A P2P Dropbox

@mafintosh

dat

DATA

ALFRED P. SLOAN FOUNDATION

S

1934

8 person team

Based in 5 countries

>1500 npm modules

>1500 npm modules

(~0.5% of npm)

We make tools that help scientists share data

We make tools that help scientists share data

(and other people as well)

Data === Files

Existing great file sharing tools

Dropbox

- Extremely easy to use
- Centralised / High cost
- Who owns the data?
- Sustainable?

BitTorrent®

- Decentralised / P2P
- Massive adopted / Simple protocol
- Only works for static files
- Scales worse on really big data sets
- No diffs

We can do better

- Easy to use, but not centralised like Dropbox
- Decentralised / P2P but not for piracy like BitTorrent
- Build for modern use cases

- Easy to use, but not centralised like Dropbox
- Decentralised / P2P but not for piracy like BitTorrent
- Build for modern **(scientific)** use cases

A next generation file sharing tool

Real time / Live data

(get only the data you need and get updates when it changes)

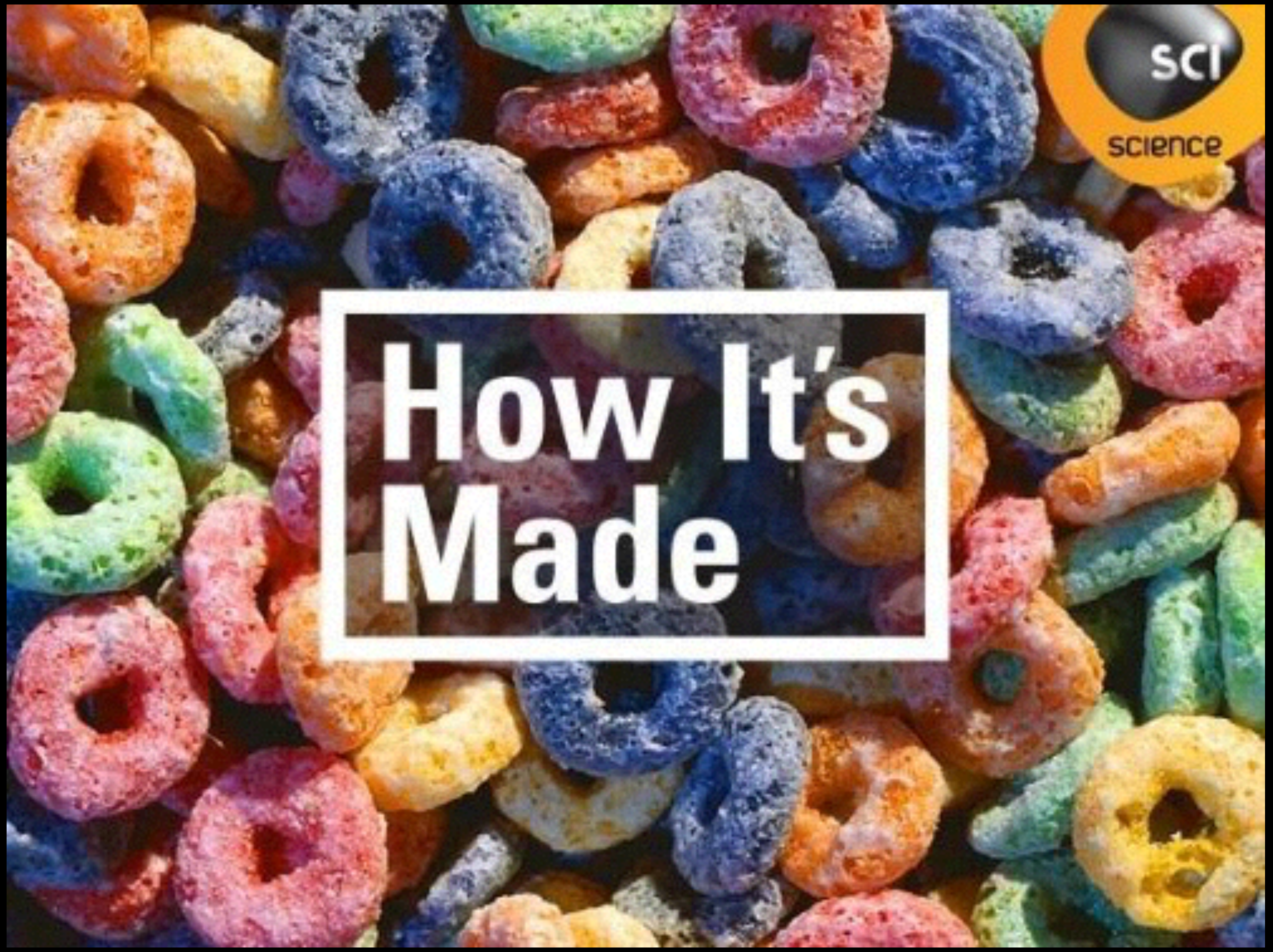
Decentralised

(no servers / data centers needed, actually serverless)

Diffable

(sharing two similar data sets should only share the diff)

npm install -g **dat**
DATA


sci
science
How It's Made

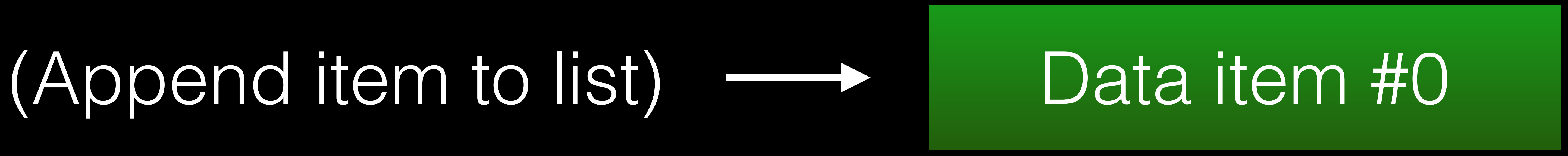
Append only logs

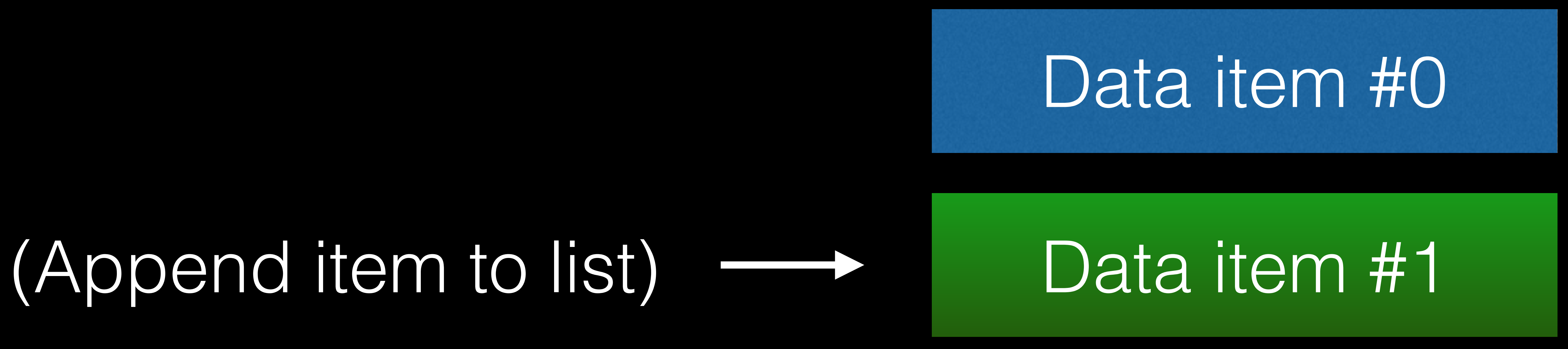
Append only logs

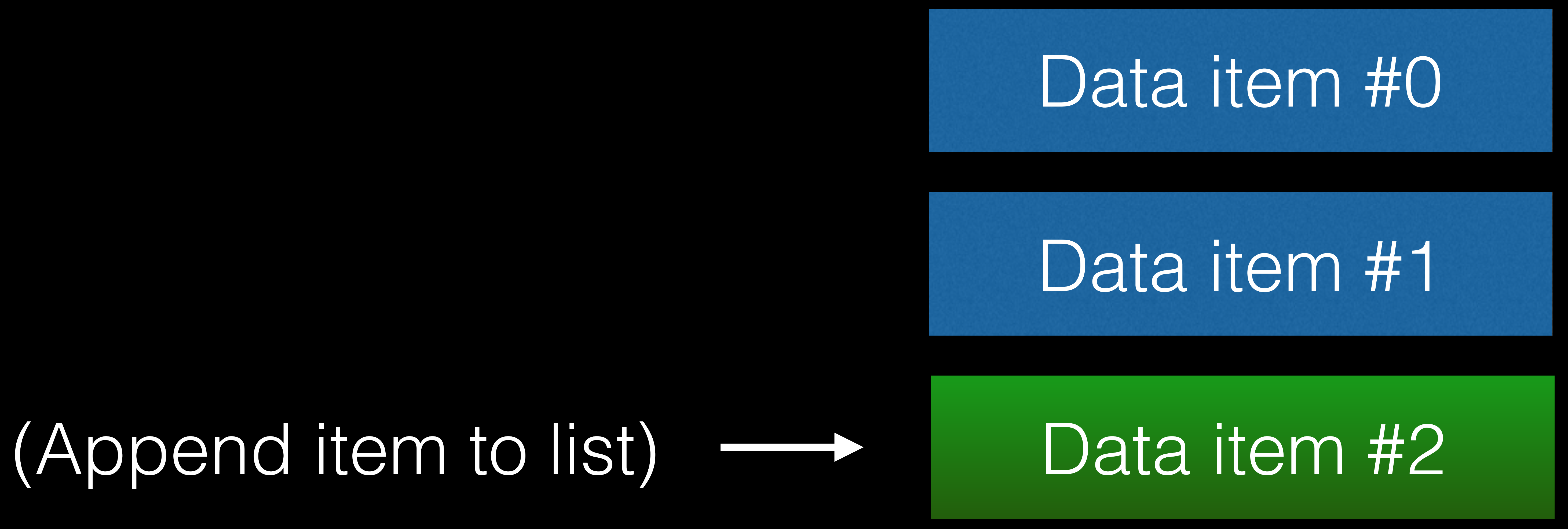
(a list of data you only ever append to, *get it?*)

Append only ~~logs~~ lists

(a list of data you only ever append to, *get it?*)


(Append item to list)
Data item #0


Data item #0
(Append item to list)
Data item #1


Data item #0
Data item #1
Data item #2
(Append item to list)

Why “Append Only Logs”?

- A simple data structure
- Immutable
- Logical ordering
- Easy to digest / index

How can we share append only logs?

How can we share append only logs?

(over a p2p network where we don’t trust other people)

Merkle Trees

Merkle Trees

(a tree structure that verifies data)

Merkle Trees

(a tree structure that verifies data)

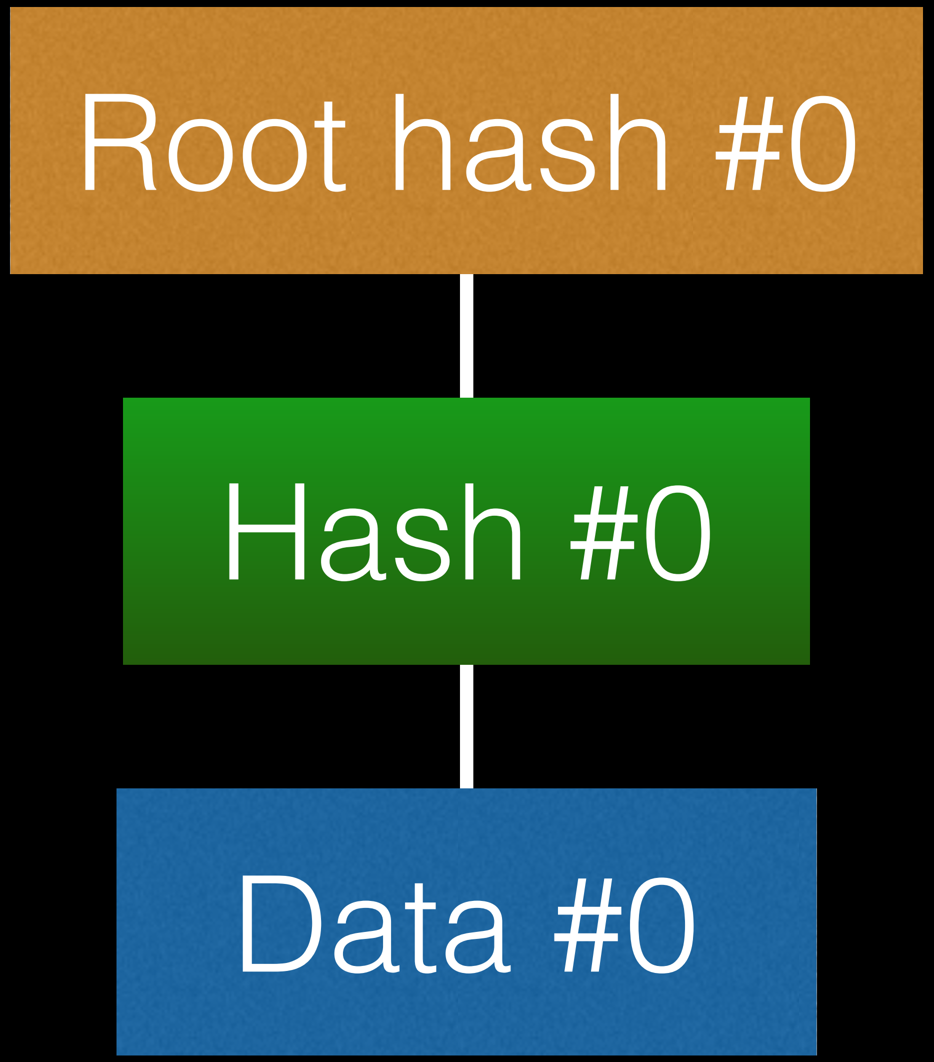
(unrelated to Angela Merkel)

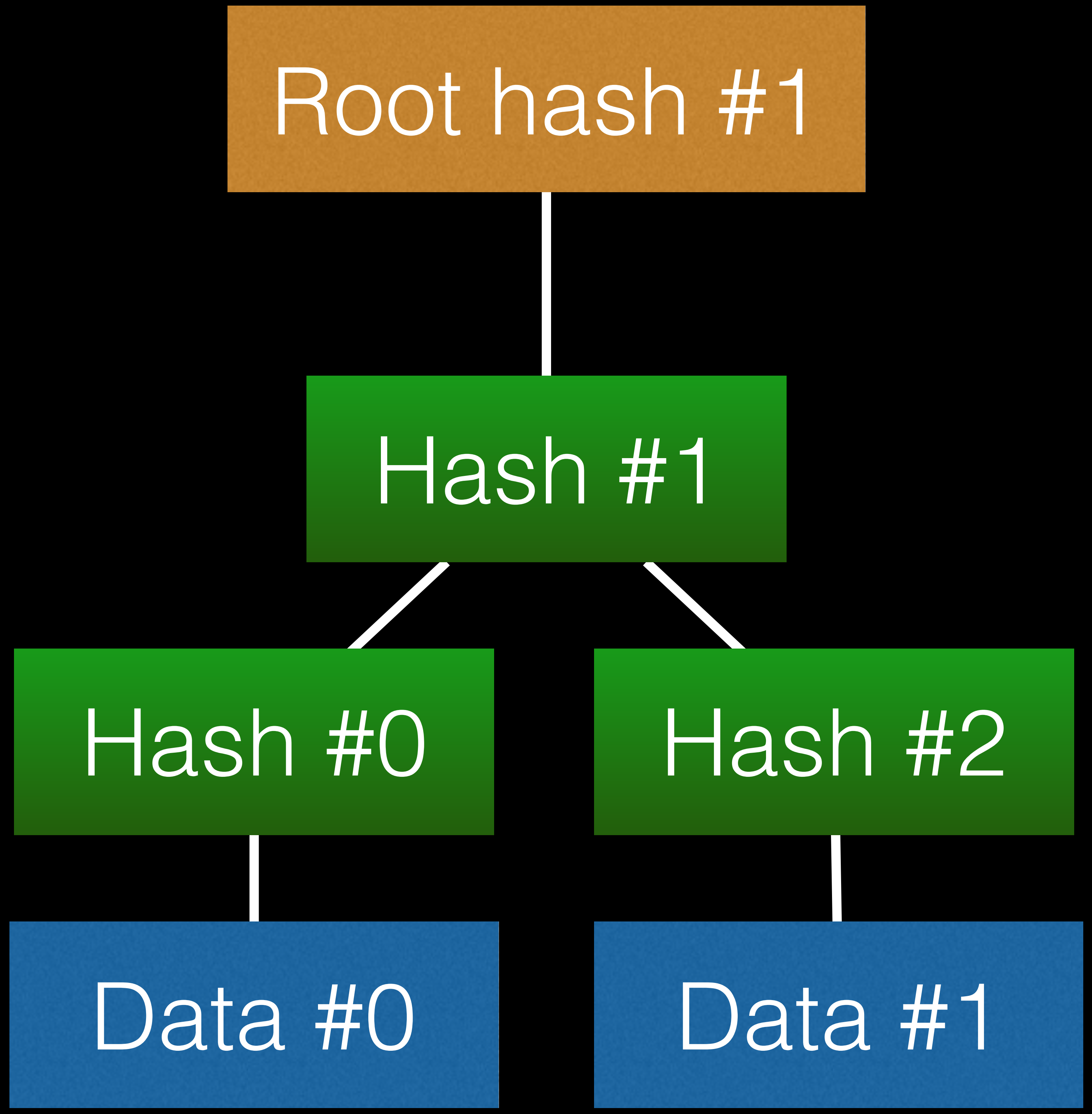
Merkle Trees

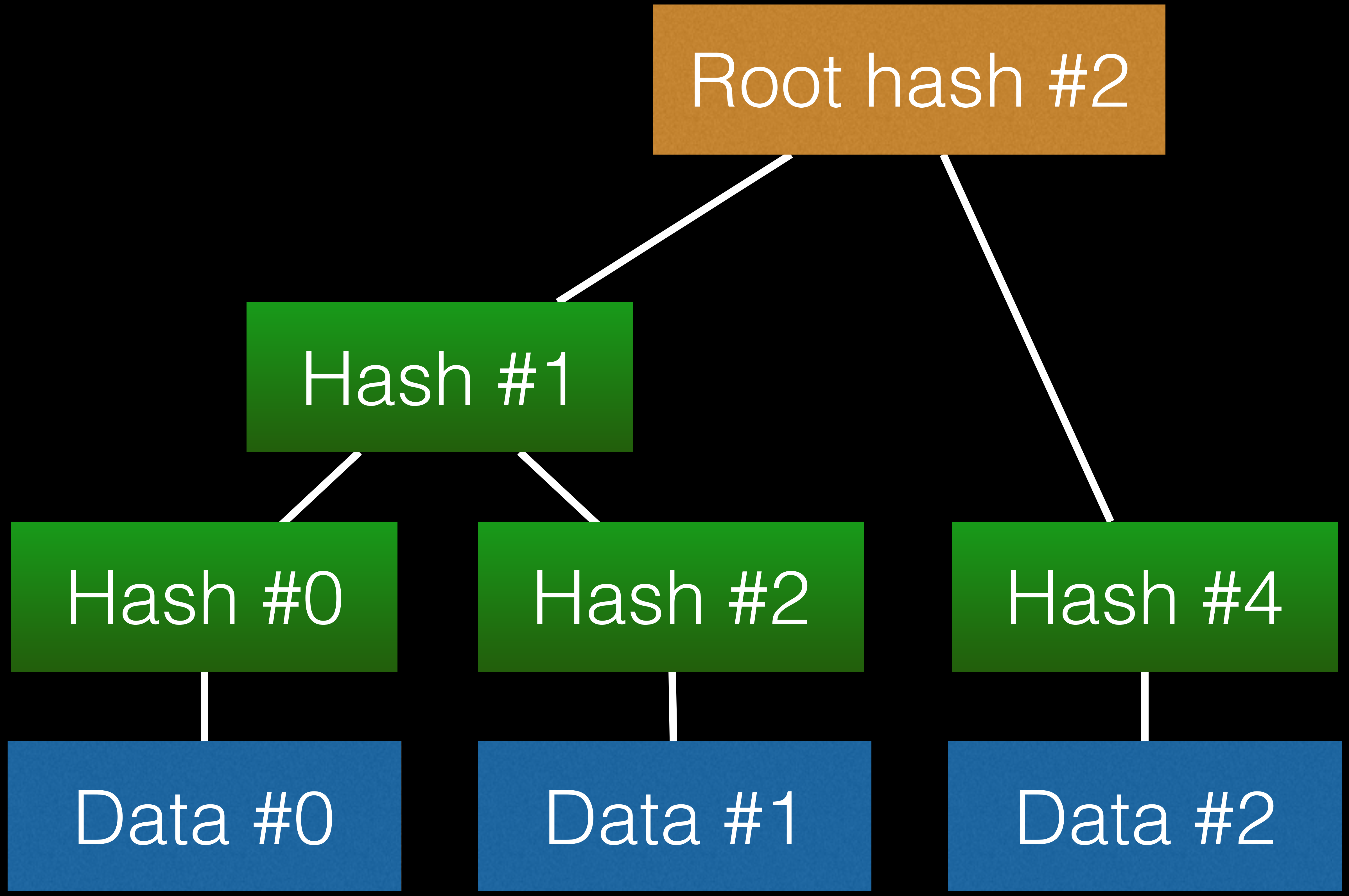
(a tree structure that verifies data)

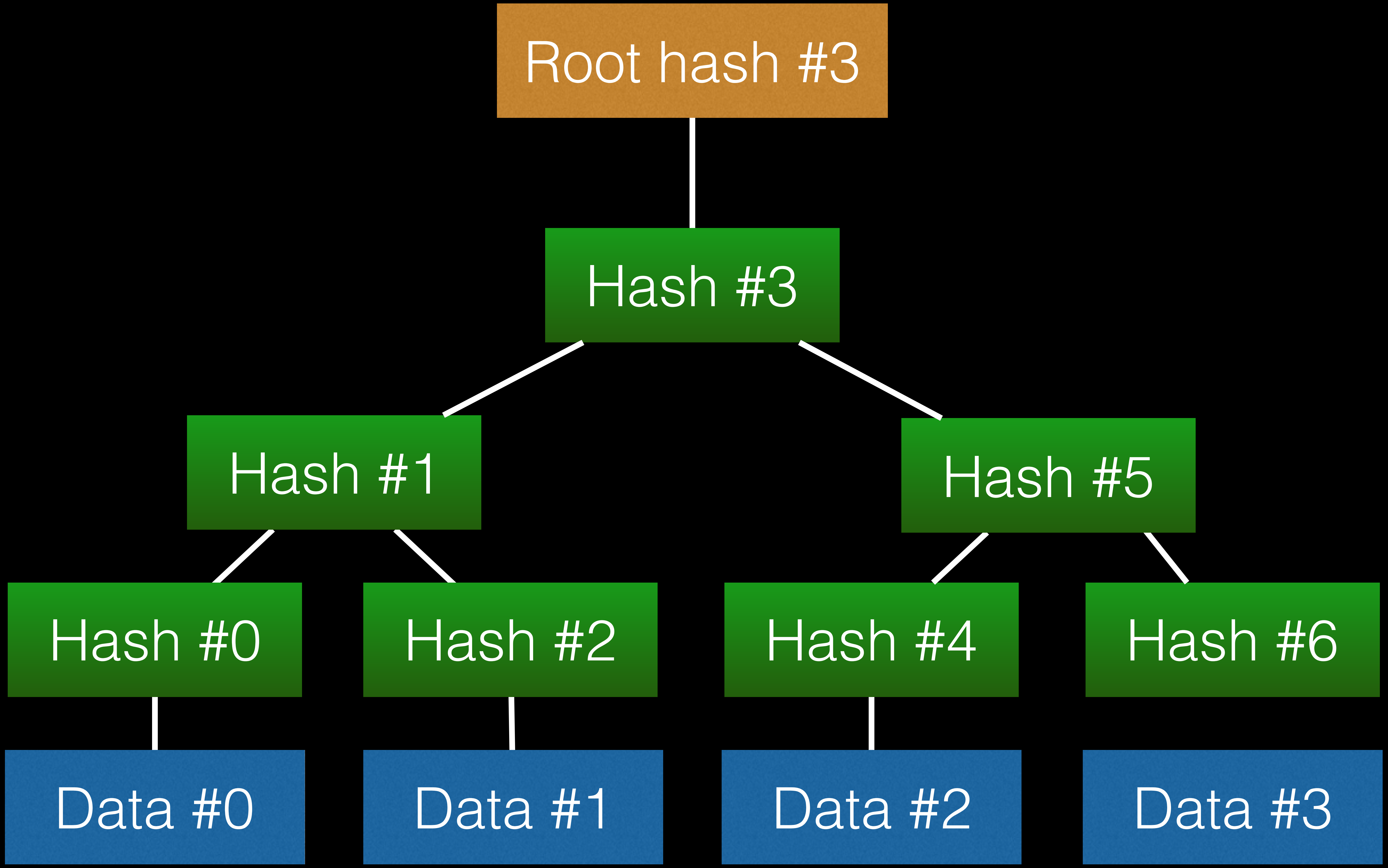
(unrelated to Angela Merkel)

Data #0


Root hash #0
Hash #0
Data #0

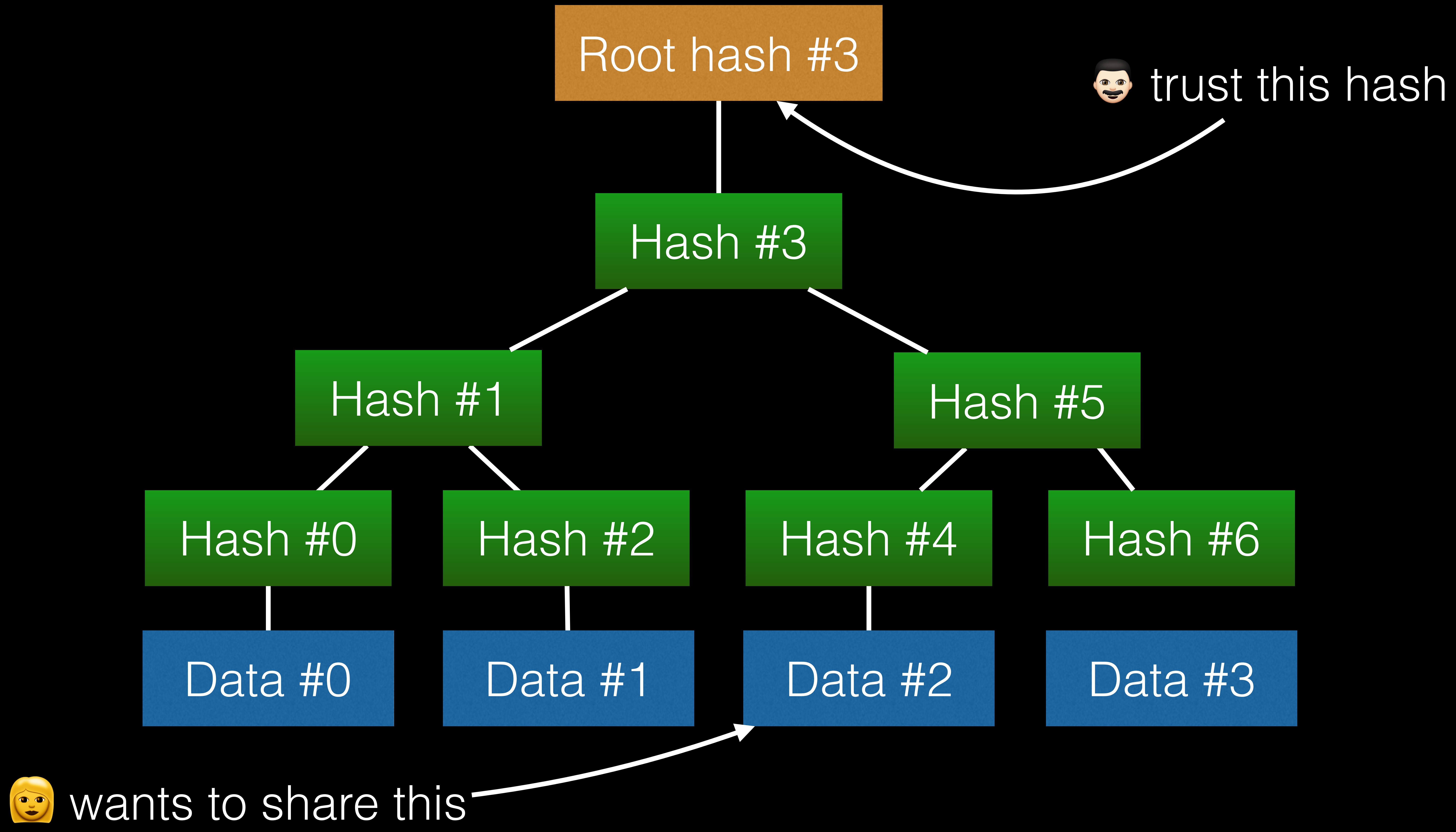

Root hash #1
Hash #1
Hash #0
Hash #2
Data #0
Data #1

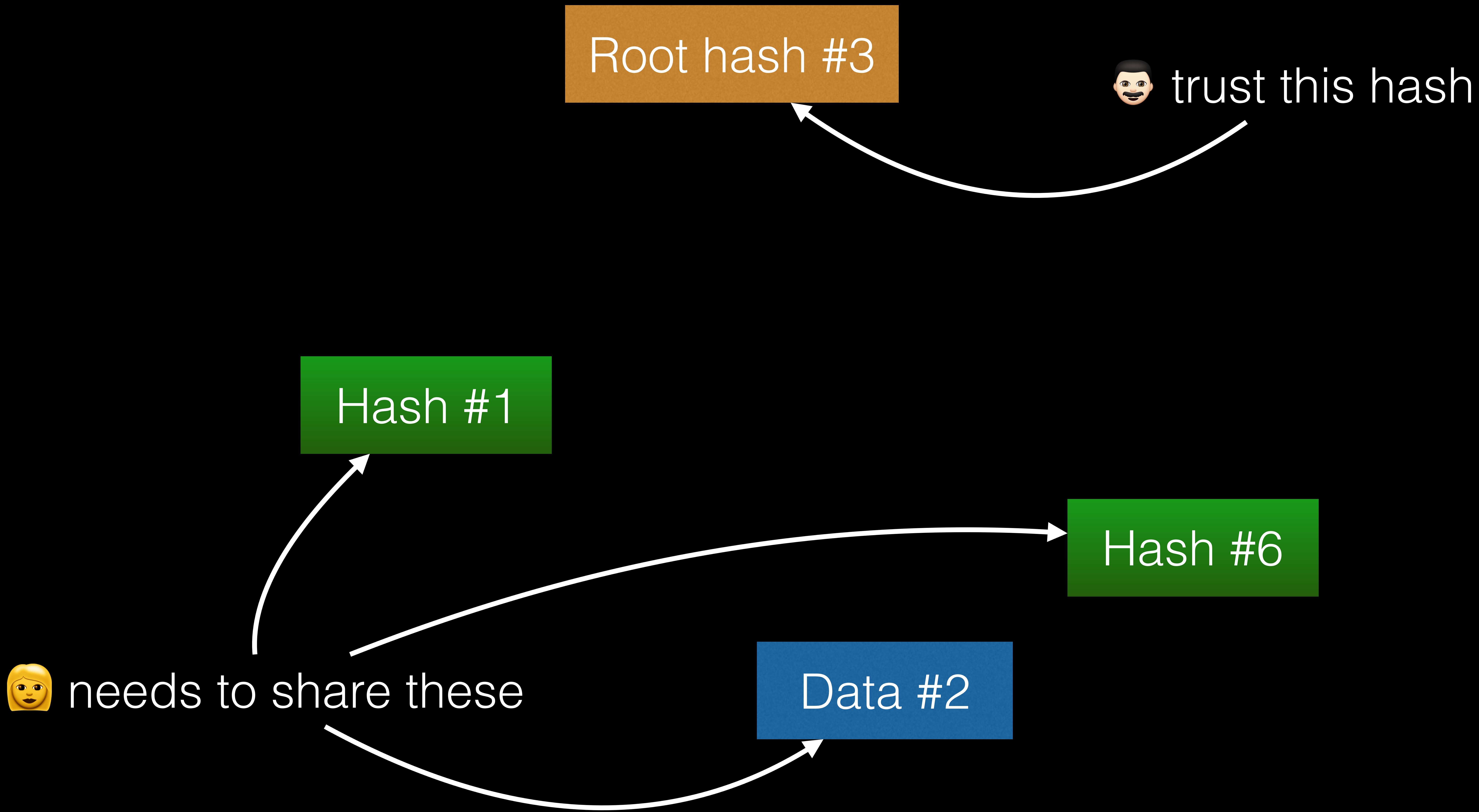

Root hash #2
Hash #1
Hash #0
Hash #2
Hash #4
Data #0
Data #1
Data #2

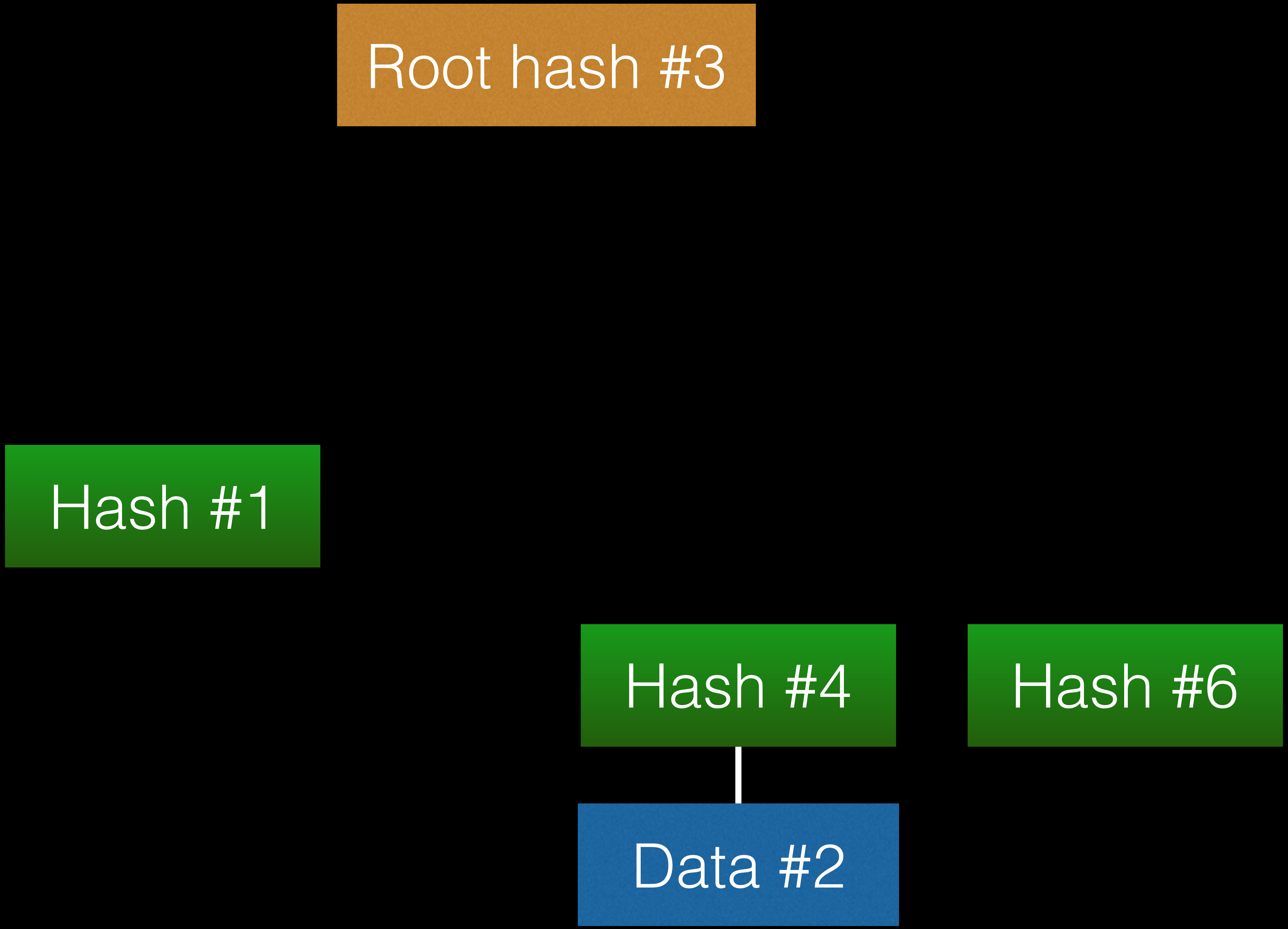

Root hash #3
Hash #3
Hash #1
Hash #5
Hash #0
Hash #2
Hash #4
Hash #6
Data #0
Data #1
Data #2
Data #3

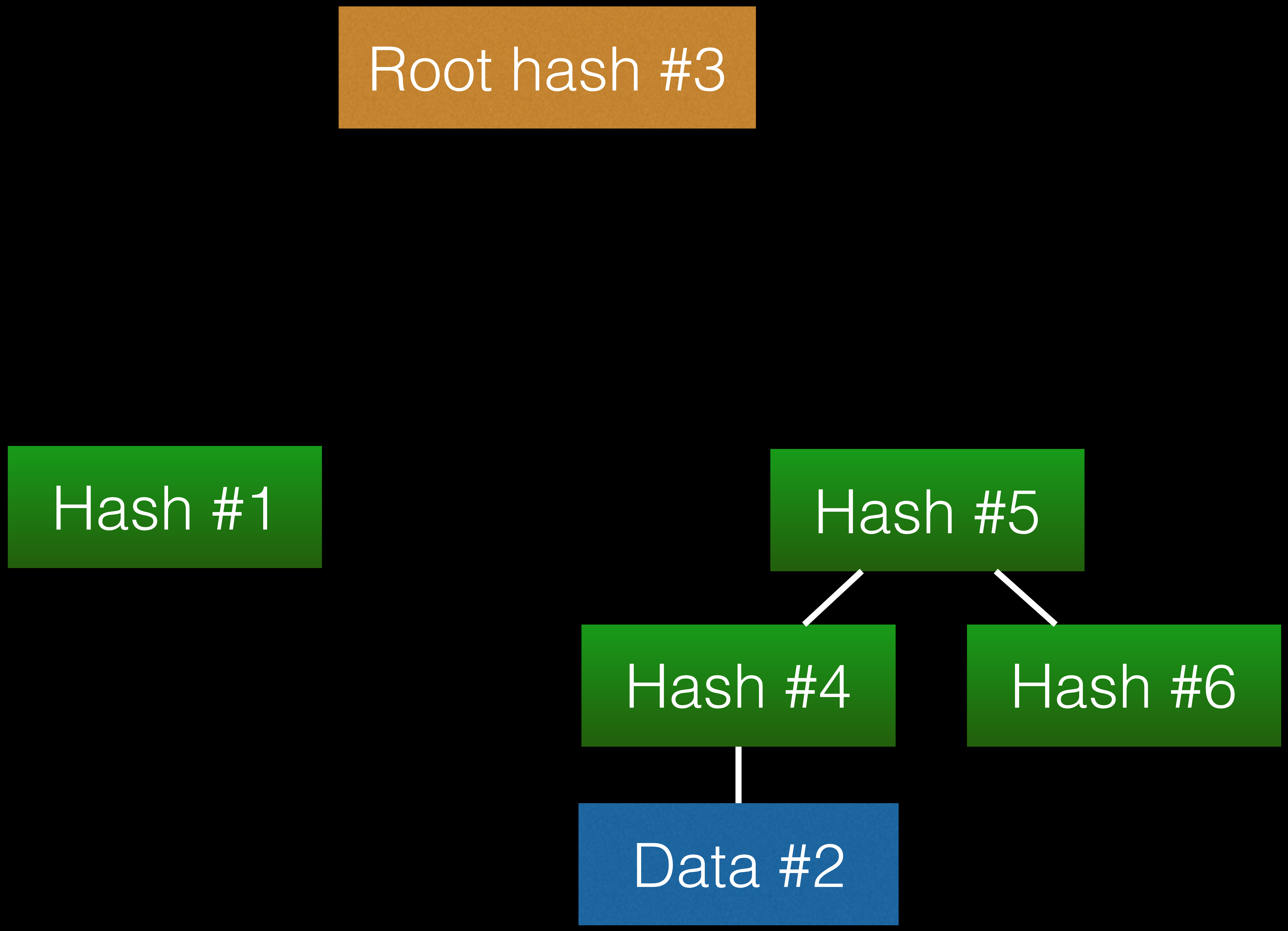
Root hash #3 verifies all the data

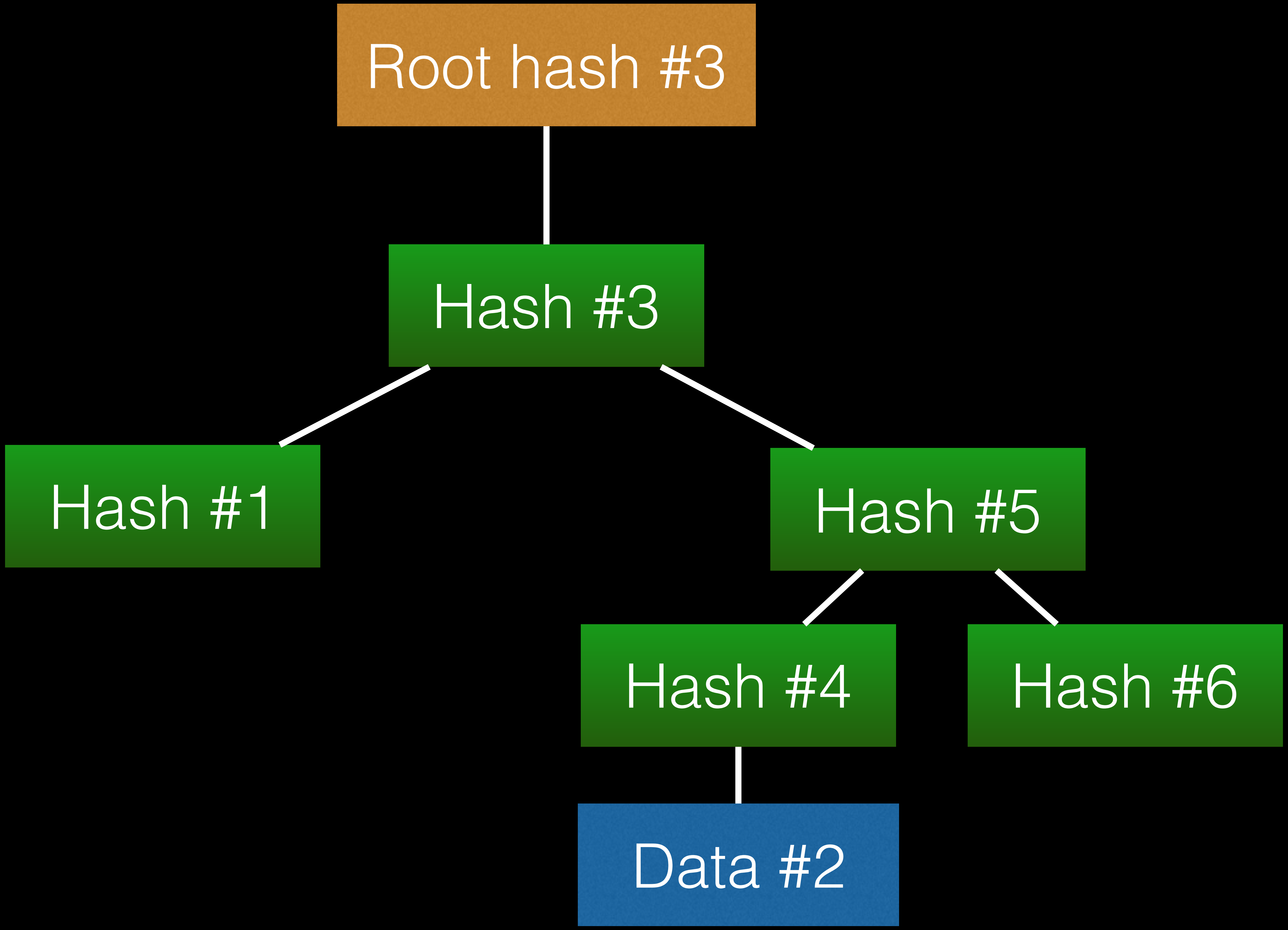
👩 wants to share Data #2 with 👨


Root hash #3
trust this hash
Hash #3
Hash #1
Hash #5
Hash #0
Hash #2
Hash #4
Hash #6
Data #0
Data #1
Data #2
Data #3
wants to share this


Root hash #3
👨 trust this hash
Hash #1
Hash #6
Data #2
👩 needs to share these

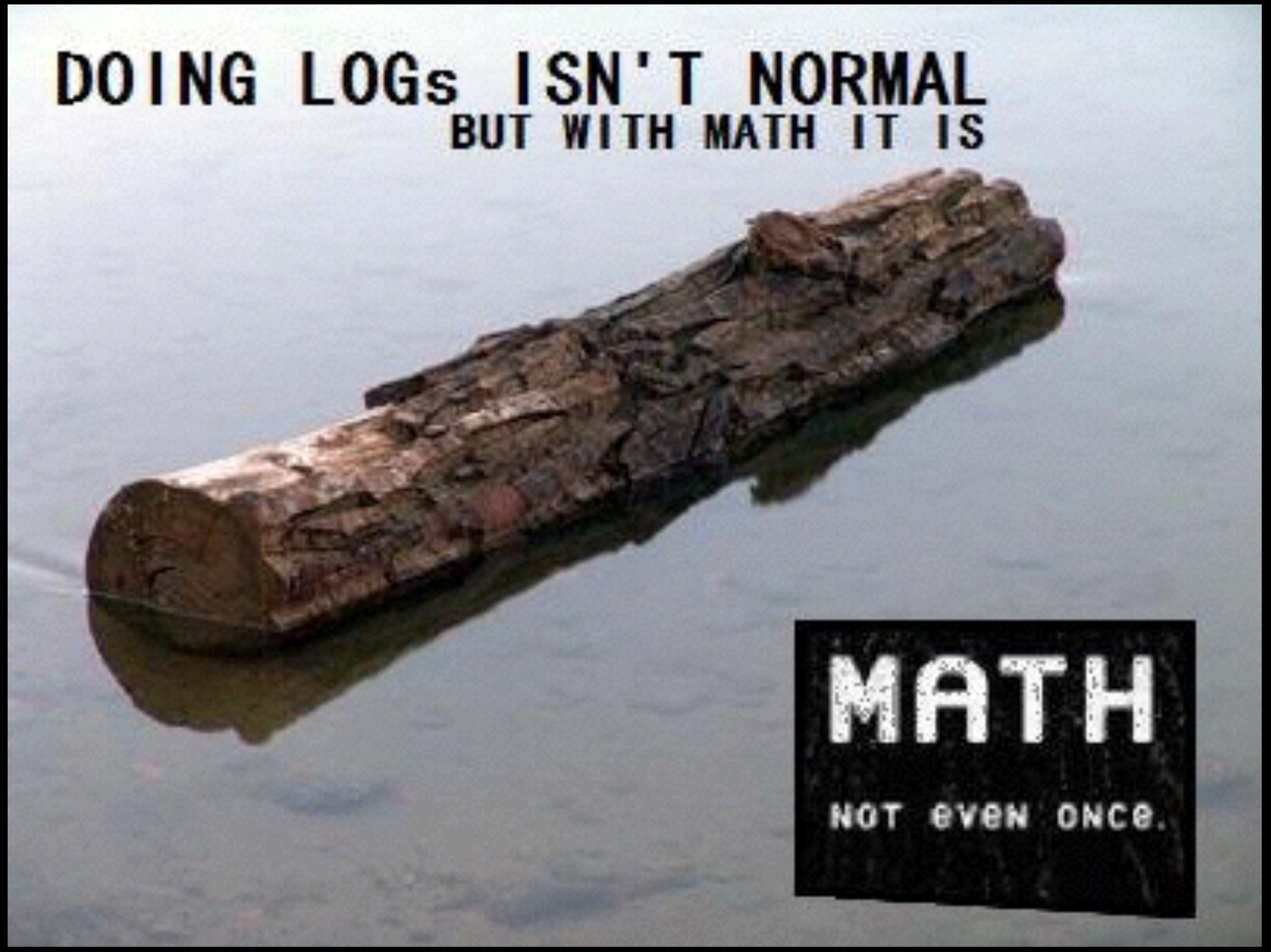

Root hash #3
Hash #1
Hash #4
Hash #6
Data #2


Root hash #3
Hash #1
Hash #5
Hash #4
Hash #6
Data #2

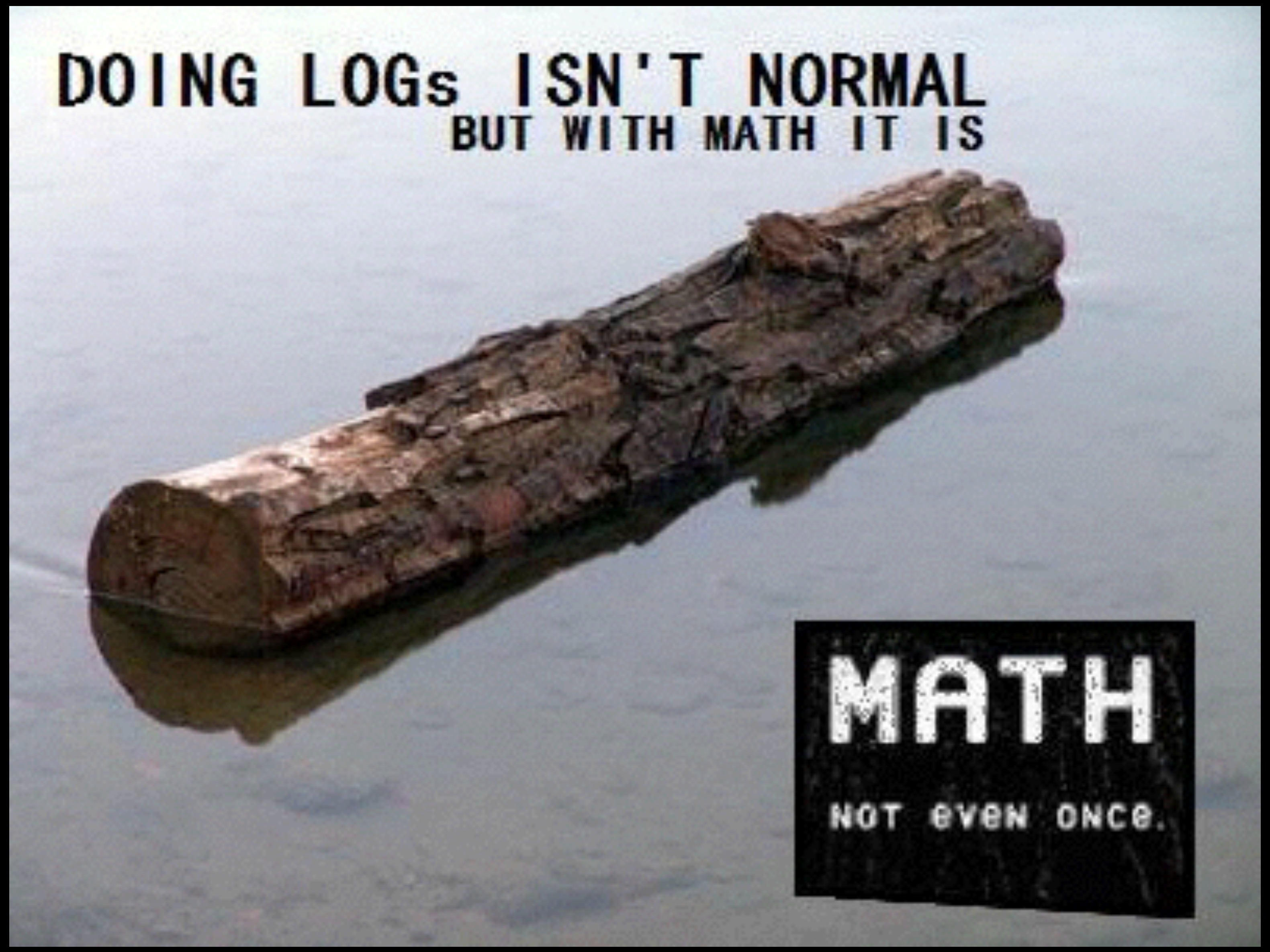

Root hash #3
Hash #3
Hash #1
Hash #5
Hash #4
Hash #6
Data #2

👨 checks that Hash #3 match Root hash #3

👱‍♀️ only needs to send `O(log(n))` hashes to 👨🏻

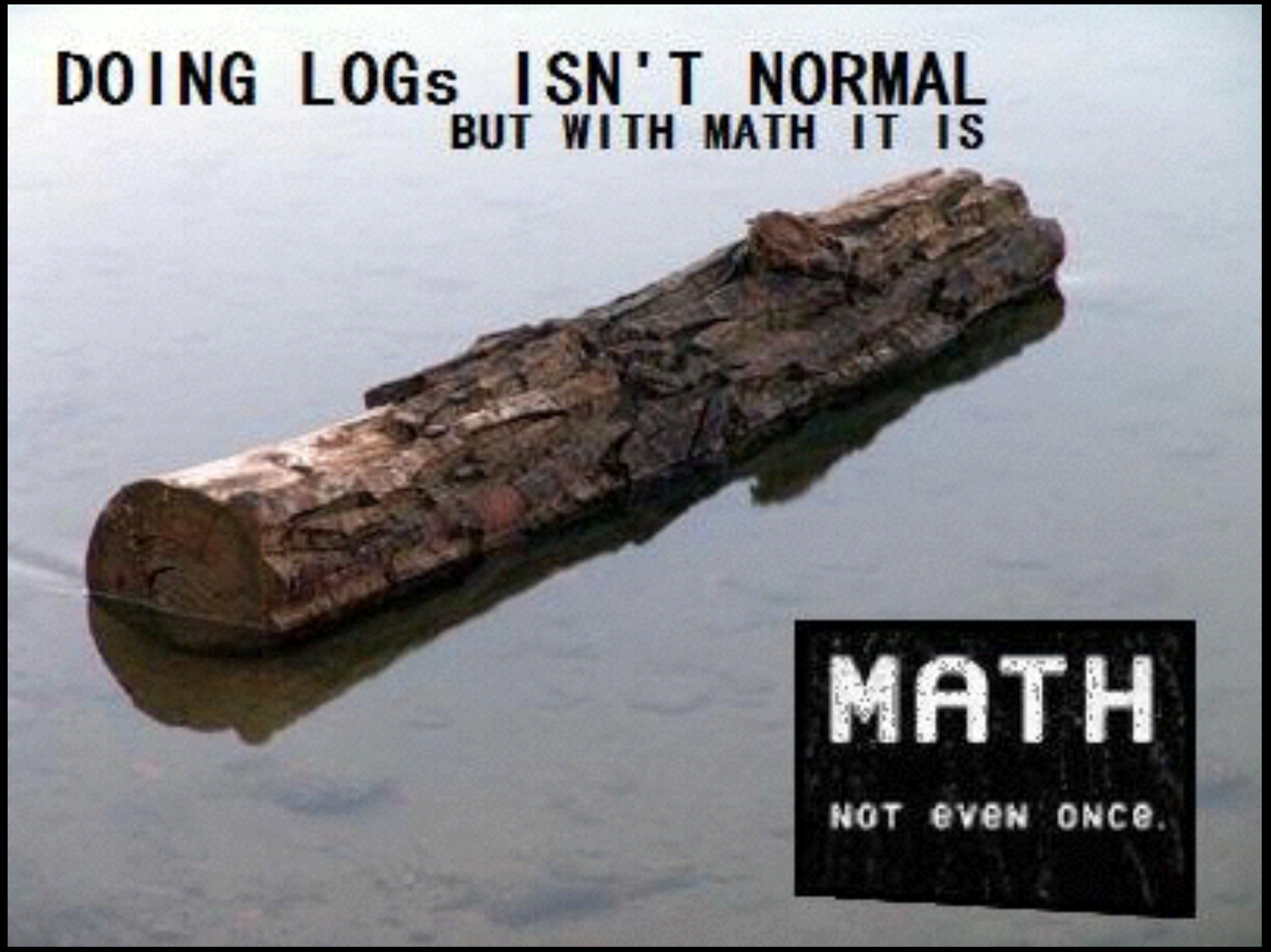


👱‍♀️ only needs to send `O(log(n))` hashes to 👨🏻



👩 only needs to send `O(log(n))` hashes to 👨

(can easily be optimised to never send the same hash twice)



👩 only needs to send `O(log(n))` hashes to 👨

(can easily be optimised to never send the same hash twice)

(come ask me later, i’m fun at parties)

Real time

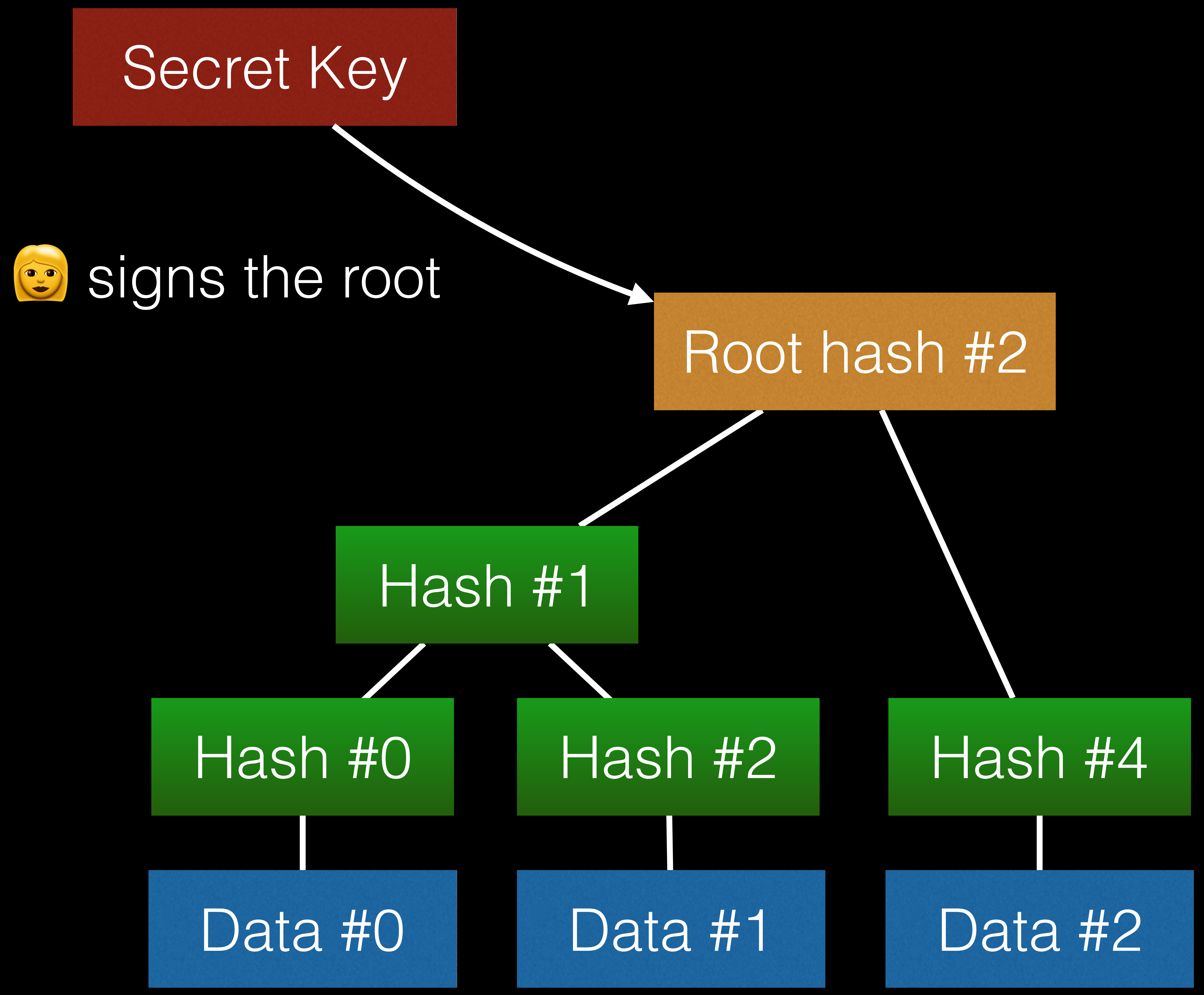
Every time we append data Root hash changes

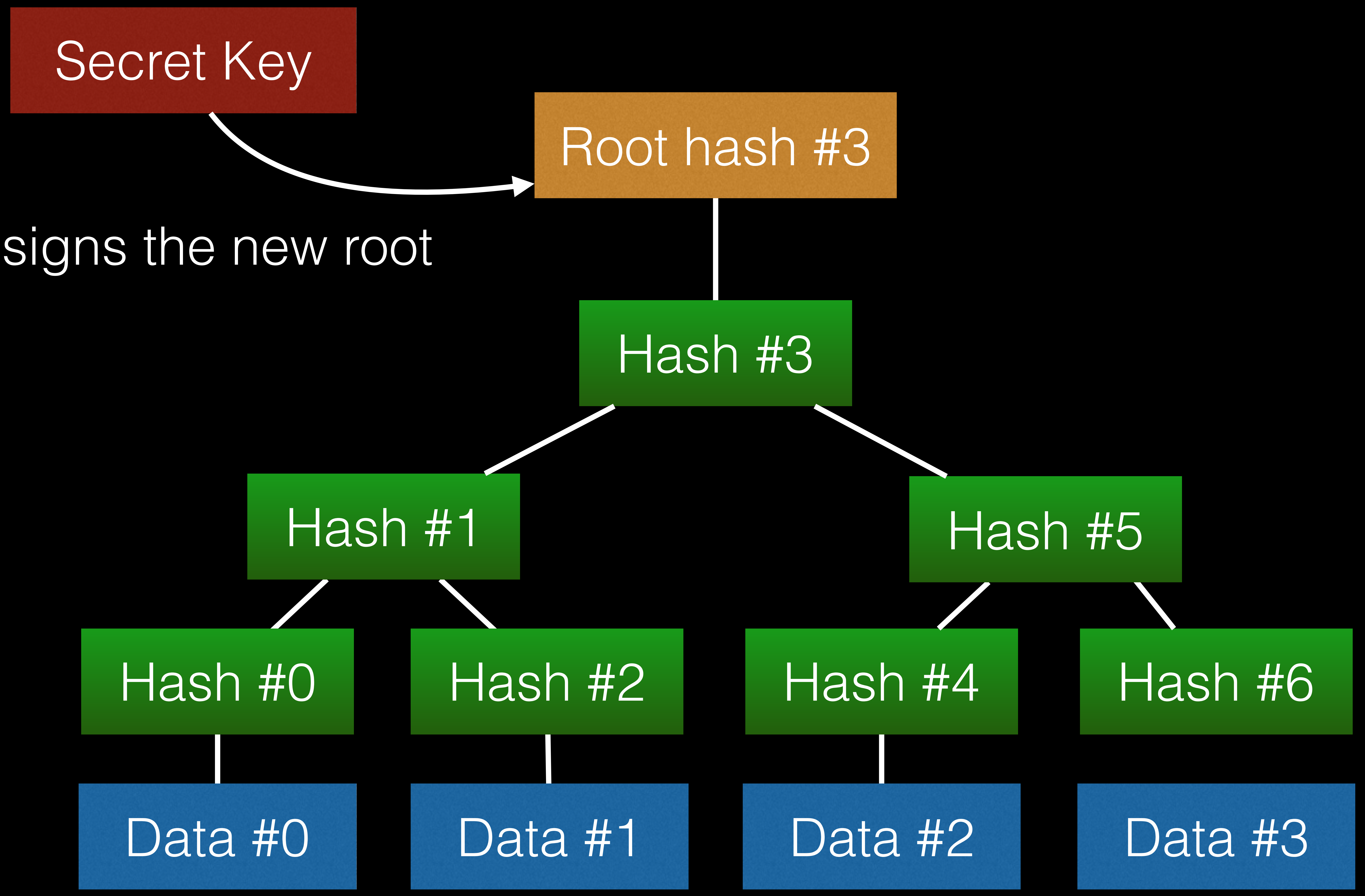
Crypto to the rescue

Generate a key pair

Secret Key + Public Key

👨 trusts Public Key


Secret Key
👩 signs the root
Root hash #2
Hash #1
Hash #0
Hash #2
Hash #4
Data #0
Data #1
Data #2



👩 signs the new root

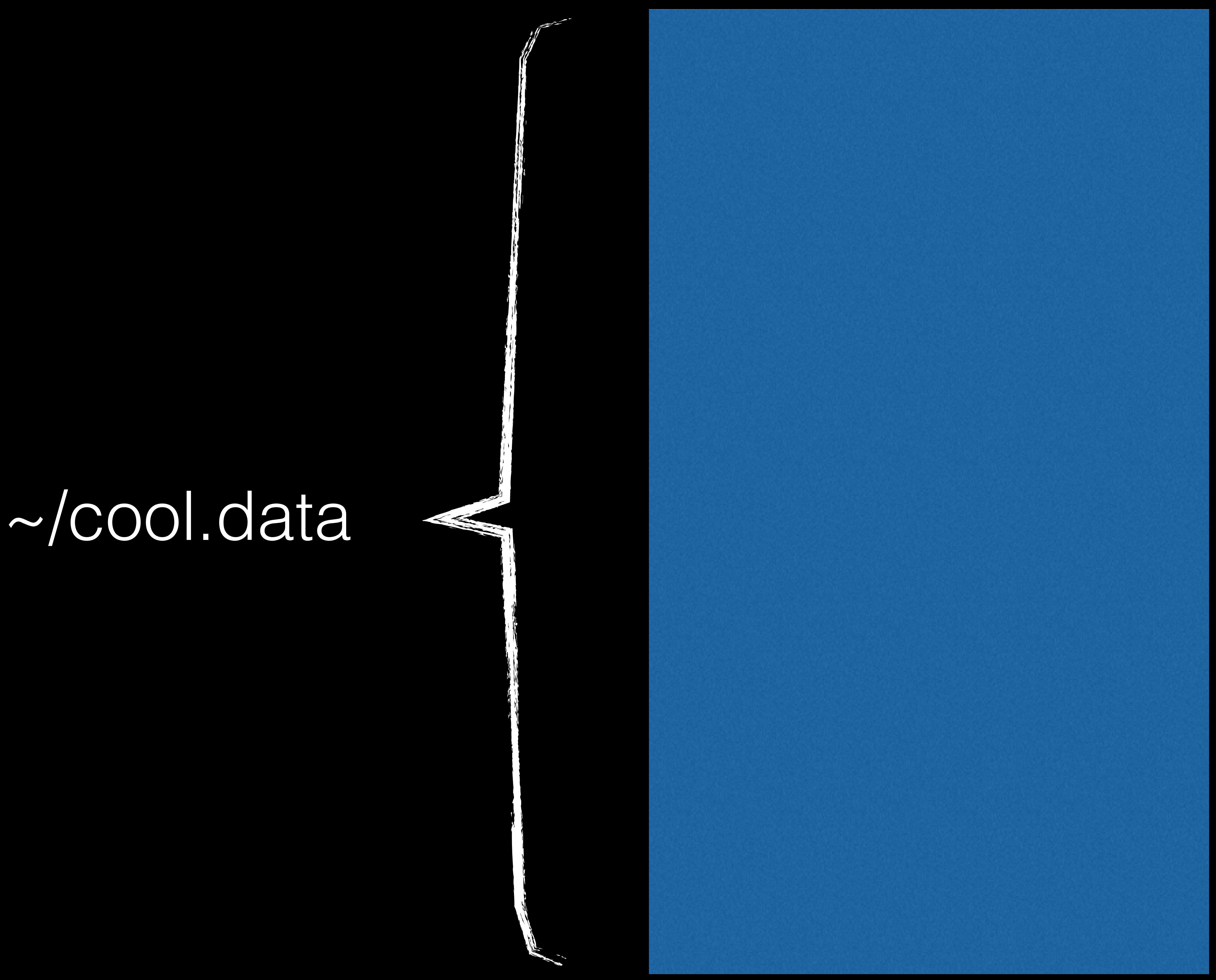
👨 uses Public Key to verify Root hash signatures

```
npm install hypercore
```

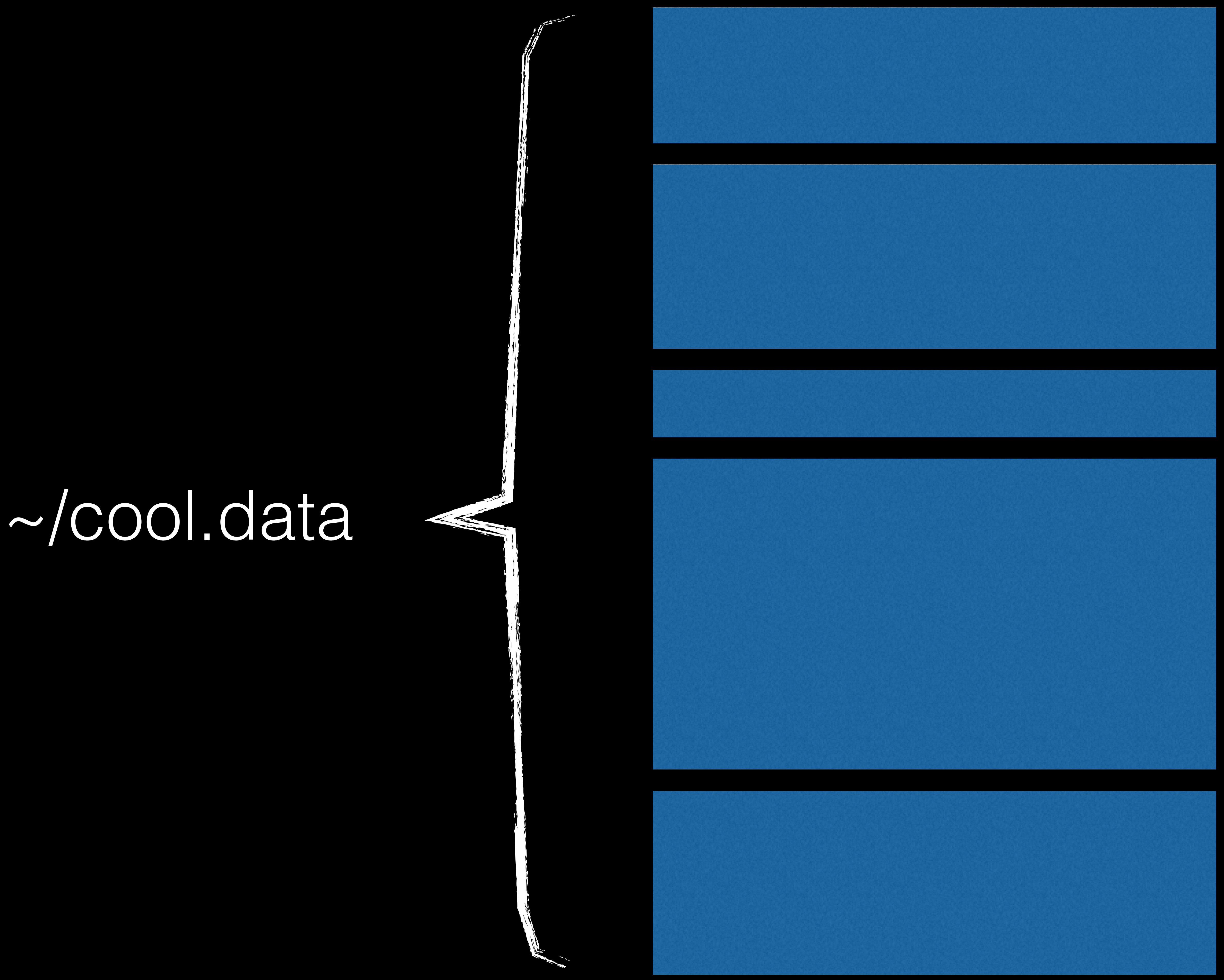
(demo)

How do we turn append only logs into a file sharing tool?

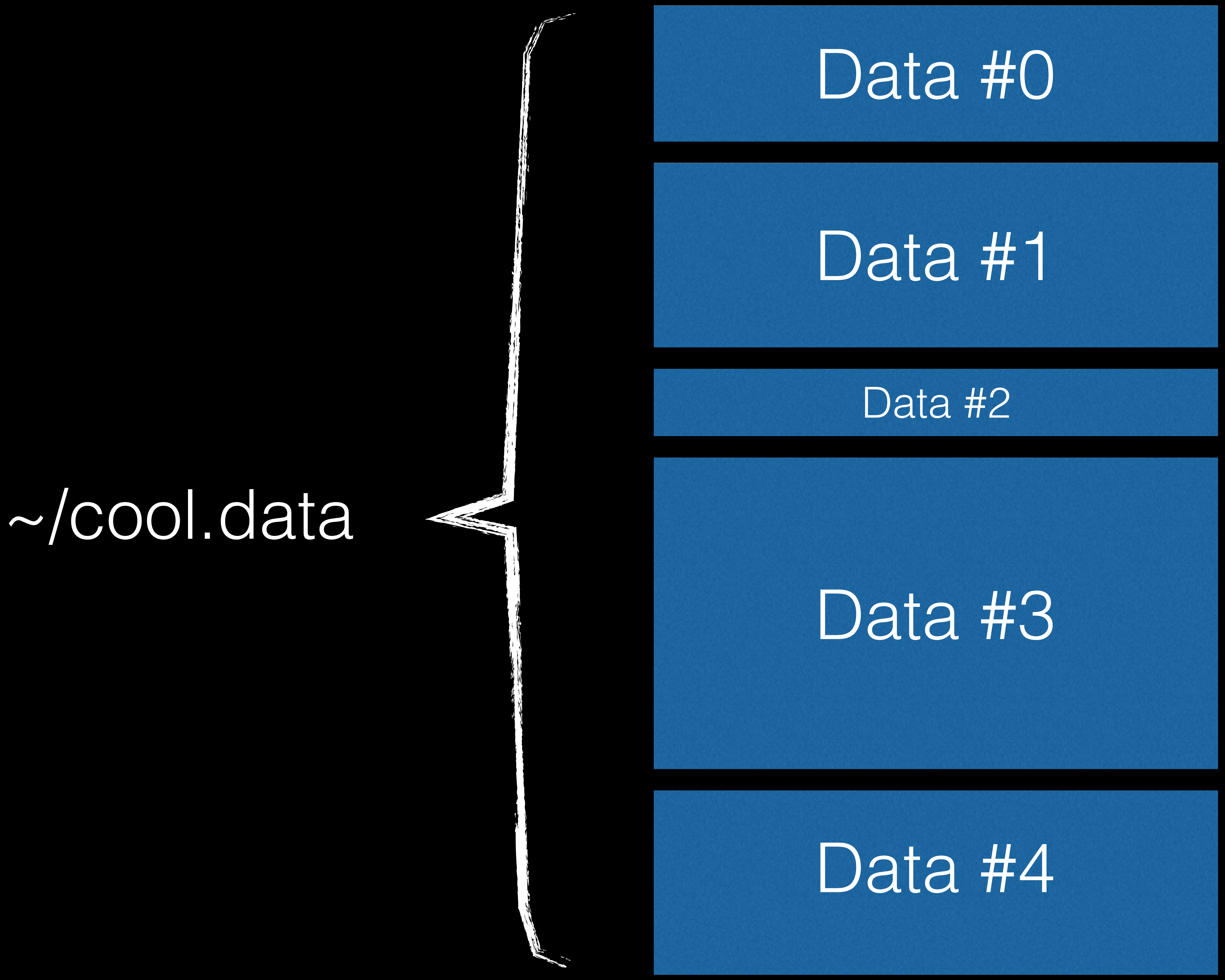
Take a file



Cut it into pieces



Insert each piece into the log



Diffable

Divide a file into chunks that are unlikely to change when the file is updated

Example: git

```
function hello () {
  var world = 'world'
  console.log('hello', world)
}
```

```
function hello () {
  var world = 'world'
  console.log('hello', world)
}
```

(One line per chunk)

```
function hello () {
  var world = 'universe'
  console.log('hello', world)
}
```

(Edit one line)

```
function hello () {
  var world = 'universe'
  console.log('hello', world)
}
```

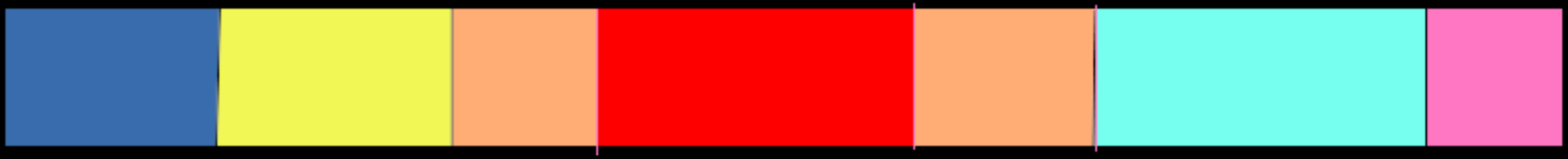
(3/4 chunks unchanged)

Only works for text files

Rabin fingerprinting

(Content defined chunking)

Scans through the file and creates chunks based on the actual file content

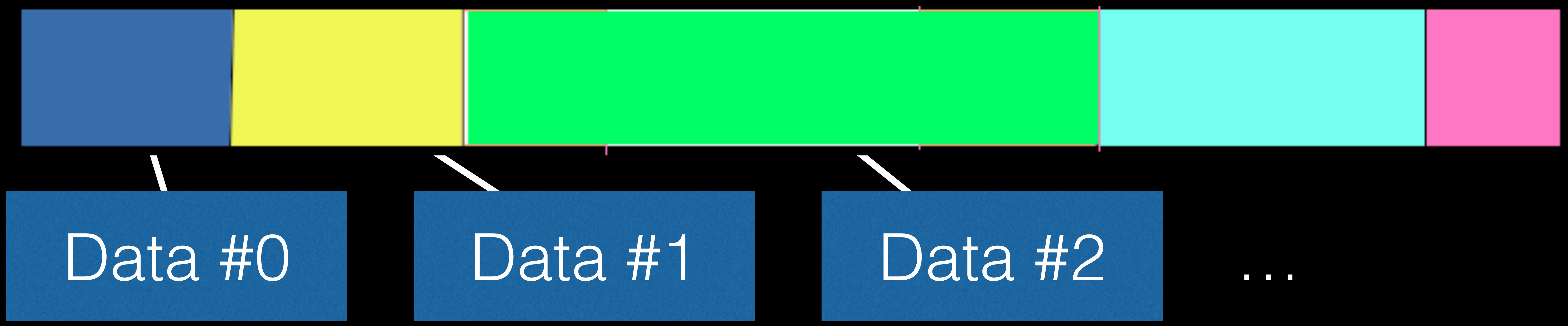


(A new part is inserted in the middle of the file)

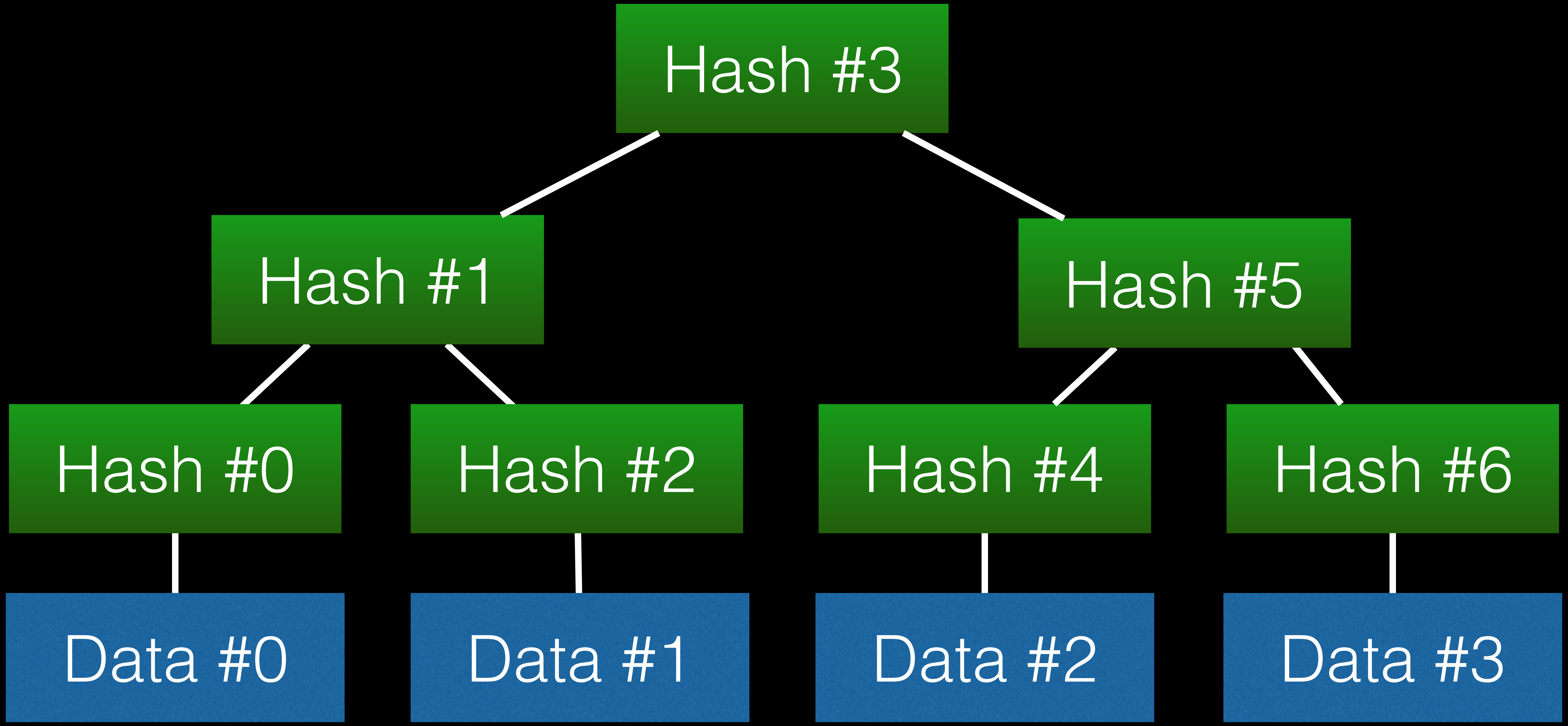
(Only the neighbouring chunks are changed)

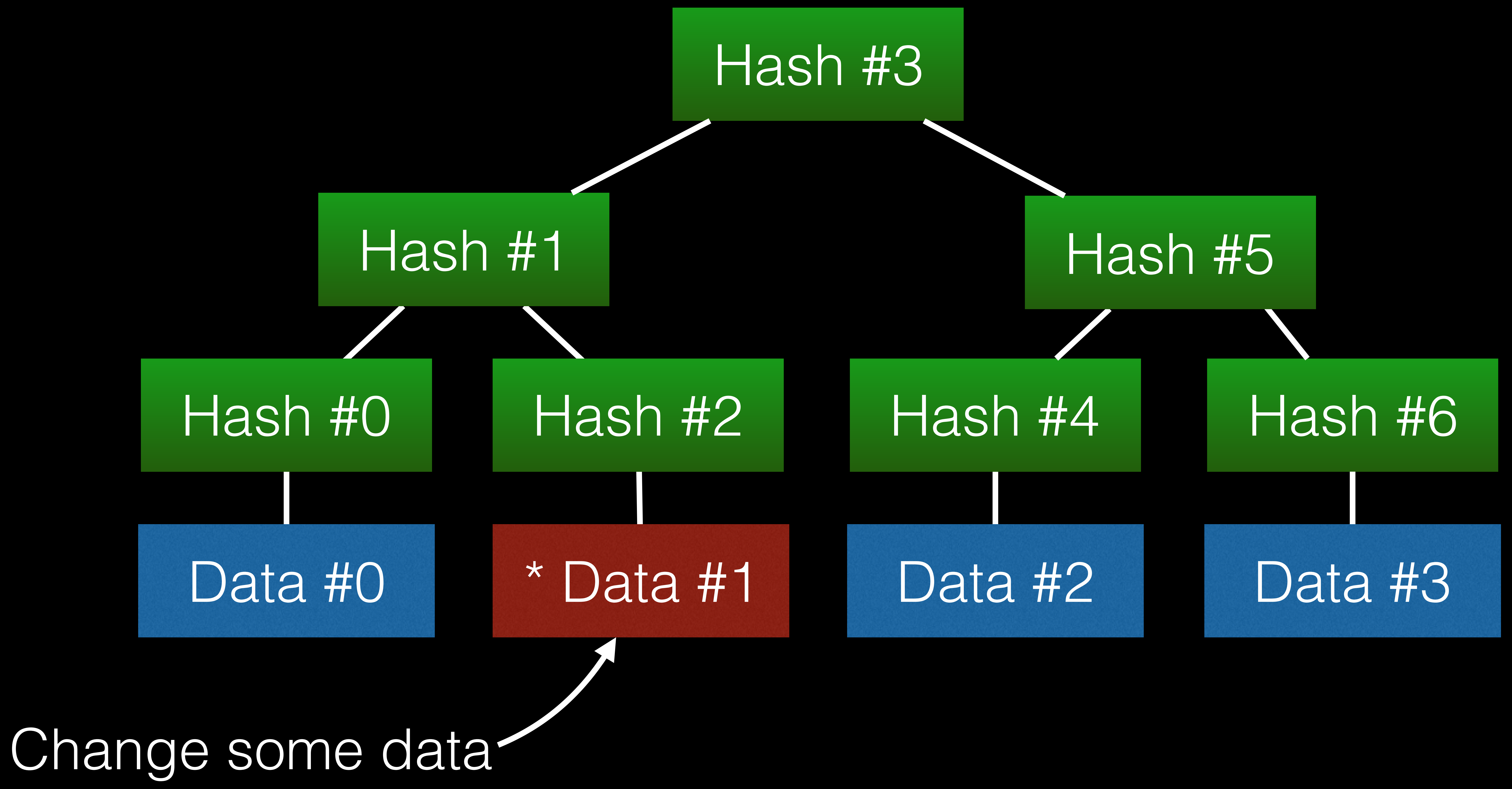
```
npm install rabin
```

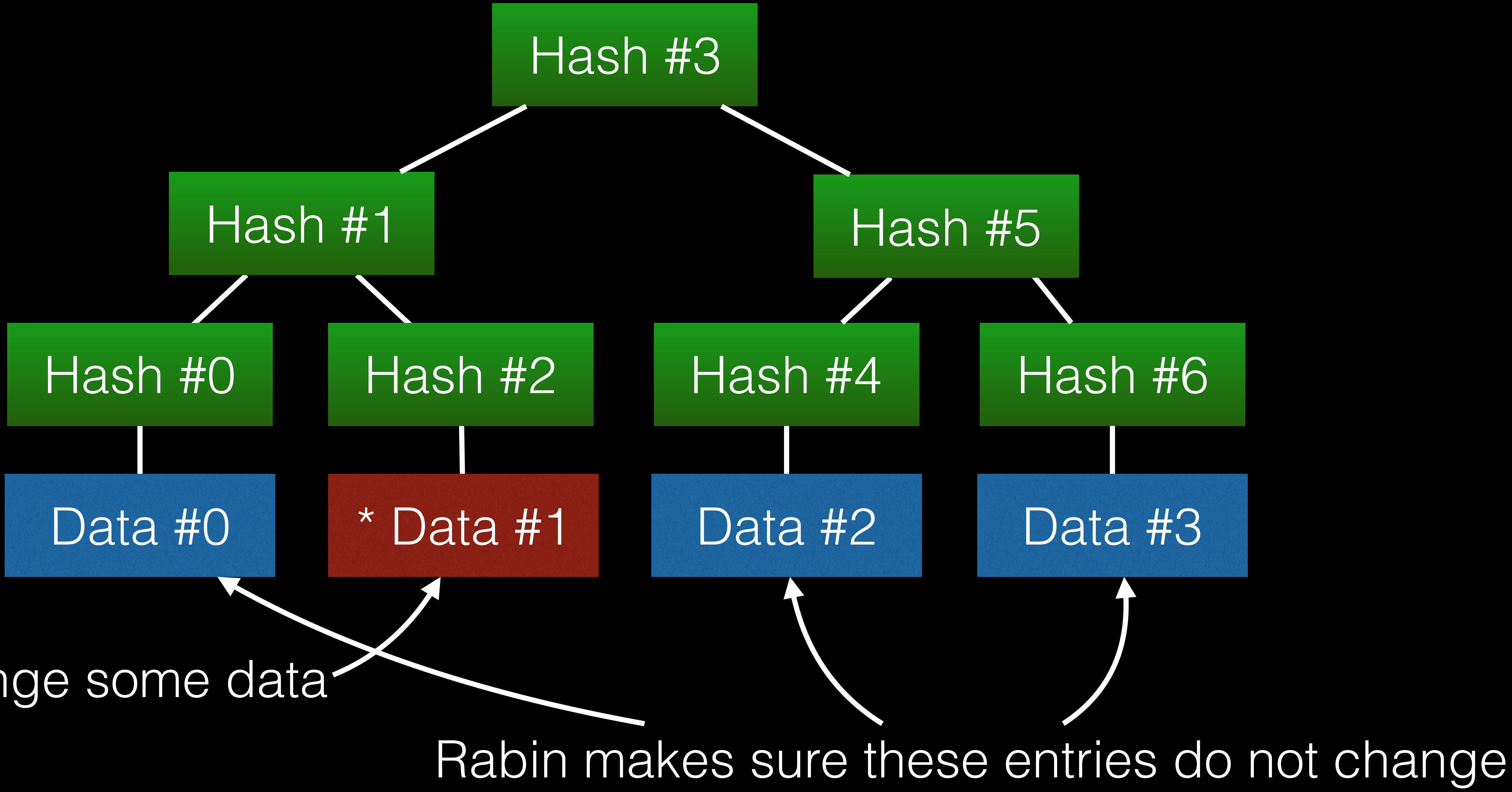
Each Rabin chunk is an entry in our append only log

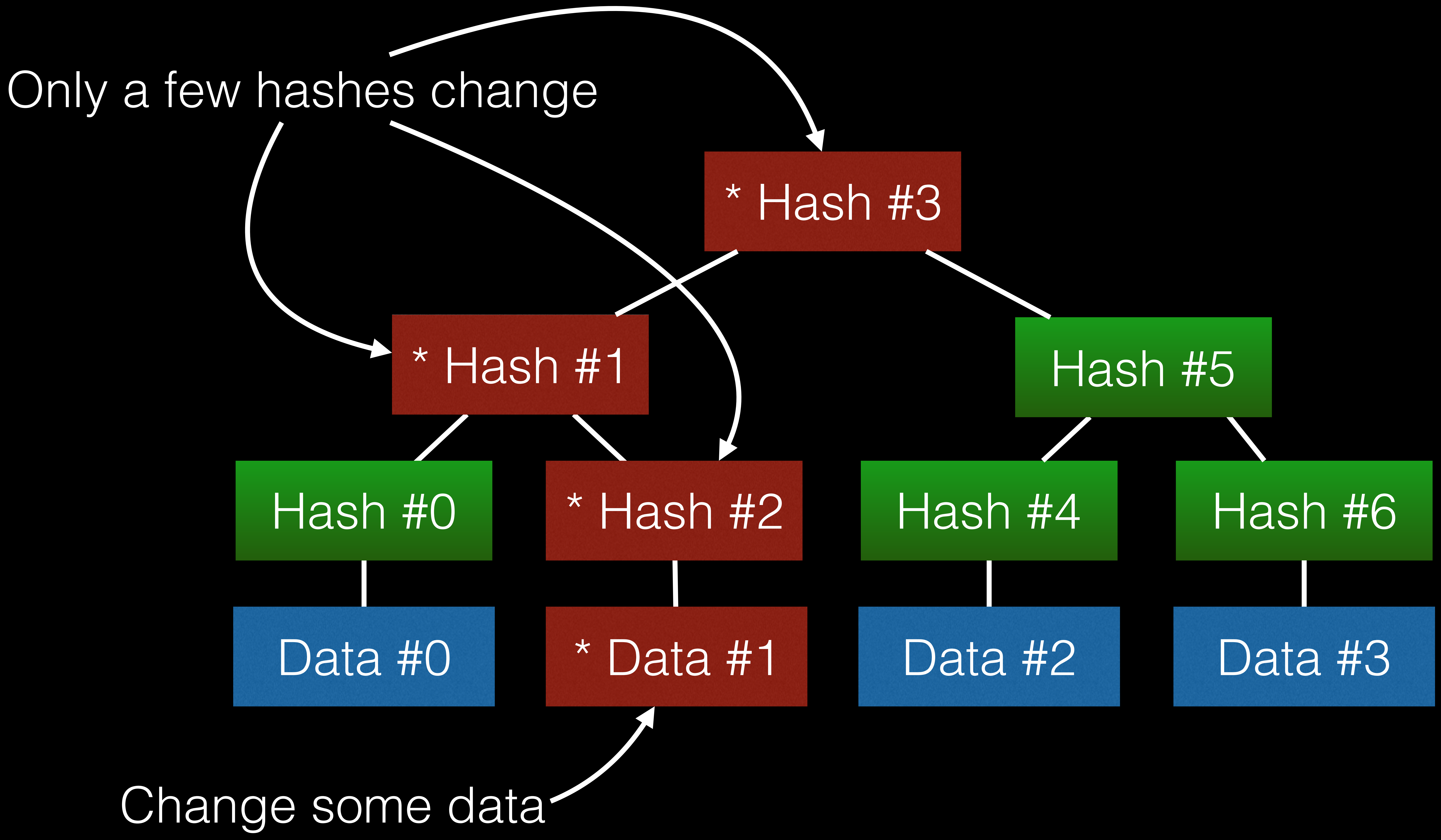

Data #0
Data #1
Data #2
…

Merkle trees + Rabin = ❤️

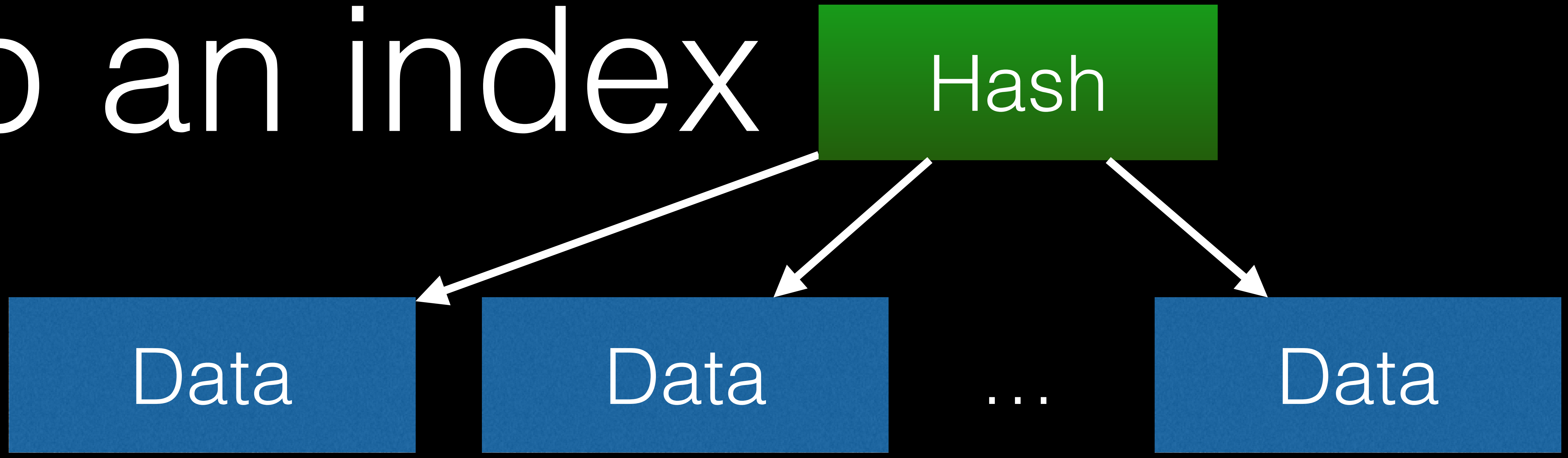

Hash #3
Hash #1
Hash #5
Hash #0
Hash #2
Hash #4
Hash #6
Data #0
Data #1
Data #2
Data #3


Hash #3
Hash #1
Hash #5
Hash #0
Hash #2
Hash #4
Hash #6
Data #0
* Data #1
Data #2
Data #3
Change some data


Hash #3
Hash #1
Hash #5
Hash #0
Hash #2
Hash #4
Hash #6
Data #0
* Data #1
Data #2
Data #3
Change some data
Rabin makes sure these entries do not change


Only a few hashes change
* Hash #3
* Hash #1
Hash #5
Hash #0
* Hash #2
Hash #4
Hash #6
Data #0
* Data #1
Data #2
Data #3
Change some data

Keep an index



See the same Hash twice, just copy the Data

See the same Hash twice, just copy the Data

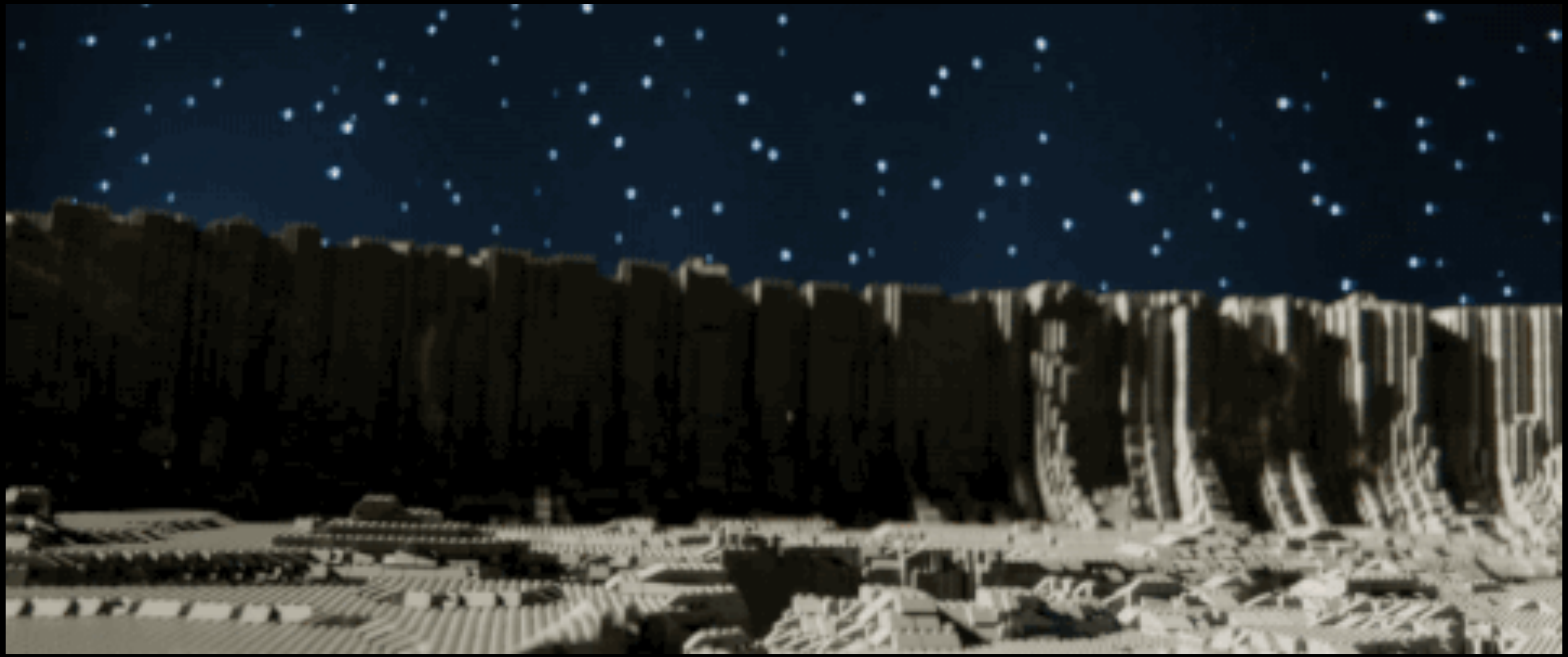
(no need to re-download it)

See the same Hash twice, just copy the Data

(no need to re-download it)

(can be … easily … optimised for space)

```
npm install hyperdrive
```



(demo)

dat
DATA

is a cli tool and desktop app that manages hyperdrives

(demo)

Great apps build on dat DATA

Beaker browser

https://github.com/beakerbrowser/beaker

Science Fair

https://github.com/codeforscience/sciencefair

Read our paper

https://github.com/datproject/docs/blob/master/papers/dat-paper.pdf

Thank you!

https://github.com/mafintosh/hypercore

https://github.com/maxogden/rabin

https://github.com/mafintosh/hyperdrive

https://github.com/datproject/dat