswap is prefixed with the cid version, multicodec, and multihash

On receiving a block, the parameters are used to hash the data and calculate the CID.

If that CID is still on your Wantlist, you save it to your local repo.

If not, it's dropped.



- I want this CID.
- I want these CIDs...
- Repeat

Bitswap Sessions

At a protocol level, bitswap is simple.
It's message based and all messages
contain wantlists, or blocks



Bitswap Sessions add optimizations to
only send wantlists to peers most likely
to have the blocks you want.

Get Involved Make IPFS Faster

Bitswap sessions are an ongoing research and development effort.

You can help make IPFS faster and more efficient by getting involved, trying out the PRs and proposing new optimizations



graphsync □

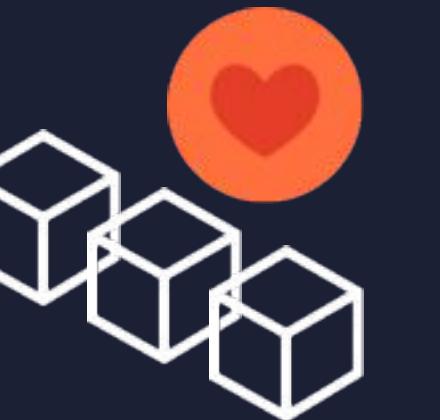


Pinning





Pinning



Keep these blocks



IPFS makes it feel like every block is local, there is no “*retrieve this file from a remote server*”

Pinning allows you to tell IPFS to always keep a given block locally



@whyrusleeping

Every **block** you **add**
or **get** from others
is cached in **your** repo

Garbage-collection

deletes all unpinned
blocks from **your** local
repo.

Demo

IPFS Block Party!



DAG builder visualization × +

localhost:1234

IPFS

Block party 🎉

1,024 byte chunks

Unpin all Run GC Reset

UnixFS File CID QmQjDCznvoZBgJuM4gwXrLLxWzSmUna07DS5WSEjW7fmHA 1756 bytes total 0 bytes data 38 links

This screenshot shows a web-based interface for visualizing the Data Availability Proof (DAG) of an IPFS file. The main area displays a 4x10 grid of colored squares, each representing a 1,024 byte chunk of the file. The colors used are teal, grey, and orange. The interface includes a header with the IPFS logo and a 'Block party' badge, a dropdown menu for chunk size, and buttons for 'Unpin all', 'Run GC', and 'Reset'. At the bottom, it provides metadata about the file: UnixFS File, CID (QmQjDCznvoZBgJuM4gwXrLLxWzSmUna07DS5WSEjW7fmHA), total size (1756 bytes), data size (0 bytes), and link count (38 links).

The **red** one is pinned “recursively”



```
⠚ ipfs pin ls
QmR3UaHqCPJ6AnU85BHoPM8s4wQxQMYnz6HHpDHYPLhB9N recursive
Qmbc46hVTfR4hA2KXhLXMTUVpUHKdywawrBDvg4rAJt1gn indirect
QmdvnZNsKid4Hg1K2JLZo7N79Pwv98cpI2hFh4C3WBhUjw indirect
```

Pin this **block**
and all **blocks** it links to
and all **blocks** they link to

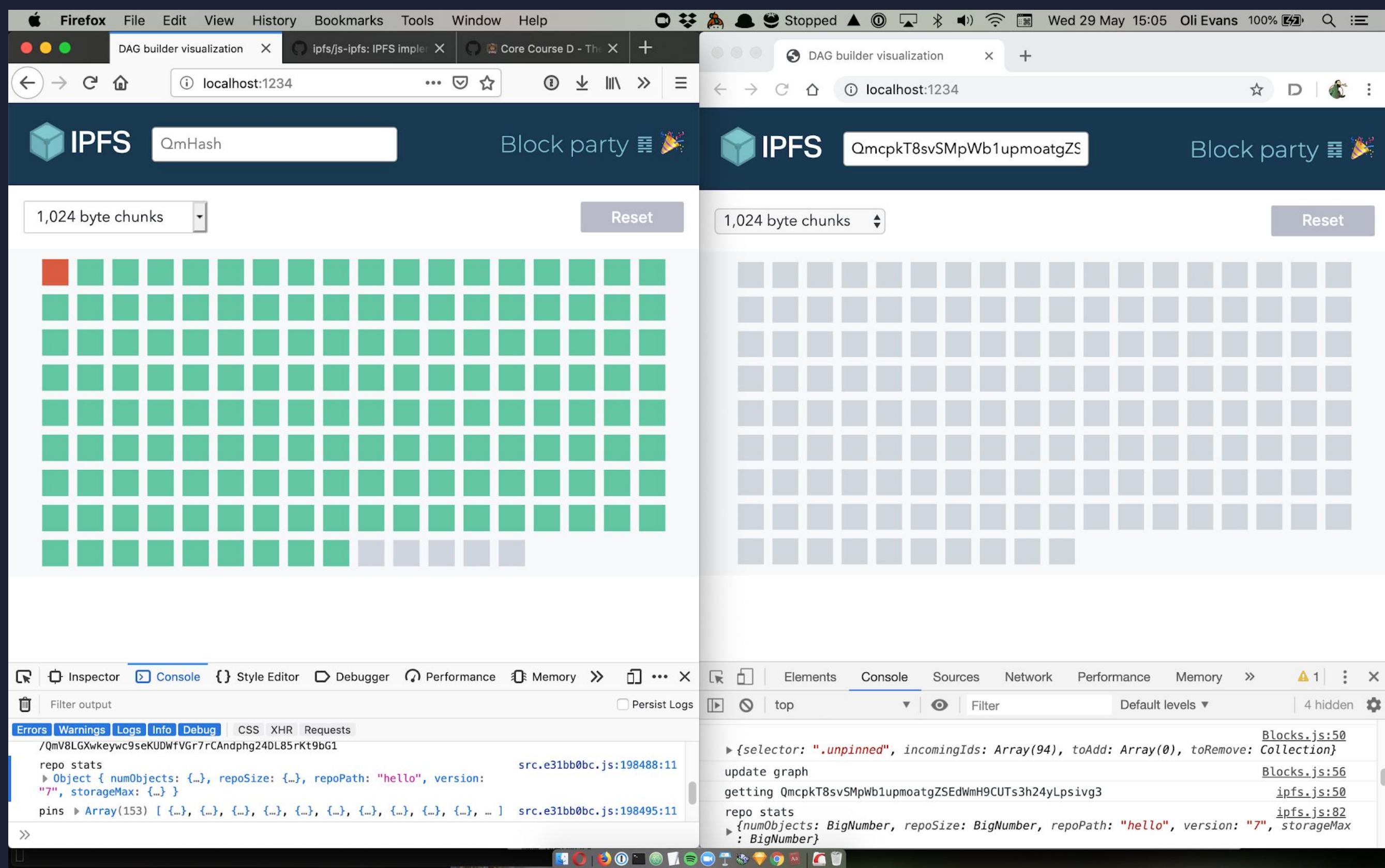
The **green** ones are pinned “indirectly”



```
⠚ ipfs pin ls
QmR3UaHqCPJ6AnU85BHoPM8s4wQxQMYnz6HHpDHYPLhB9N recursive
Qmbc46hVTfR4hA2KXhLXMTUVpUHKdywawrBDvg4rAJt1gn indirect
QmdvnZNsKid4Hg1K2JLZo7N79Pwv98cpI2hFh4C3WBhUjw indirect
```

The grey ones are cached
in your repo but not pinned

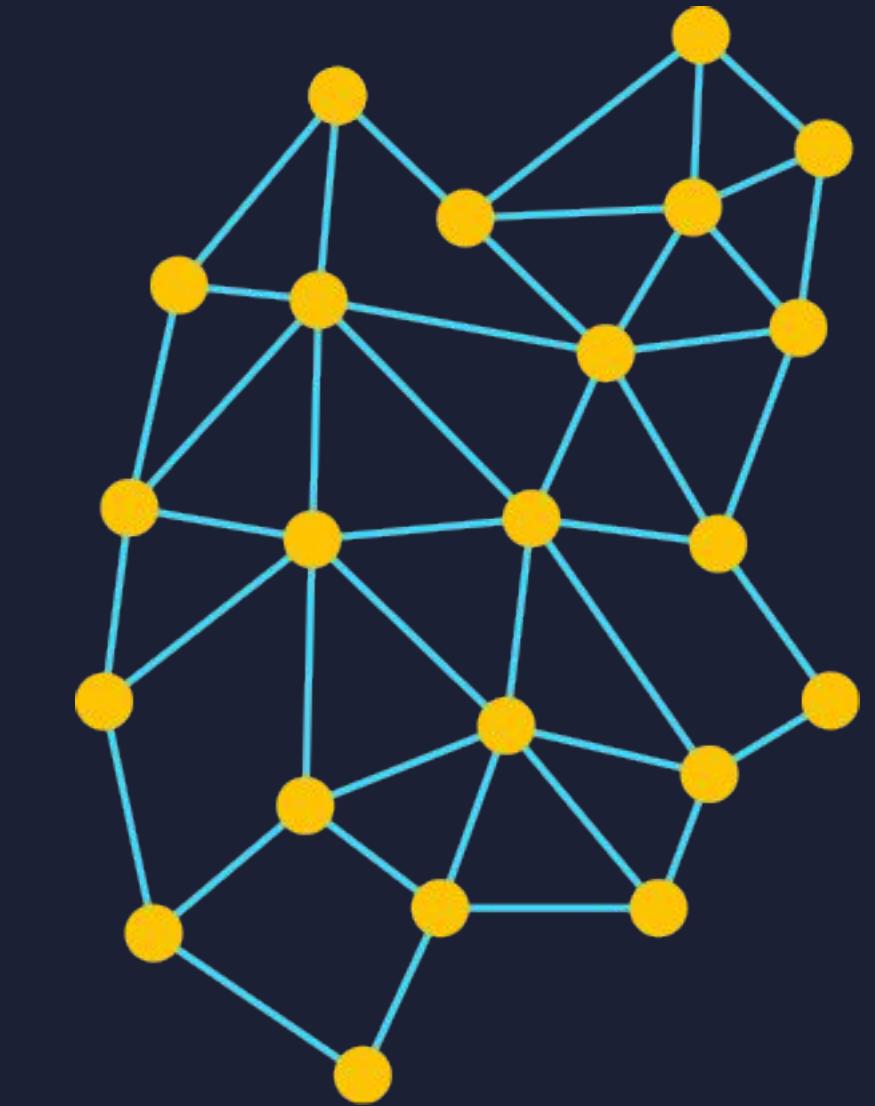
Why do i have unpinned blocks?



Reciprocal
co-hosting

Pinning is local!

Pinset



Metadata containing the list of unique CIDs pinned on your node. The pinset is stored in your local IPFS datastore.

Care about ***a lot*** of blocks?

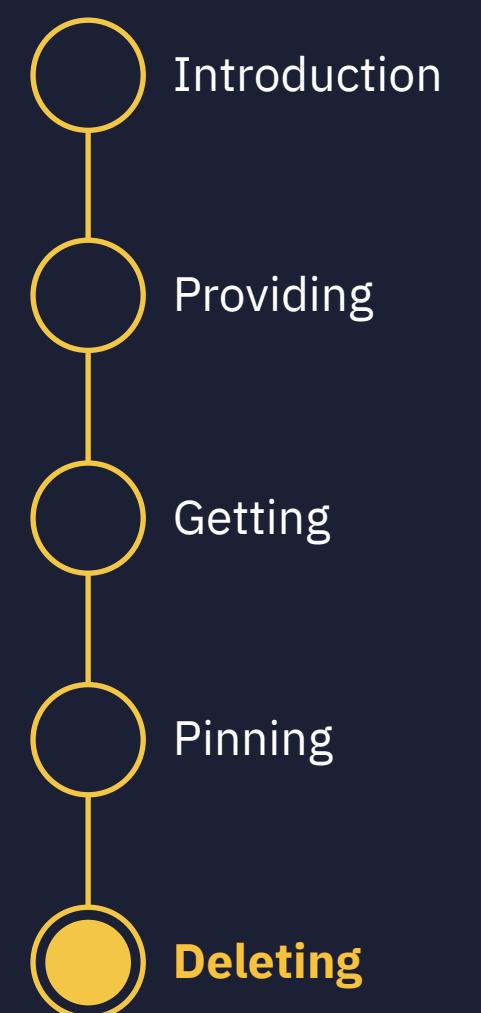
Shard your pinset
across a cluster

Try IPFS Cluster!

Covered in Elective Course B -
Building with Cluster



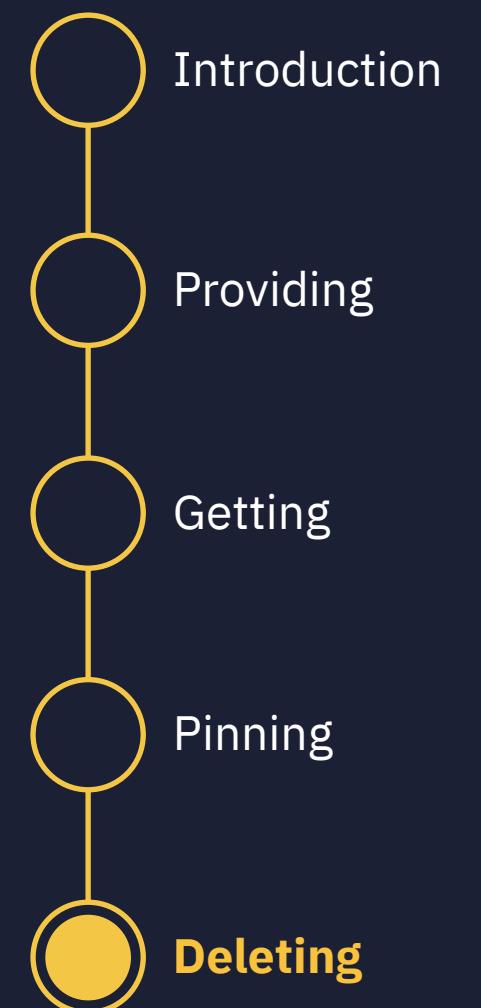
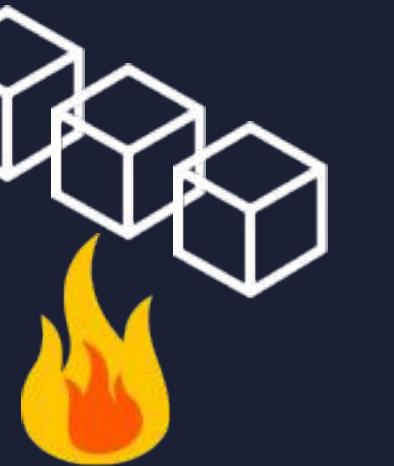
Deleting





Deleting

DO NOT WANT!



Deleting is local!



```
f ipfs pin rm QmTbw4KZA5t7TQkSSnJtKnm8PXTHwpTMGqy3JaWnow8V7F  
unpinned QmTbw4KZA5t7TQkSSnJtKnm8PXTHwpTMGqy3JaWnow8V7F
```

```
f ipfs block rm QmTbw4KZA5t7TQkSSnJtKnm8PXTHwpTMGqy3JaWnow8V7F  
removed QmTbw4KZA5t7TQkSSnJtKnm8PXTHwpTMGqy3JaWnow8V7F
```

Do Not Replicate...

YOU HAVE COMPLETED



THE LIFECYCLE OF DATA IN THE DWEB

CORE COURSE D



Oli Evans

IPFS GUI



Michelle Hertzfeld

IPFS Cluster



Kishan Sagathiya

IPFS Cluster

Content planning in

- <https://hackmd.io/wp2otwYZQHaAKcjJIAYMdA?edit>
- https://docs.google.com/document/d/16PcIzXxqzAU7-1ac-hFdsQ_7pSUkb5qqMqj9oQwmI08/edit#
- <https://www.youtube.com/watch?v=tpqXUmokFZ0>

How does
ipfs.io deal with
bad bits?

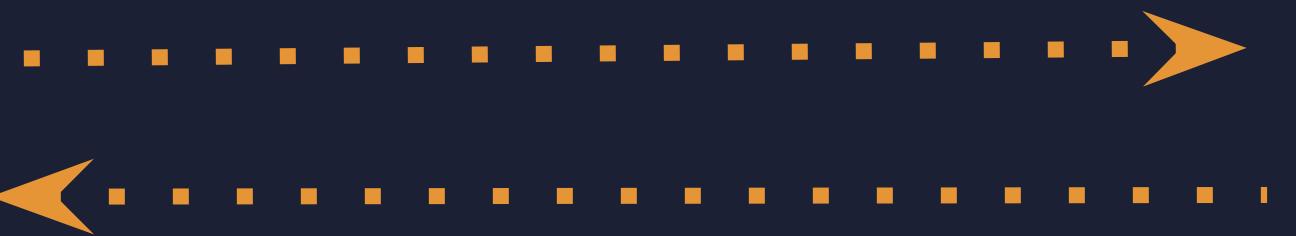
Agency!

all operations are local

You find where CIDs are by
asking someone with a
similar PeerID.

BitSwap

Root CID



Root Block

Parse Block, Get links

CIDs of links



Blocks of CIDs

Several roundtrips

Collaborating in the unknown

A dance of discovery

Who has the **block** I want?

Ask the peers with **PeerIDs**
most similar to the **CID**



QmBrian

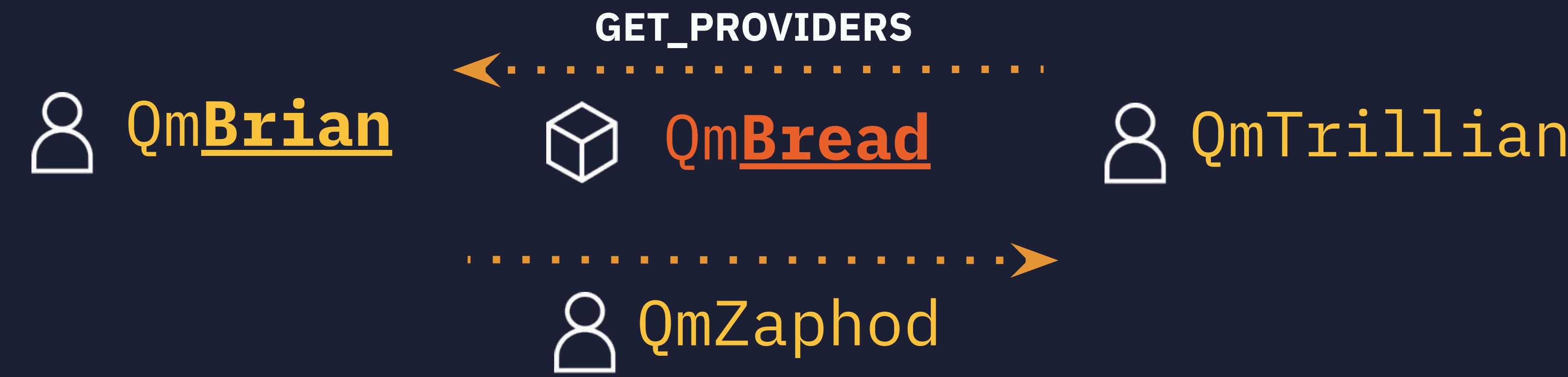


QmBread



QmTrillian

GET_PROVIDERS



libp2p



We connect to them...
now what?

graphsync □

BitSwap

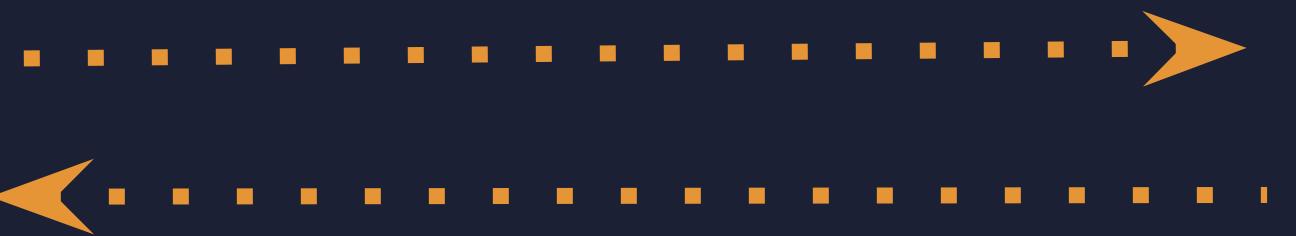
Single CID



Single Block

BitSwap

Root CID



Root Block

Parse Block, Get links

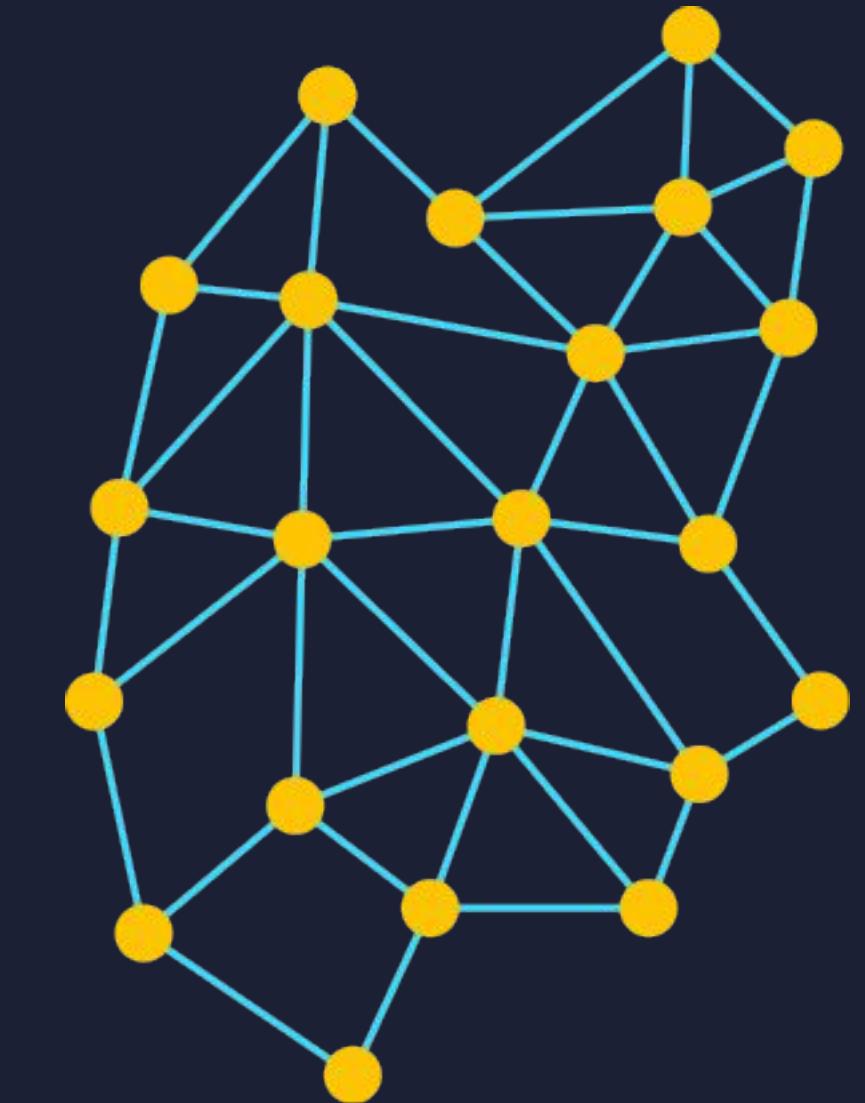
CIDs of links



Blocks of CIDs

Several roundtrips

GraphSync



A protocol for synchronizing IPLD Graphs among peers. It allows to make a single request to a remote peer for all results of traversing an IPLD Selector on the remote peer's local IPFS graph.

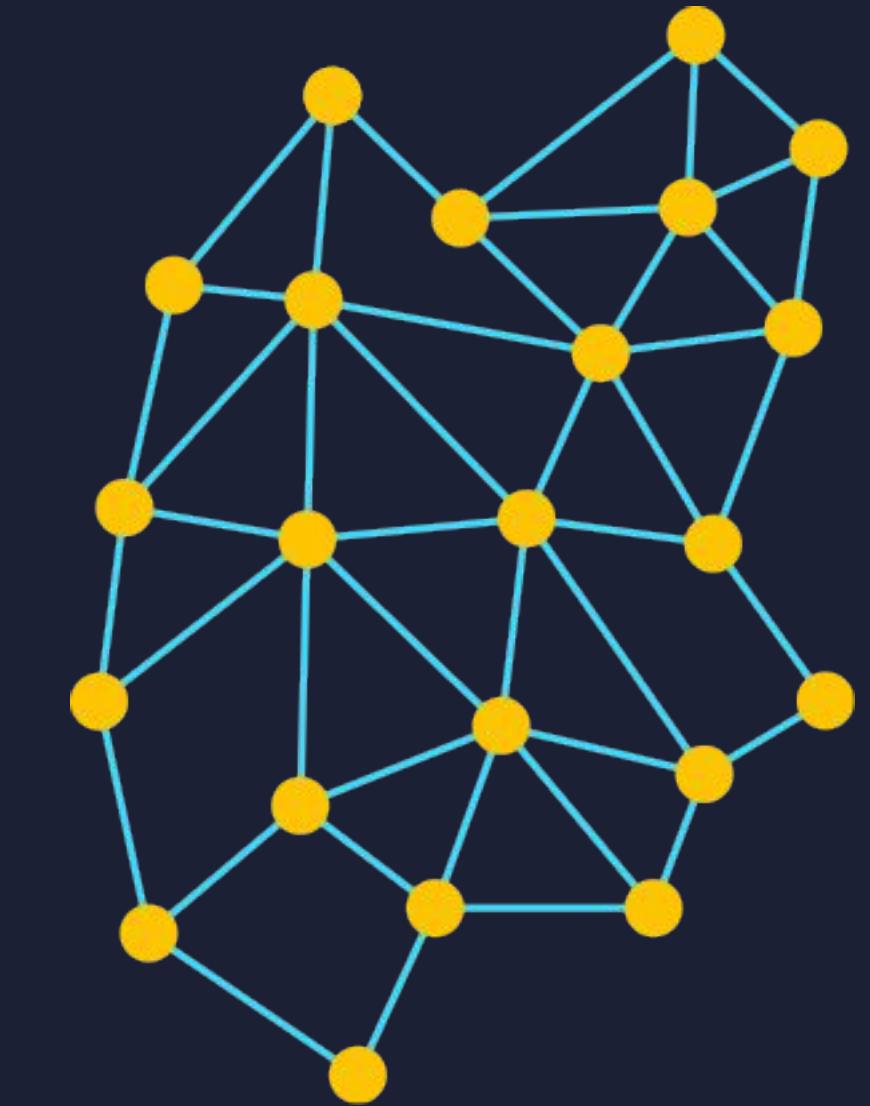
GraphSync

Single CID + Metadata



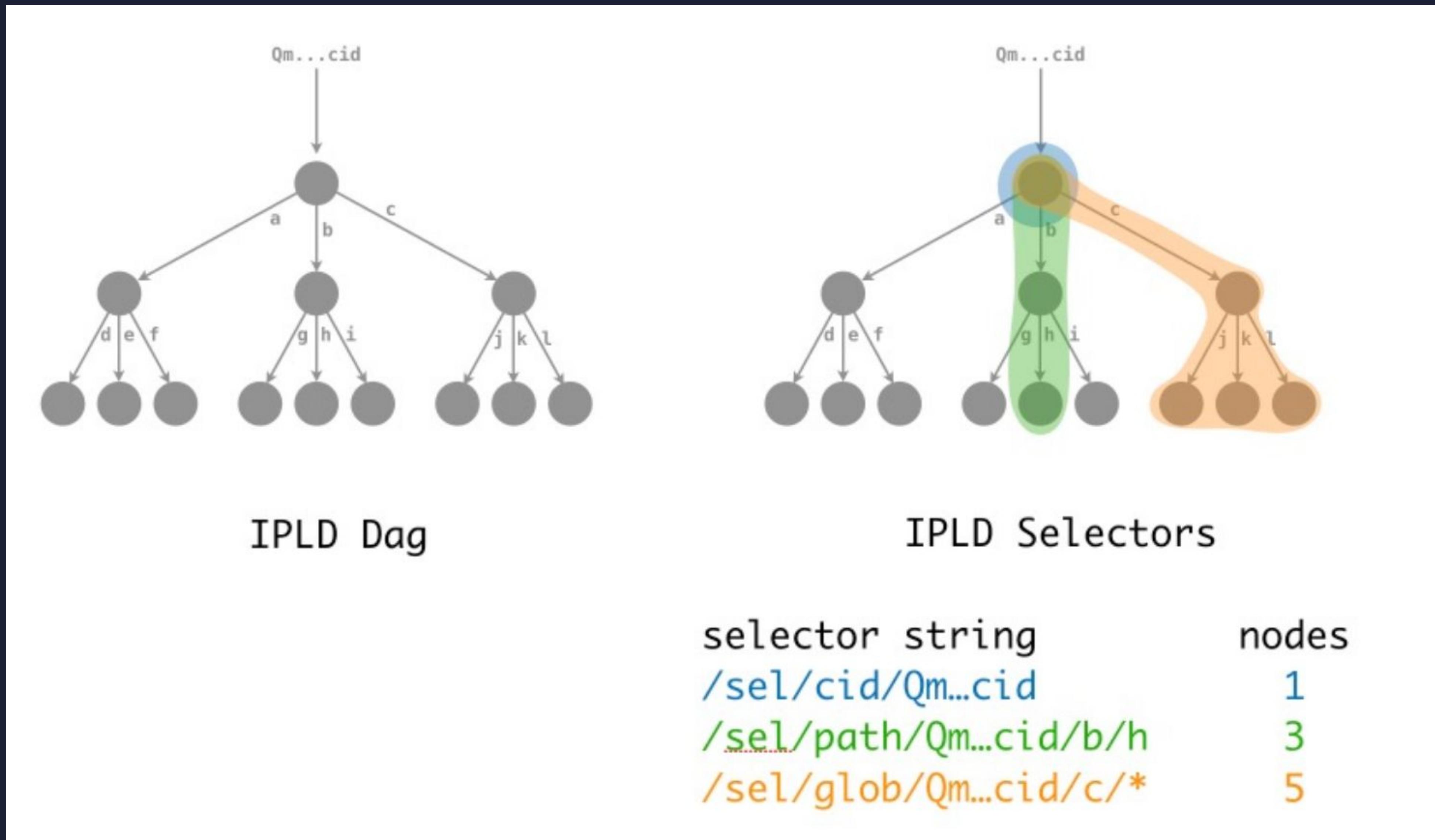
Several Blocks

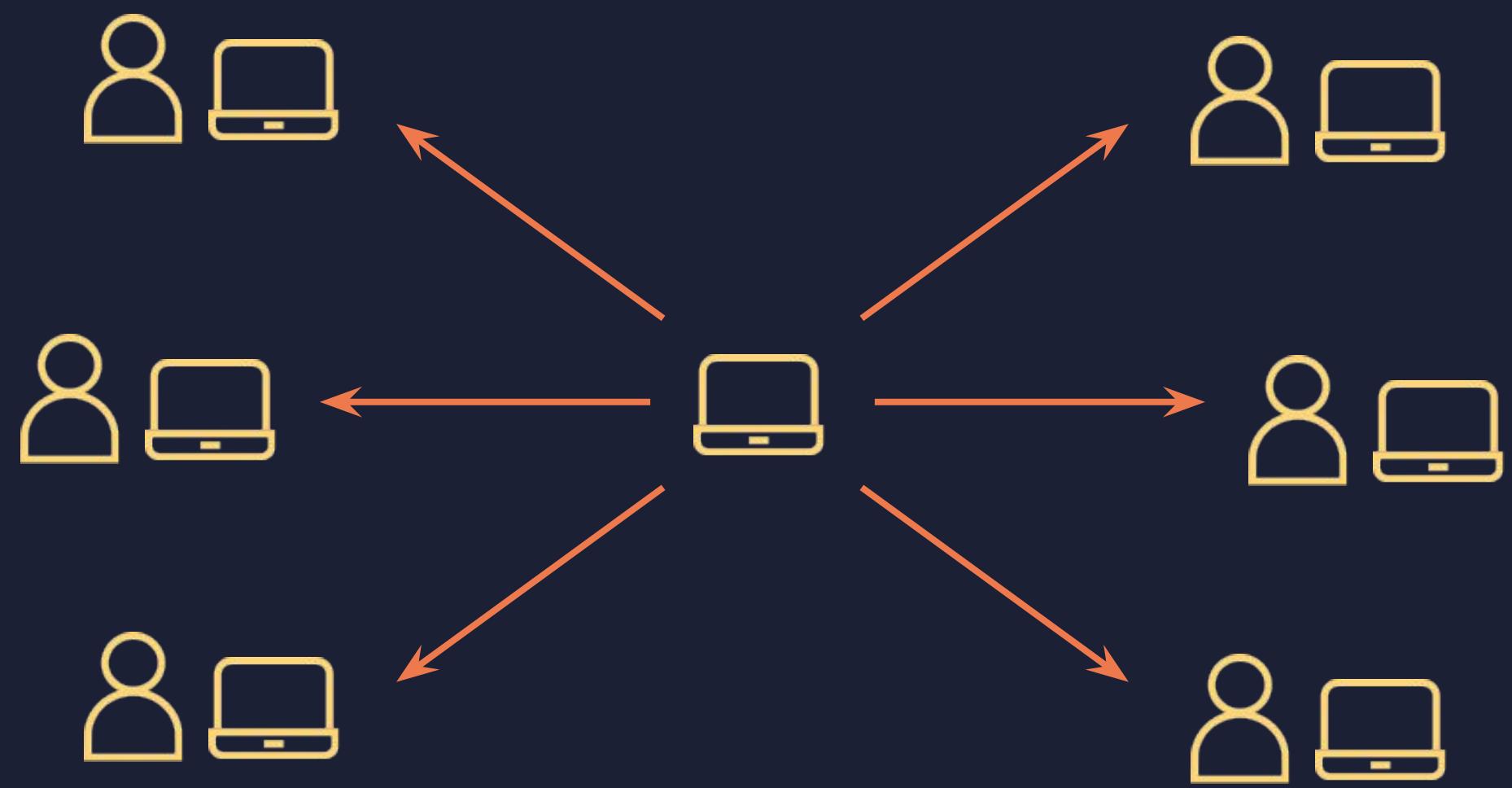
IPLD Selectors



Expressions that identify ("select") a subset of nodes in an IPLD dag.

IPLD Selectors





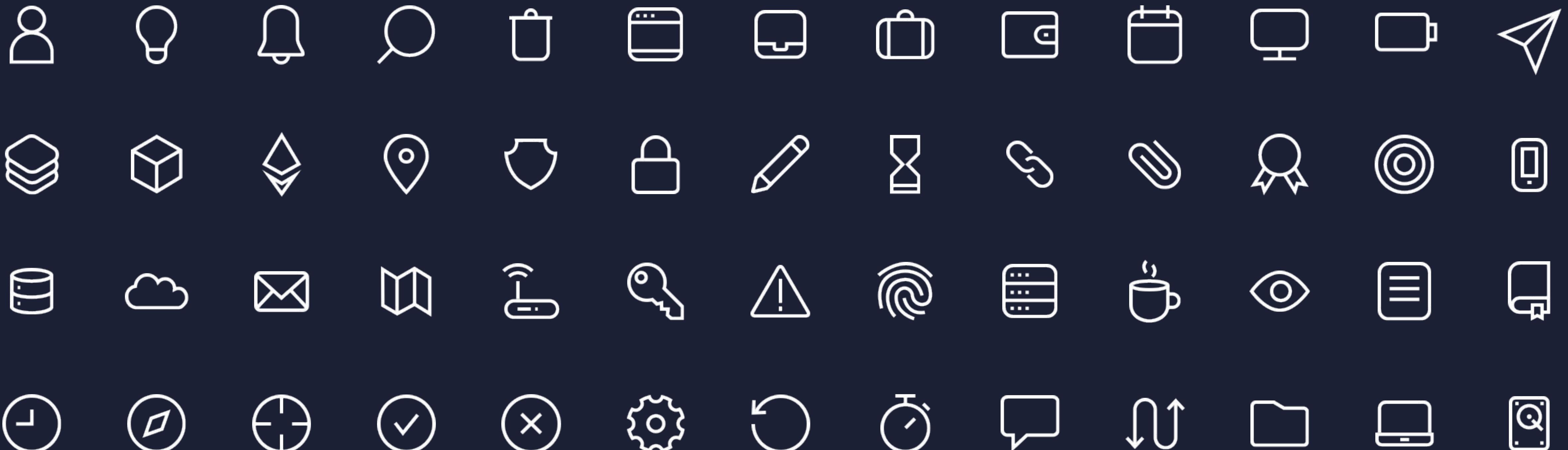
Don't want to run your own infrastructure?

Try a pinning service!

Pinning services such as
Pinata, Infura, and Eternum
will pin content for you via their APIs

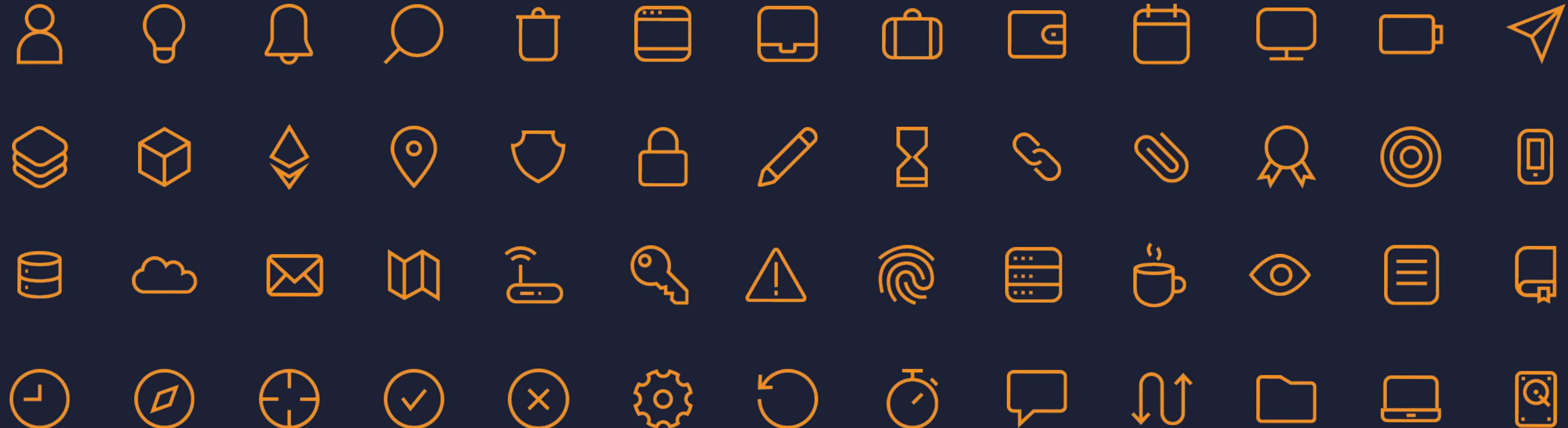
Subtitle:

52 ICONS FREE TO USE



Subtitle:

52 ICONS FREE TO USE



Subtitle: EMOTICONS FREE TO USE

