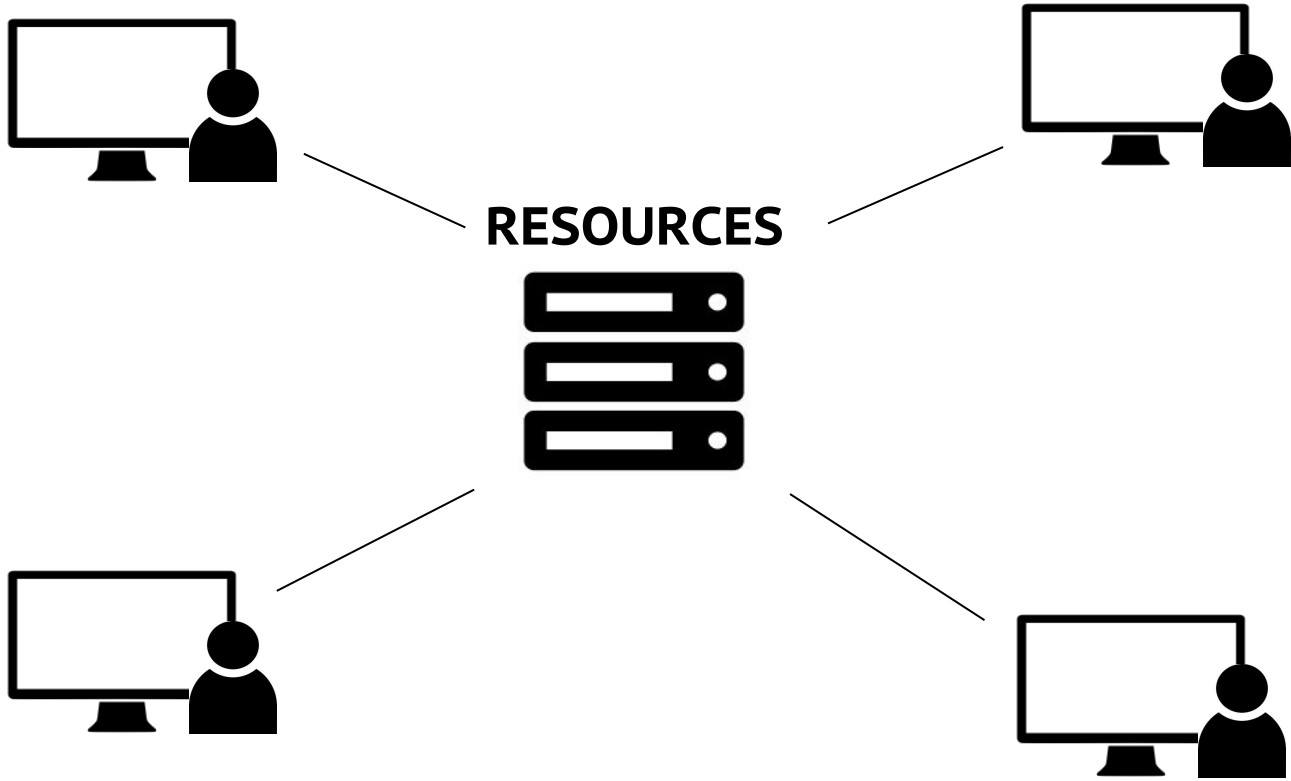


# How Merkle trees enable the decentralized Web!

---

@taravancil

taravancil.com



Host-based addressing


youtube.com/myvideo

# Host-based addressing

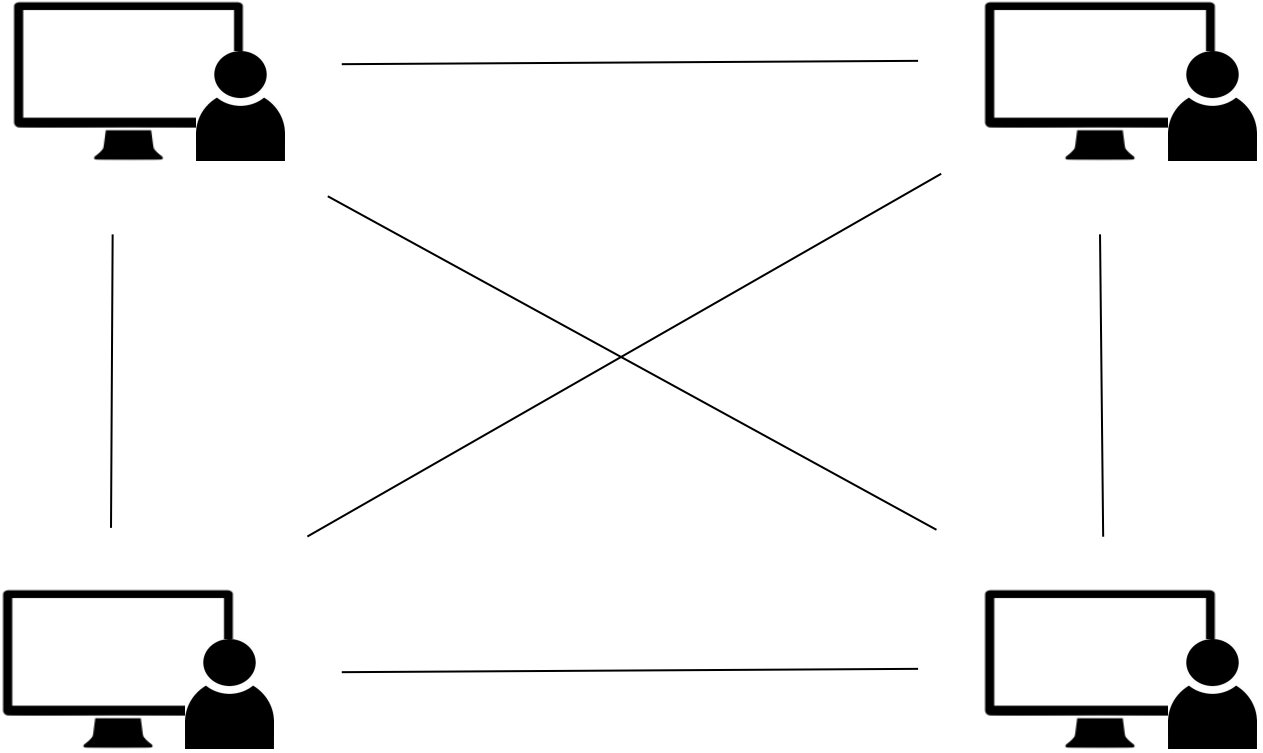
`youtube.com/myvideo -> vimeo.com/myvideo`

# Content addressing - hash functions

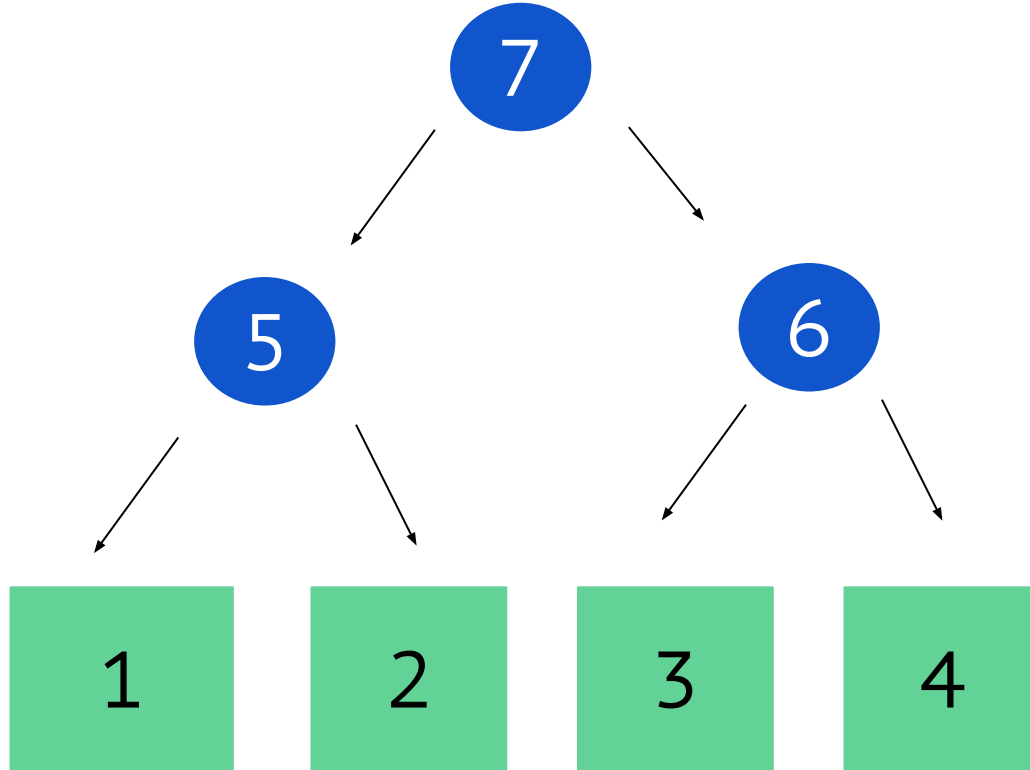
# Content addressing - hash functions

hash() ->


2cf24dba5fb0a30e26e83b2ac5  
b9e29e1b161e5c1fa7425e7304  
3362938b9824

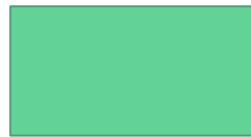


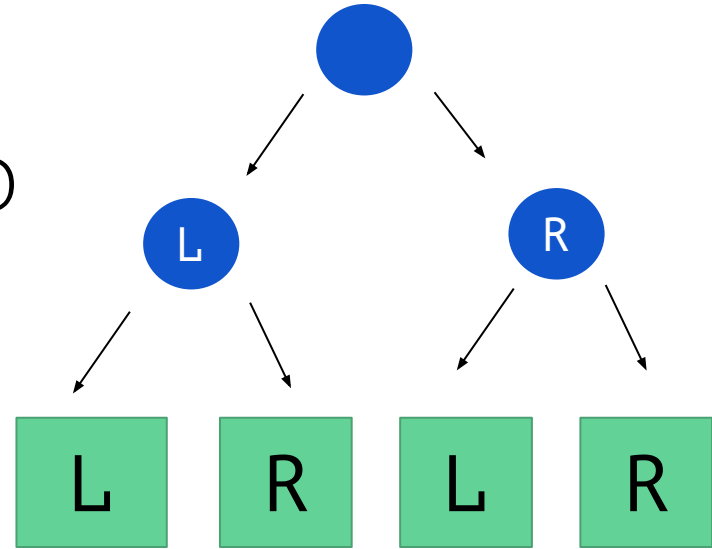
# Regular binary tree



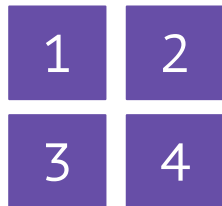


 = hash(left, right)

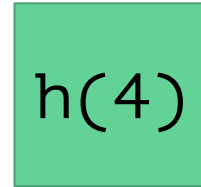
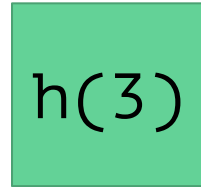
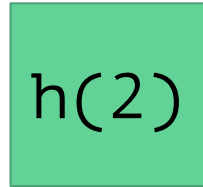
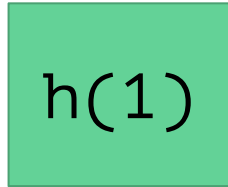
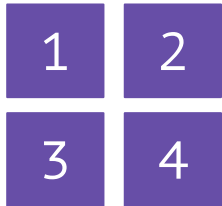
 = hash(data)



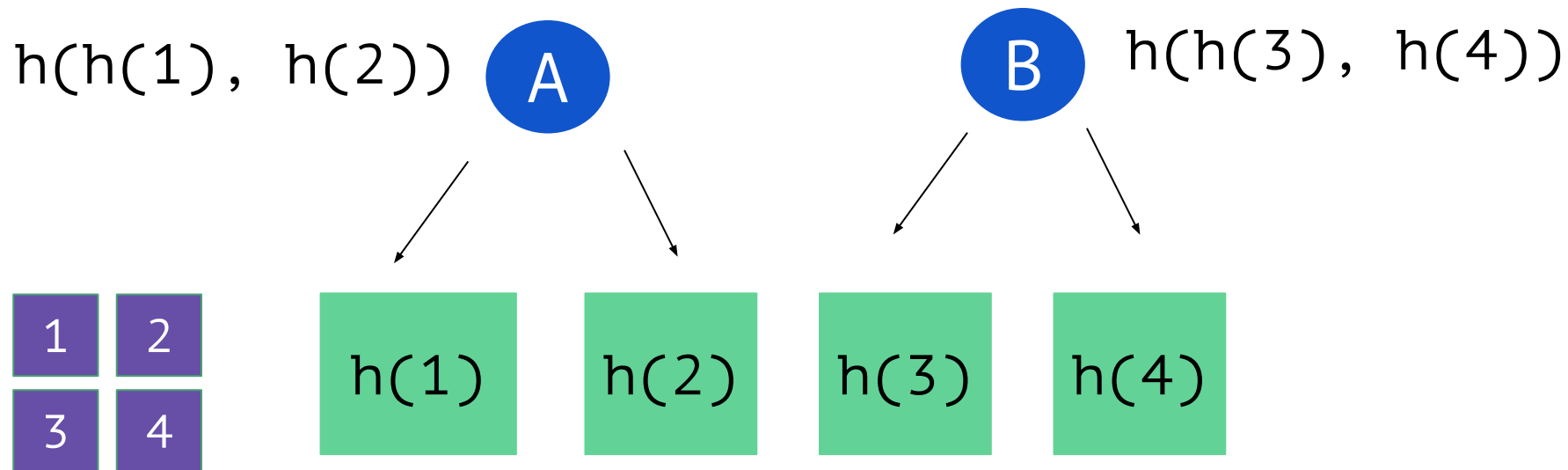
# Merkle tree



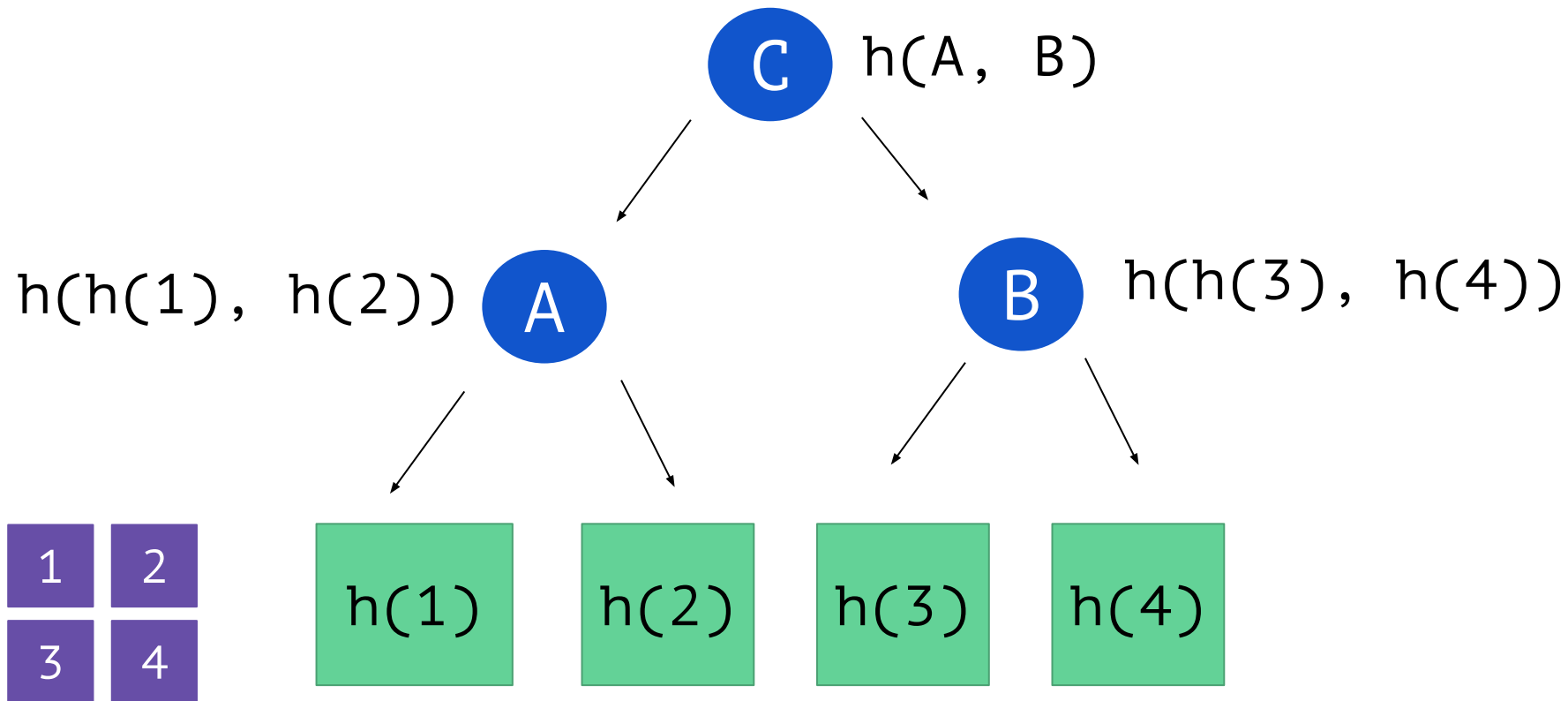
# Merkle tree



# Merkle tree



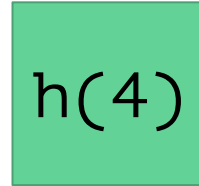
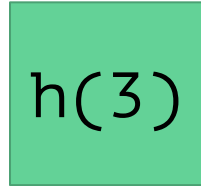
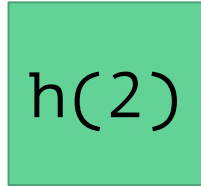
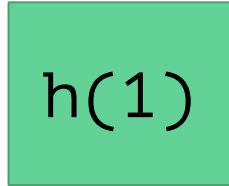
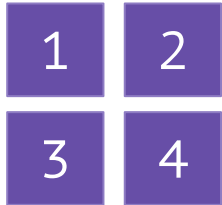
# Merkle tree



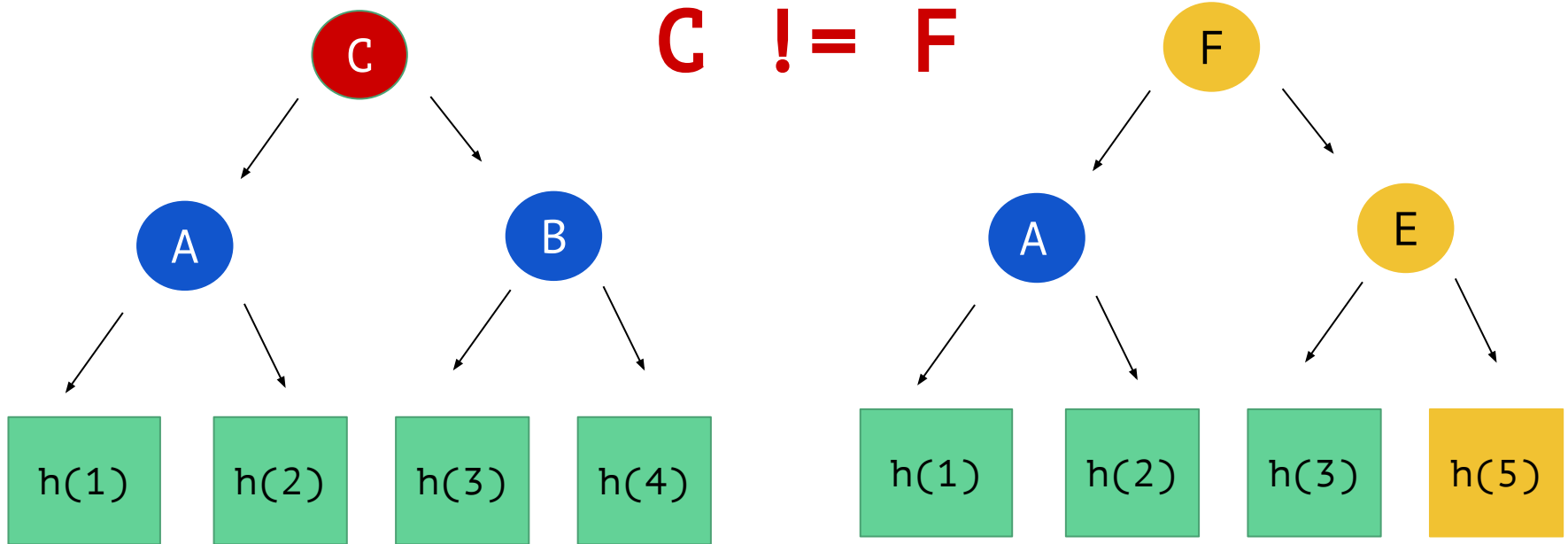
# Merkle tree

Root hash  $\rightarrow$  **C**  $h(A, B)$

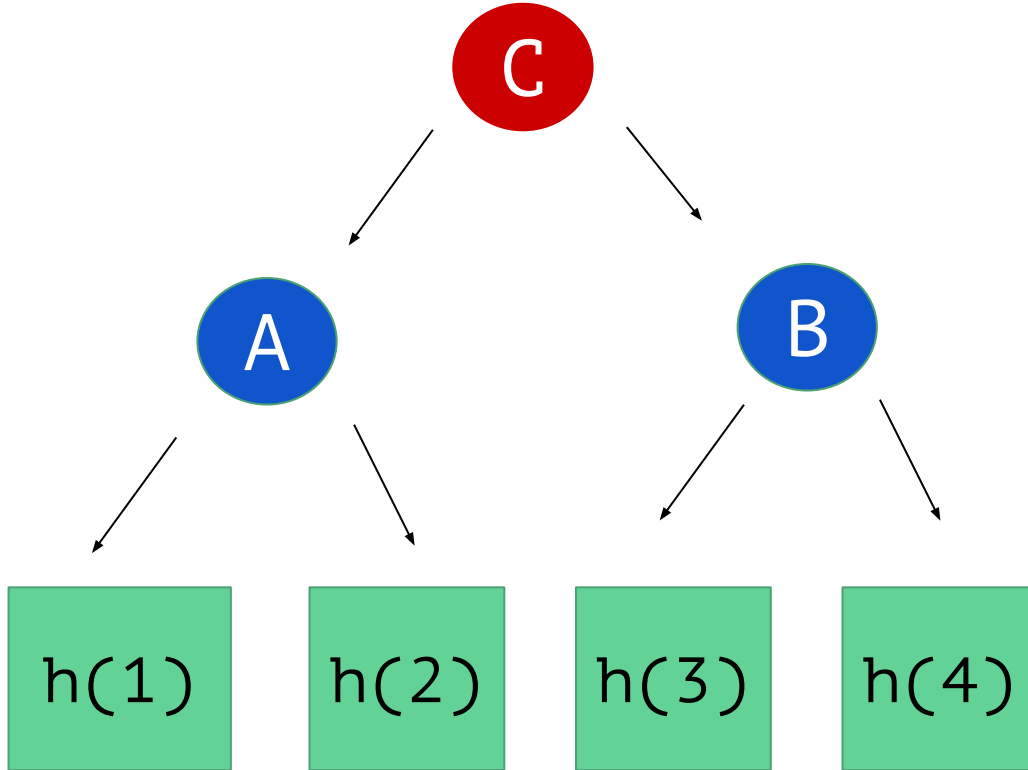
$h(h(1), h(2))$  **A**  $h(h(3), h(4))$  **B**



# Checking for equality




C



1	2
3	4



What if?

hash(     )

=

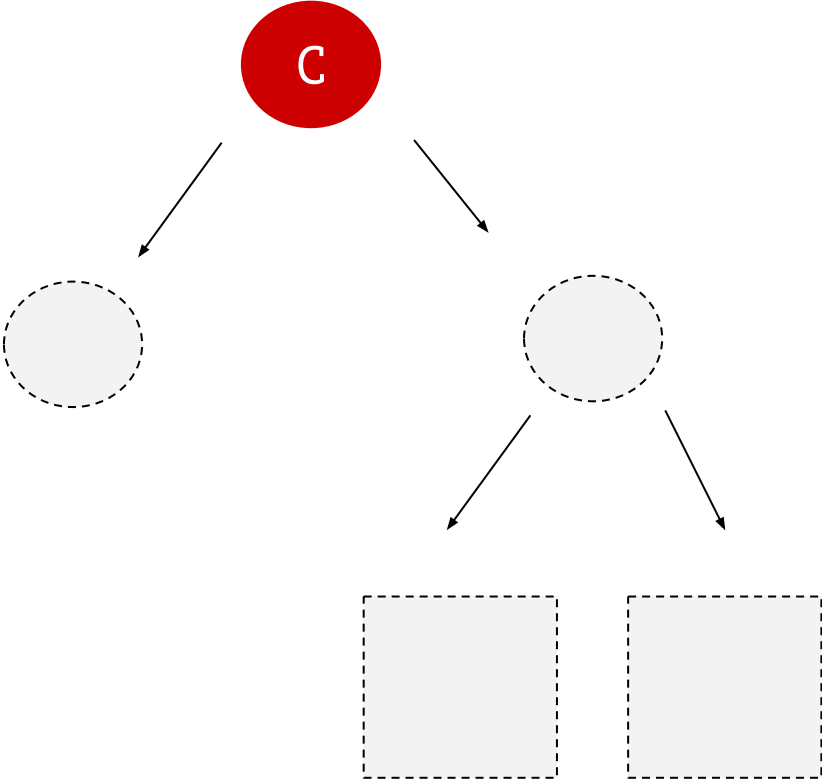
Root hash

**partial verification**



**UNTRUSTED PEER**

**RECIPIENT**

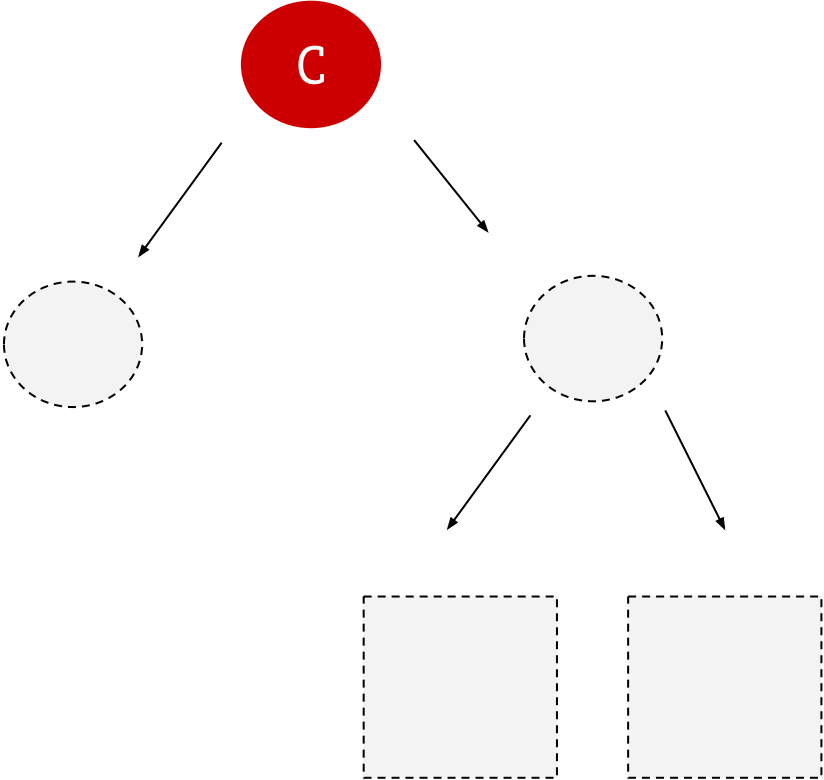


**UNTRUSTED PEER**

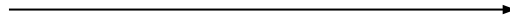


**RECIPIENT**

4

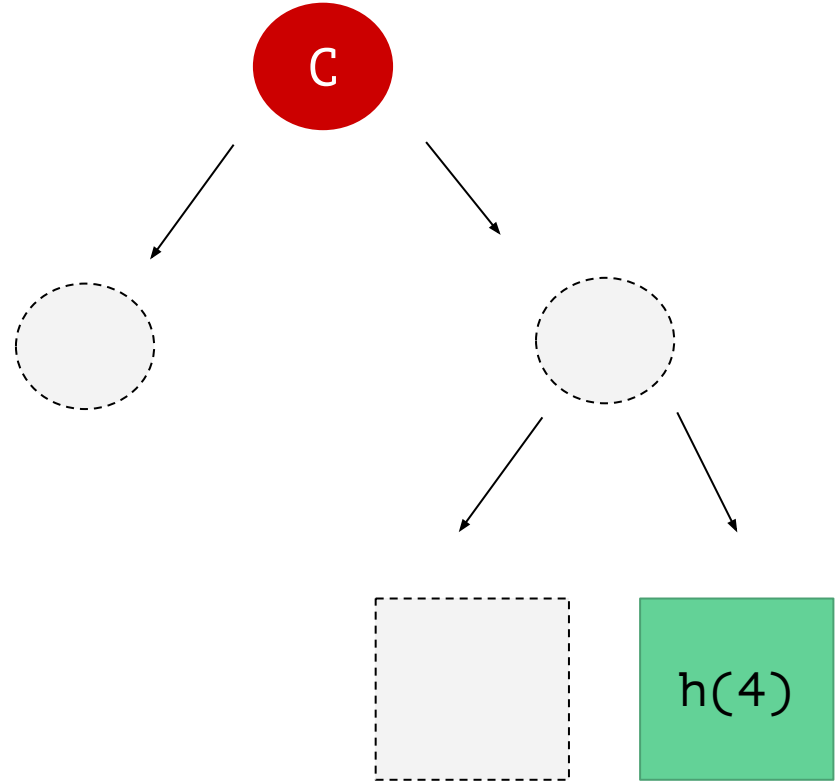


UNTRUSTED PEER



RECIPIENT

4



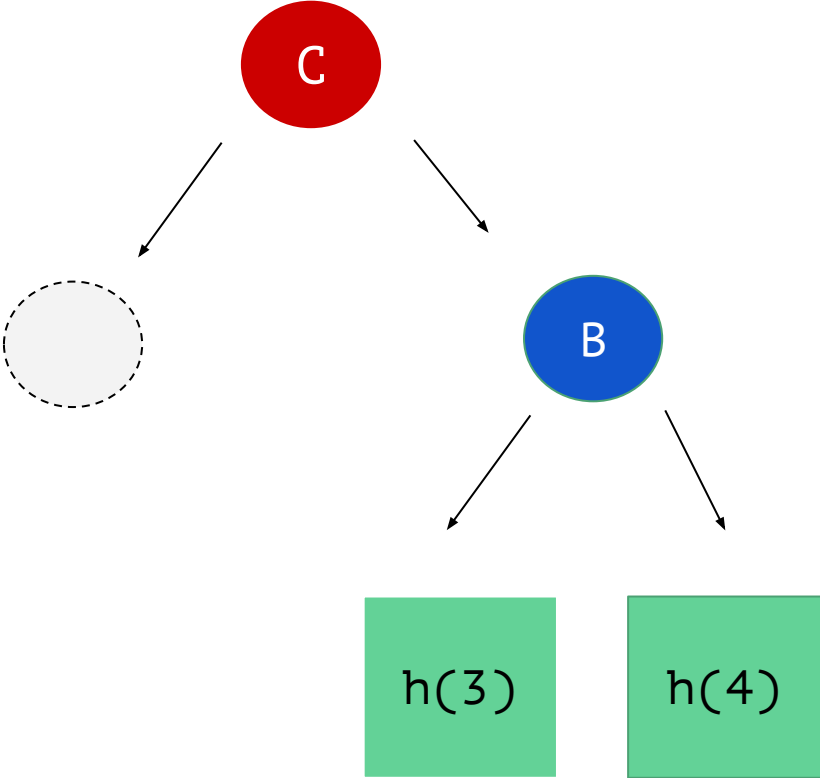
**UNTRUSTED PEER**



**RECIPIENT**

4

$h(3)$



**UNTRUSTED PEER**

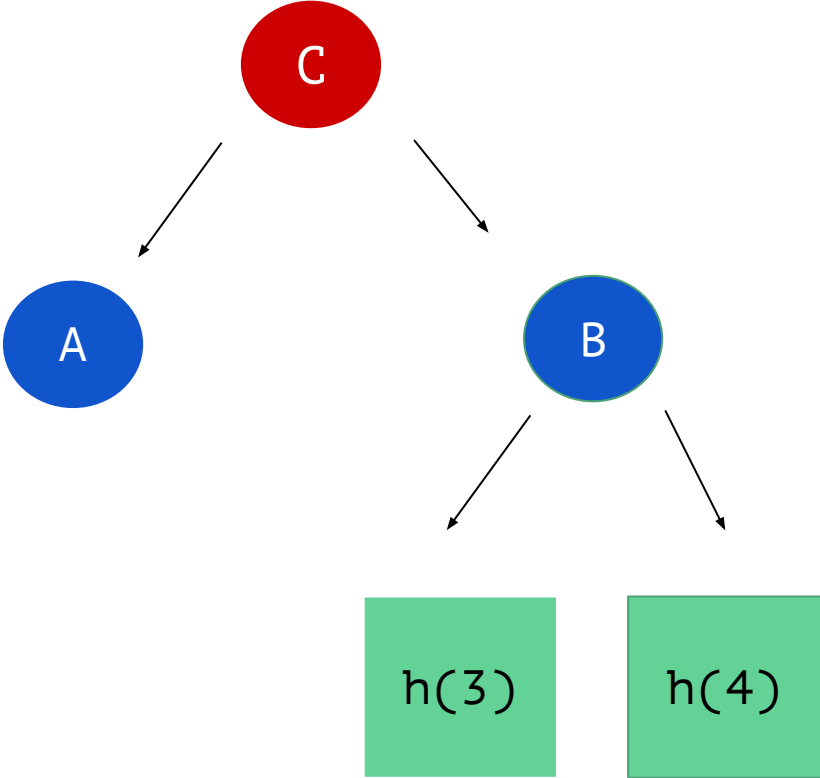


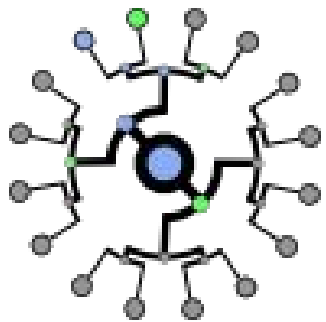
**RECIPIENT**

4

$h(3)$

A





 **riak**



ethereum



IPFS

Tahoe-LAFS 



*cassandra*



***bitcoin***



**thanks!**

@taravancil