

Hyper Ecosystem

Emil Bay

@emilbayes

Copenhagen
Denmark 🇩🇰

Hyperdivison
Crypto(logy), Math
Dist. Sys.



Core Modules



```
npm install hypercore
```

Problem:

p2p data feed

```
const hypercore = require('hypercore')  
  
const feed = hypercore('./data')  
  
feed.ready(function (err) {  
  if (err) throw err  
  
  console.log(feed.key.toString('hex'))  
})
```

```
const hypercore = require('hypercore')  
  
const feed = hypercore('./data')  
  
feed.append(Buffer.from('hello world'))
```



```
const hypercore = require('hypercore')
```

```
const feed = hypercore('./data',  
{valueEncoding: 'json'})
```

```
feed.append({  
  symbol: 'BTCUSD',  
  ask: 7000,  
  bid: 6999  
})
```

```
const hypercore = require('hypercore')

const feed = hypercore('./data',
{valueEncoding: 'json'})

feed.get(2, function (err, data) {
  if (err) throw err
  console.log(data)
})
```

```
const hypercore = require('hypercore')

const key = Buffer.from('...', 'hex')
const feed = hypercore('./copy', key,
{valueEncoding: 'json'})

var stream = feed.createReadStream({live: true})

stream.on('data', data => console.log(data))
```

```
const hypercore = require('hypercore')

const key = Buffer.from('...', 'hex')
const feed = hypercore('./copy', key, {valueEncoding:
  'json'})

pump(
  transport,
  feed.replicate({live: true}),
  transport,
  function ondone (err) {
    if (err) throw err
  }
)
```

DEMO

```
const WebSocket = require('uws')
const hypercore = require('hypercore')
const swarm = require('hyperdiscovery')

const feed = hypercore('./ticker-data', {valueEncoding: 'json'})

feed.once('ready', () => console.log(feed.key.toString('hex')))
feed.once('ready', () => swarm(feed))

const sock = new WebSocket('wss://api.bitfinex.com/ws')

sock.onopen = function () {
  sock.send(JSON.stringify({
    event: 'subscribe',
    channel: 'ticker',
    pair: 'BTCUSD'
  })))
}

var dataChannel = null
sock.onmessage = function (raw) {
  var msg = JSON.parse(raw.data)

  console.log(msg)
  if (msg[0] === dataChannel && msg[1] !== 'hb') return feed.append([msg])
  if (msg.event && msg.event === 'subscribed') dataChannel = msg.chanId
}
```

**Public Key is
read capability**

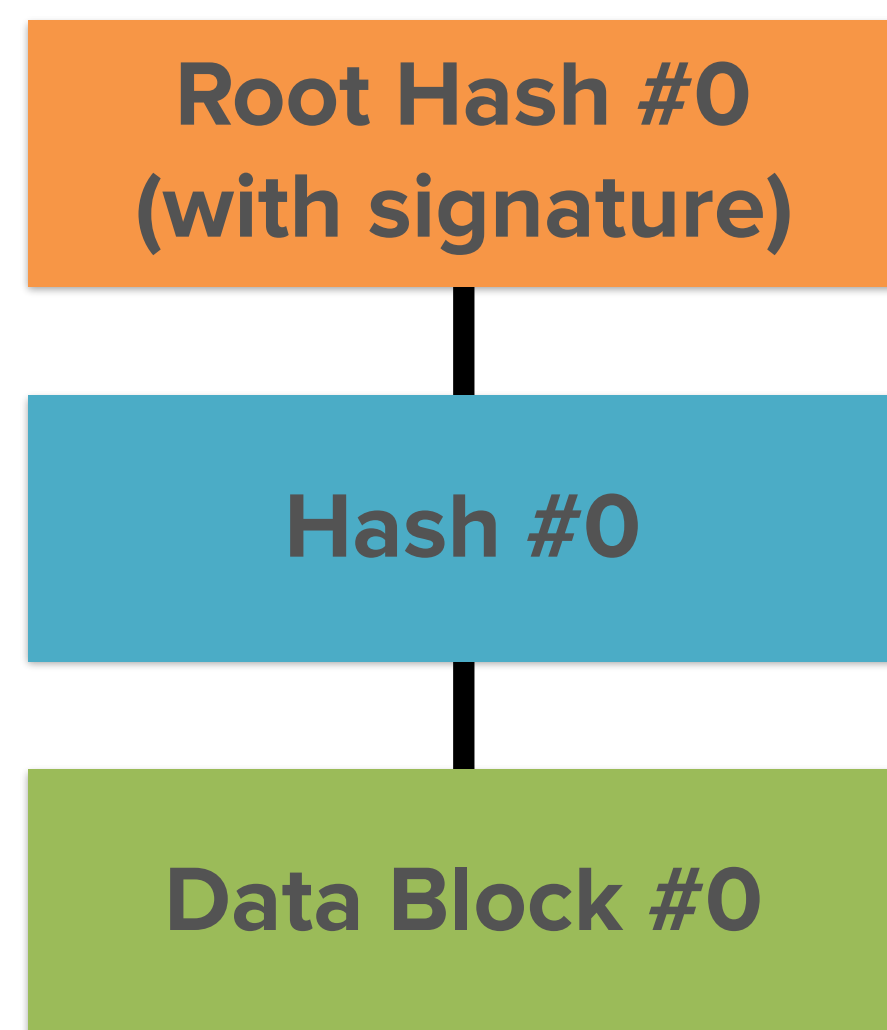
**Secret Key is
write capability**

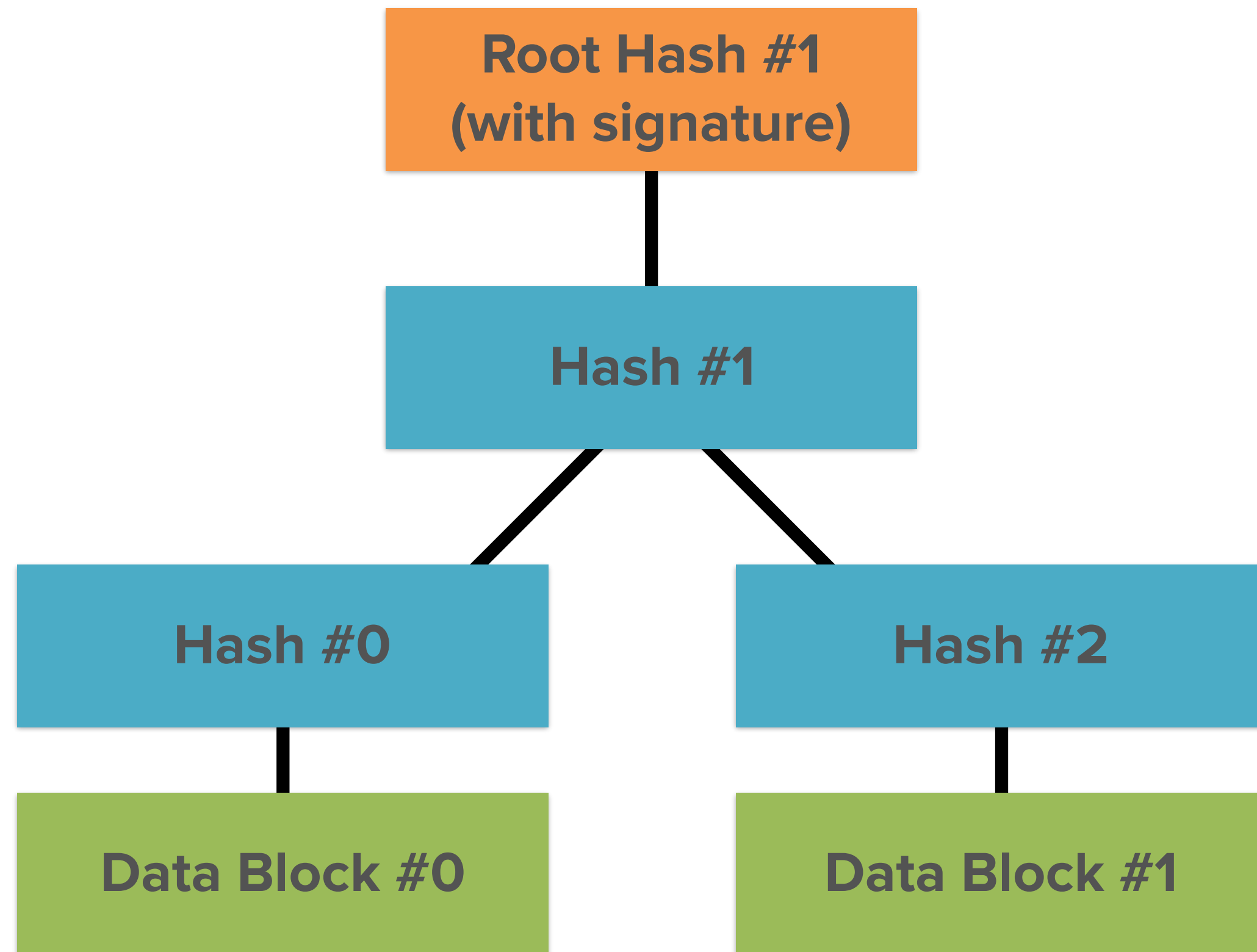
Still single-writer

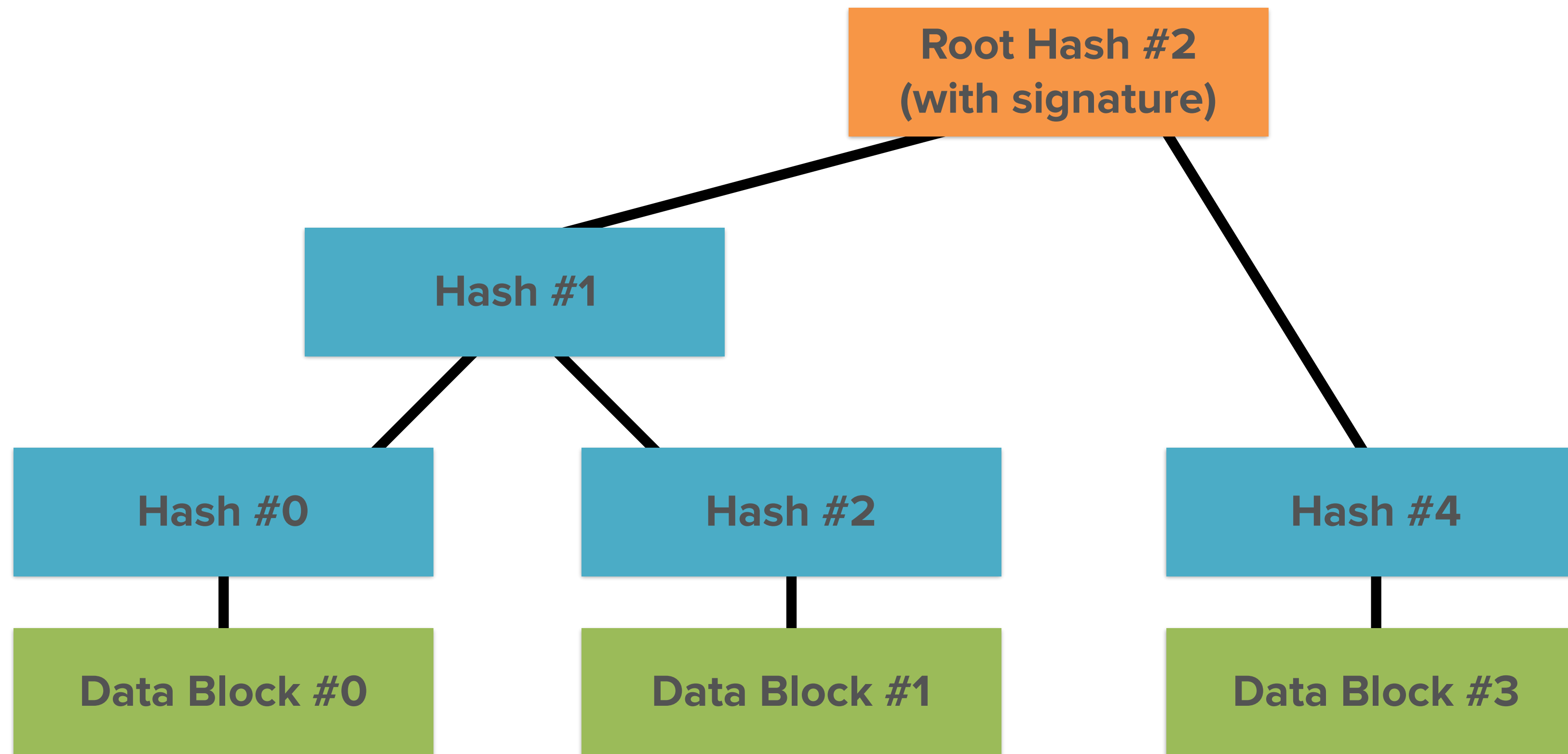
Authentication with Merkle Tree

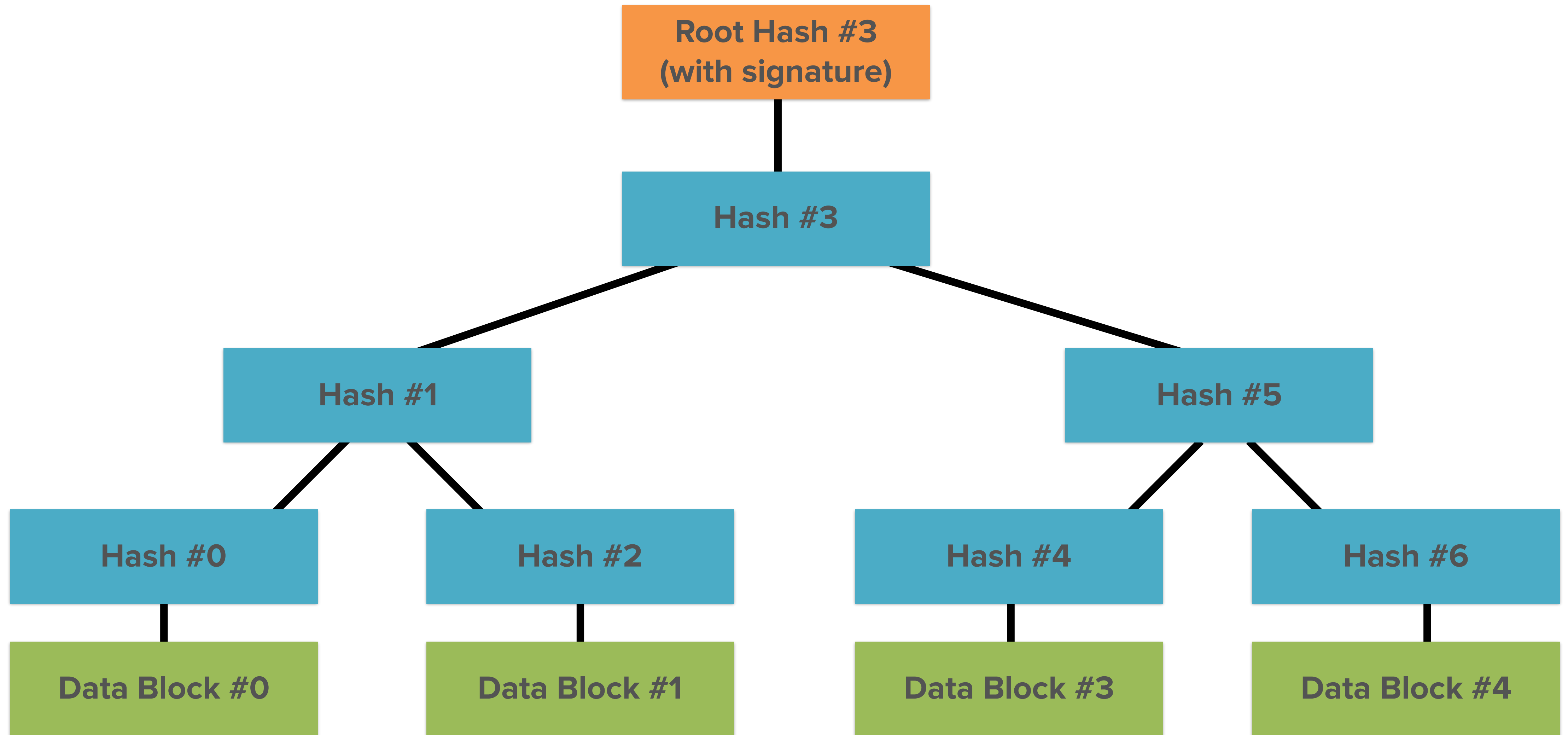


Data Block #0









**Means any peer
can copy for any
other peer**

```
npm install hyperdiscovery
```

```
const hypercore = require('hypercore')
const hyperdiscovery = require('hyperdiscovery')

const key = Buffer.from('...', 'hex')
const feed = hypercore('./copy', key,
{valueEncoding: 'json'})

feed.ready(function () {
  hyperdiscovery(feed)
})
```

Announce **b1ake2b(pk, 'hypercore')** **on Mainline DHT**

Broadcast on mDNS

(locally and predefined DNS servers)

npm install hyperdb

```
const hyperdb = require('hyperdb')
const hyperdiscovery = require('hyperdiscovery')

const db = hyperdb('./my.db', {
  valueEncoding: 'utf8',
  reduce: (a, b) => a // conflict resolution
})

hyperdiscovery(db)

db.put('/hello', 'world', function (err) {
  if (err) throw err
  db.get('/hello', function (err, node) {
    if (err) throw err
    console.log('/hello --> ' + node.value)
  })
})
```

```
var hyperdb = require('hyperdb')

var db = hyperdb('./my.db', {valueEncoding: 'utf-8'})

db.ready(function () {
  var aliceKey = db.key
})

db.authorize(bobKey, function (err) {
  if (err) throw err

  // Bob can now also an admin to the database
  // This trust is propagated to peers
})
```



```
var hyperdb = require('hyperdb')

var db = hyperdb('./my.db', {valueEncoding:
  'utf-8'})

db.list('/tokens', function (err, tokens) {
  console.log(tokens)
})
```

Path like keys
/token/btc/ticker

**HAMT makes
random-access on keys
efficient**

Multiwriter!

Applications

Hyperpipe

Pipe logs into a hypercore

```
$ ./server | hyperpipe /var/logs/server  
KEY
```

Pipe into a local log file

```
$ hyperpipe /tmp/hyperpipe KEY > logs.txt
```

```
var hypercore = require('hypercore')
var hyperdiscovery = require('hyperdiscovery')
var dataDir = process.argv[2]
var key = process.argv[3]
var feed = hypercore(dataDir, key)

feed.on('ready', function () {
  var swarm = hyperdiscovery(feed)

  if (!feed.writable) {
    pump(feed.createReadStream({live: true}), process.stdout, ondone)
  } else {
    console.error(feed.key.toString('hex'))
    pump(process.stdin, feed.createWriteStream(), ondone)
  }

  ondone (err) {
    swarm.close()

    if (err) {
      console.error(err)
      process.exit(1)
    }
  }
})
```


Hyperclock

```
var hypercore = require('hypercore')
var sodium = require('sodium-native')
var feed = hypercore('./hyperclock', {valueEncoding: 'json'})

var nonce = Buffer.alloc(32)
var interval = 5 * 60 * 1000 // 5 min
setInterval(function () {
  sodium.randombytes_buf(nonce)
  feed.append({
    ts: Date.now(),
    nonce: nonce.toString('hex')
  })
}, interval)
```

Hypervision

```
const hypercore = require('hypercore')
const recorder = require('media-recorder-stream')
const cluster = require('webm-cluster-stream')
const pump = require('pump')

const feed = hypercore('./hypervision')

// media instanceof MediaStream
const mediaRecorder = recorder(media, {
  interval: 1000,
  videoBitsPerSecond: 800000,
  audioBitsPerSecond: 128000
})

pump(
  mediaRecorder,
  cluster(),
  feed.createWriteStream(),
  function ondone () {}
)
```

Cabal

Hyperdrive

Hypercore-dag

Hypertrie

Append-tree

Multifeed

<https://github.com/emilbayes/bitfinex-hackathon-2018>

<https://github.com/emilbayes/bitfinex-hackathon-2018>