

Methods of inclusion:

shift (Eq: 93)
insert (Eq: 73)

append

[C:3, h:10, a:1 ...]
What method is most friendly toward least amount of reshakes?

constant size
Sha3

constant size
Sha3

Accumulator
data
Grows with # of users

Can cause a temporal bandwidth backlog, e.g. 5 txs

Submit (Hash prev, Hash next, State prev, Transfer[] txs)

Merkle Proof
apply (Transfer[] txs, State prev)
== Hash next ✓

Batch #1 [] Oxa
Batch #2 [] Oxb

a:1
b:2
:
z:7

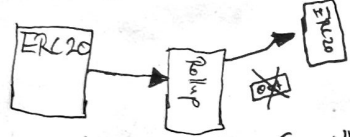
a → b:1
c → d:3
e → z:5

Consolidation



Can we design an incentive for it?
- tx fee: bytes/ETH

"Batch chain"



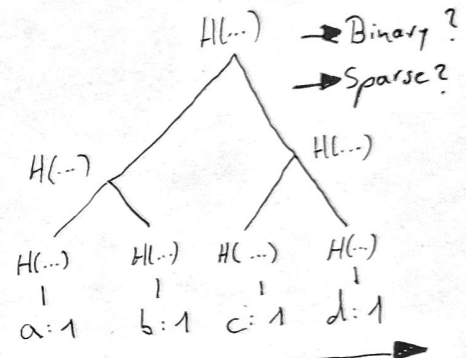
thinking in
UTXO is easier

UTXO?

a:1
b:2

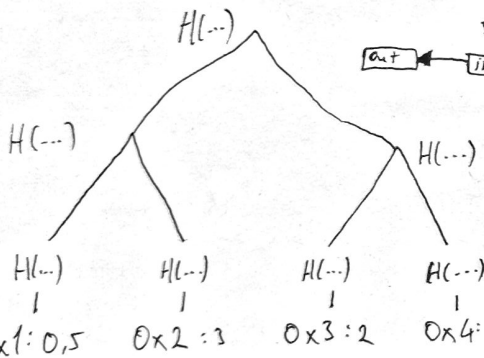
Account-based?

thinking in
accounts is more interesting



Criteria for establishing Order:

- As little reshaking as possible
- Order should NOT move leaves
- Order can only arise through implicit data e.g. balance but not e.g. transfer frequency.



Order criteria are similar to accounts: In what way is the least amount of reshaking necessary?

Handling deposits and withdrawals becomes easier.

First balance
d:15
a:13
z:9
e:1

Last access
1s f:2
2min z:5
3min b:3
1h c:13

Lexical and reverse
a:2
b:5
c:3
d:2
e:13

Lowest Balance
g:0
m:0
L:1
b:3

Control over
utxo.filter(0 → 0.val < 0.5)
size of utxo tree
activity filter?
active | inactive