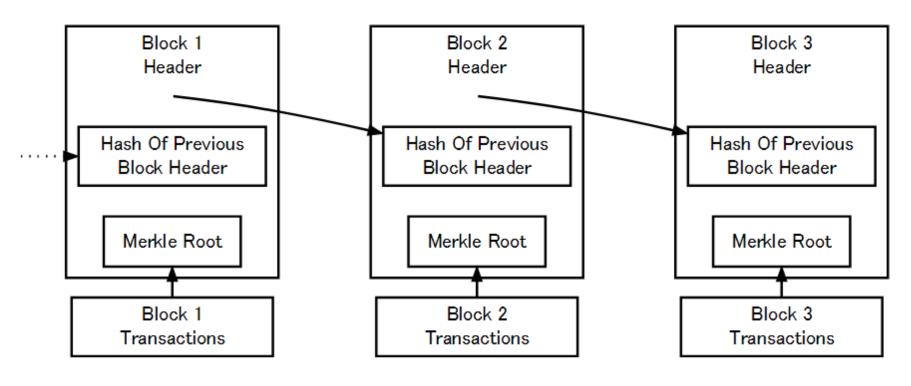
Bitcoin ブロックチェーン概観

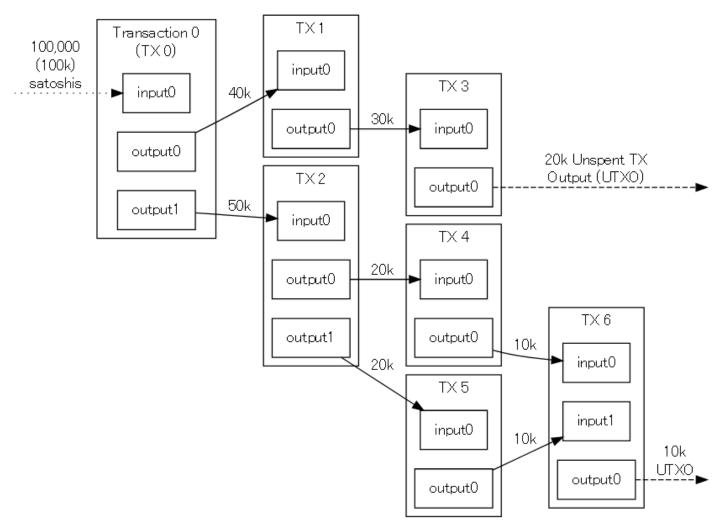
https://bitcoin.org/en/developer-guide#block-chain



Simplified Bitcoin Block Chain

Bitcoin ブロックチェーン概観

https://bitcoin.org/en/developer-guide#block-chain



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

UTXO?

- ブロックチェーンには大きく2つのデータモデルが存在する。
- 1つ目はアカウントに直近のデータをぶら下げるデータ格納方法。 Ethereum の State がそれ。
- 2つ目は UTXO (Unspent Transaction Output) これはトランザクションのみを格納する方法。
 代表例としては Bitcoin

例えば、①アドレスAからBへ10BTC、 ②アドレスBからCへ3BTC の順で移動させるとき、

二つの取引後のアドレスBの残高は7BTCである。

UTXOは最後の7BTCをデータとして格納するわけではなく、

①、②のトランザクションのみを格納している。

どちらのデータモデルが優れているということも無く、適材適所である。

UTXOモデルを利用した方が、アドレスの再利用が防げることによる匿名性の確保など、メリットがある。 その反面、実装が複雑になるデメリットがある。

参考: Ethereum における State

https://github.com/ethereum/wiki/wiki/[Japanese] -White-Paper

State

14c5f8ba:

- 1024 eth

bb75a980:

- 5202 eth

if !contract.storage[tx.data[0]]:
 contract.storage[tx.data[0]] = tx.data[1]

[0, 235235, 0, ALICE

892bf92f:

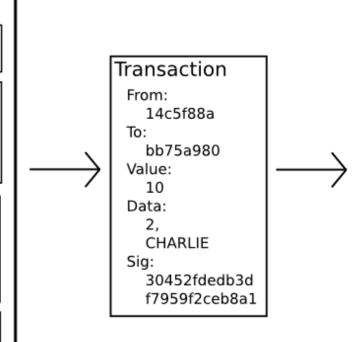
- 0 eth

send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])

[ALICE, BOB, CHARLIE]

4096ad65:

- 77 eth



State'

14c5f8ba:

- 1014 eth

bb75a980:

- 5212 eth

if !contract.storage[tx.data[0]]:
 contract.storage[tx.data[0]] = tx.data[1]

[0, 235235, CHARLIE, ALICE ..

892bf92f:

- 0 eth

send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])

[ALICE, BOB, CHARLIE]

4096ad65:

- 77 eth