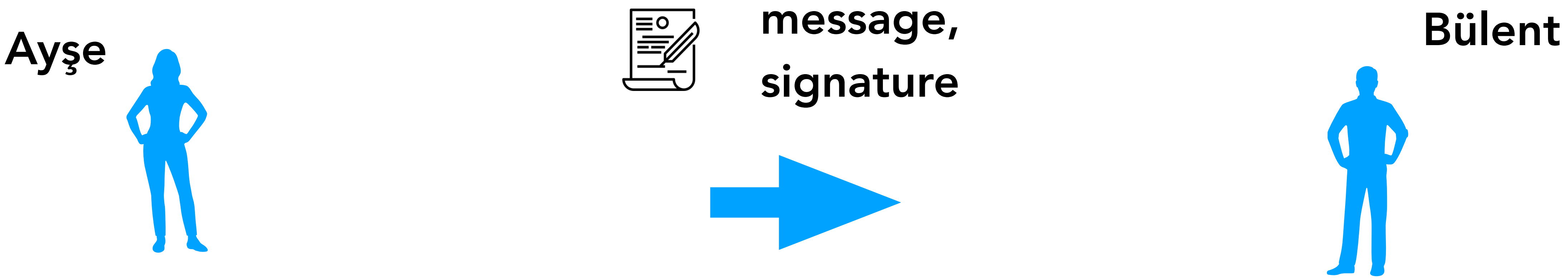
Public Key Cryptography & Digital Signatures

1. Ayşe generates public, private key pair
2. Ayşe signs a message with her private key
3. Ayşe broadcasts the message along with the signature
4. Bülent checks if Ayşe is signed the message with her public key



Ayşe

message, signature

Bülent

```
Sign(message,priv) = signature
```

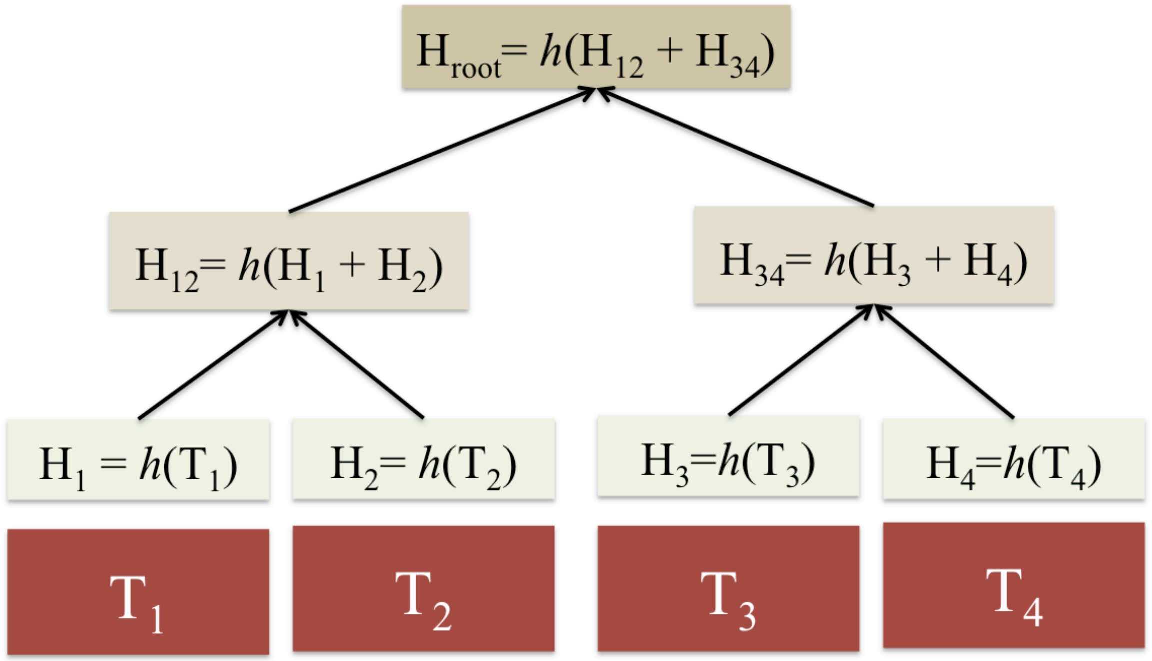
```
Recover(message,signature) = pub

isEqual(pub,pubKeyAyşe)
```

Hashes

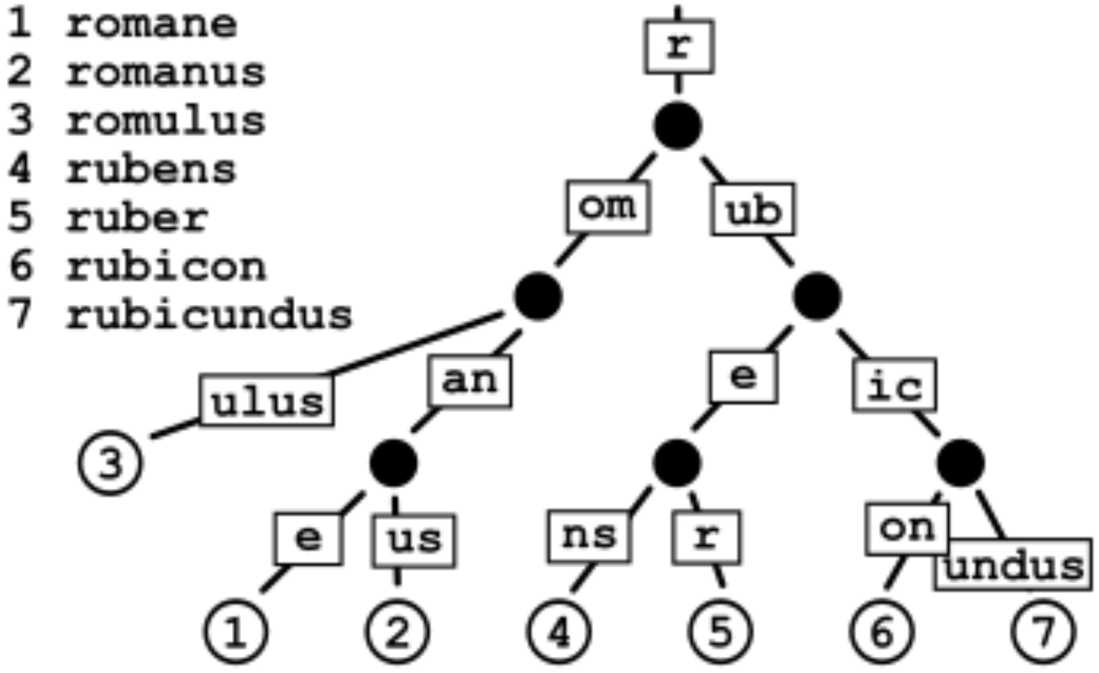
```
Hash("xyz 222 7a")
            f5836075b97a302bd33c3839c0a356dc5ea50a08d0afc11a5a4c36d66855c2a5

Hash("xyz 222 8a")
            29a8cfde6c240701b0e0d33309544b8cd3744b1a581875ca0e3aa022da793590

Hash("xyz 222 7a abc xxx uuu  eee 1999 10")
             1389bec4af924b8fff07edf489d1bdaf03a8534838dd0f02c37b4520d1ccd57a
```

One way function.
Fixed size output.
Identity of a piece of data.

Merkle Tree

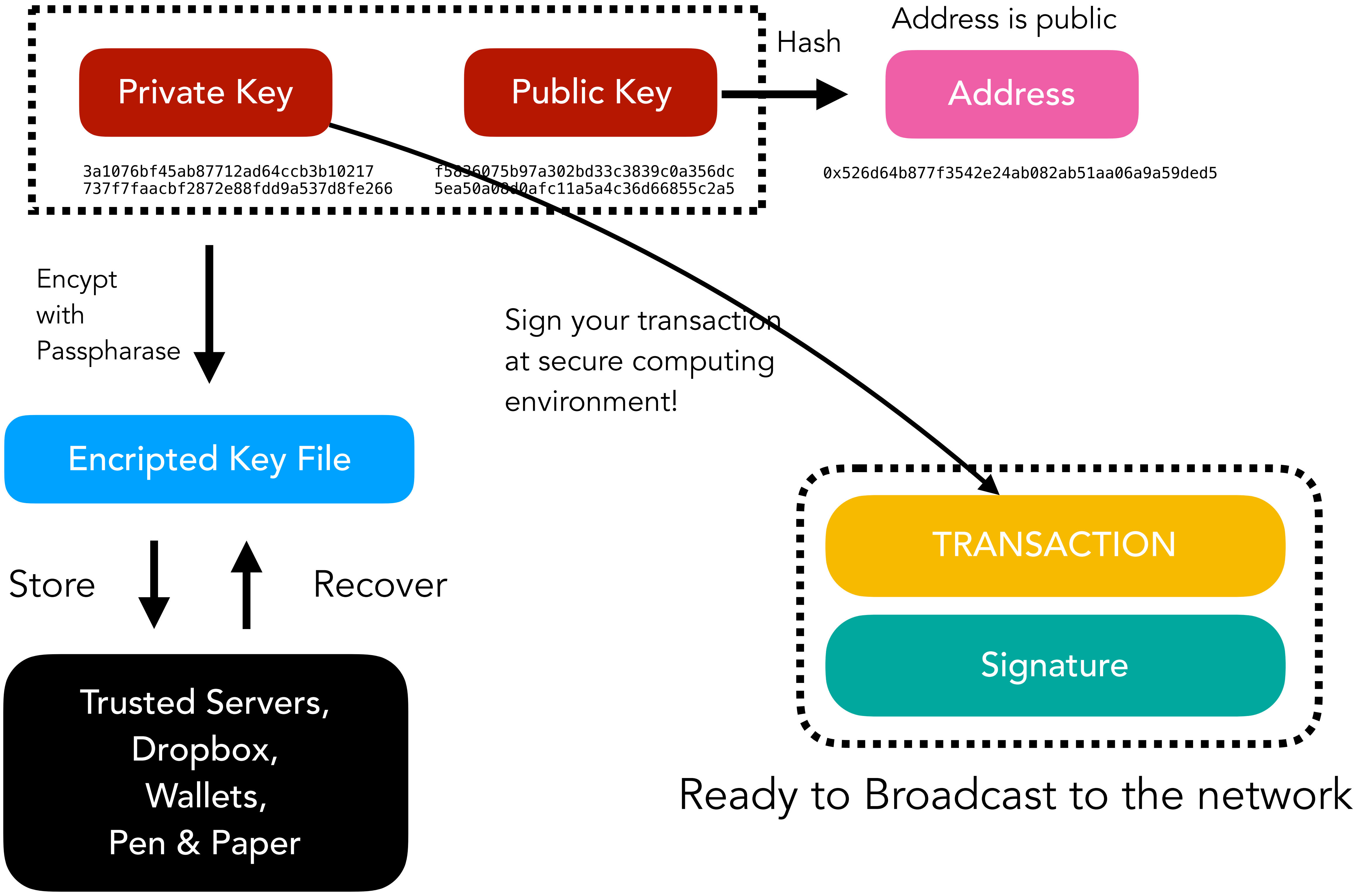


Radix Trie



Consistency Verification
Data Verification
Data Syncronization
Efficient Updates (Radix)

Account & Keys



Private Key

3a1076bf45ab87712ad64ccb3b10217
737f7faacbf2872e88fdd9a537d8fe266

Public Key

f5836075b97a302bd33c3839c0a356dc
5ea50a08d0afc11a5a4c36d66855c2a5

Hash

Address is public

Address

0x526d64b877f3542e24ab082ab51aa06a9a59ded5

Encypt
with
Passpharase

Encripted Key File

Store

Recover

Trusted Servers,
Dropbox,
Wallets,
Pen & Paper

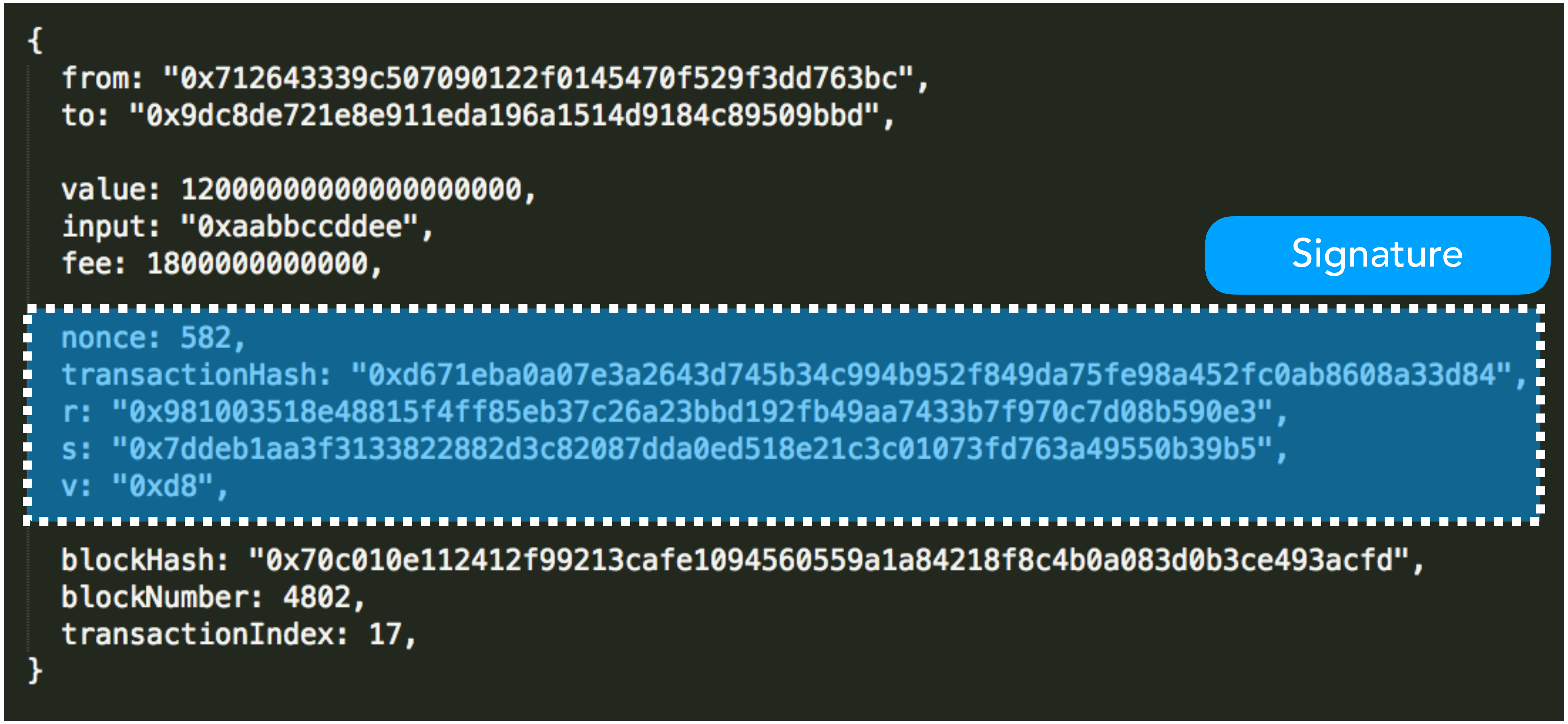
Sign your transaction
at secure computing
environment!

TRANSACTION

Signature

Ready to Broadcast to the network

Transaction

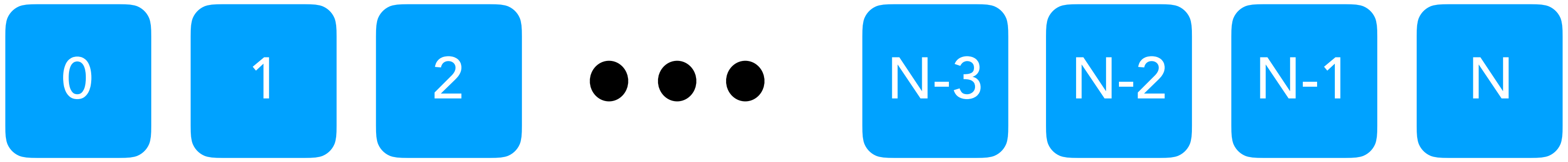


```
{
  from: "0x712643339c507090122f0145470f529f3dd763bc",
  to: "0x9dc8de721e8e911eda196a1514d9184c89509bbd",

  value: 12000000000000000000,
  input: "0xaabbccddee",
  fee: 1800000000000,

  nonce: 582,
  transactionHash: "0xd671eba0a07e3a2643d745b34c994b952f849da75fe98a452fc0ab8608a33d84",
  r: "0x981003518e48815f4ff85eb37c26a23bbd192fb49aa7433b7f970c7d08b590e3",
  s: "0x7ddeb1aa3f3133822882d3c82087dda0ed518e21c3c01073fd763a49550b39b5",
  v: "0xd8",

  blockHash: "0x70c010e112412f99213cafe1094560559a1a84218f8c4b0a083d0b3ce493acfd",
  blockNumber: 4802,
  transactionIndex: 17,
}
```

Signature

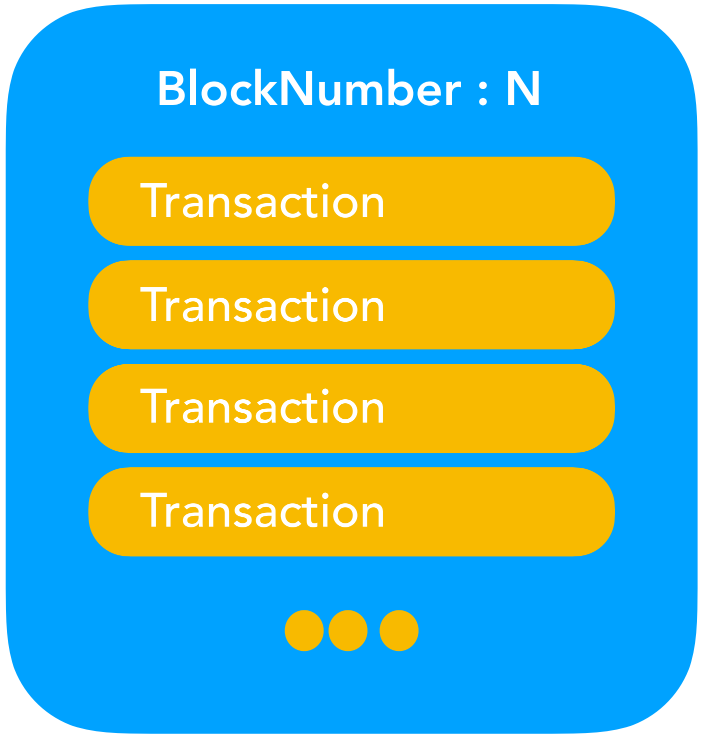
- Altering data (State Transition) in blockchain requires Transactions
- An account must have a right to alter a piece of data
- An account proves the right with her signature
- One can not clain that txn did not happen after the fact
- Txns can not be modified
- Non valid txns are ignored
- Txns can be created and signed offline!

State Transition



Latest Block
Latest State

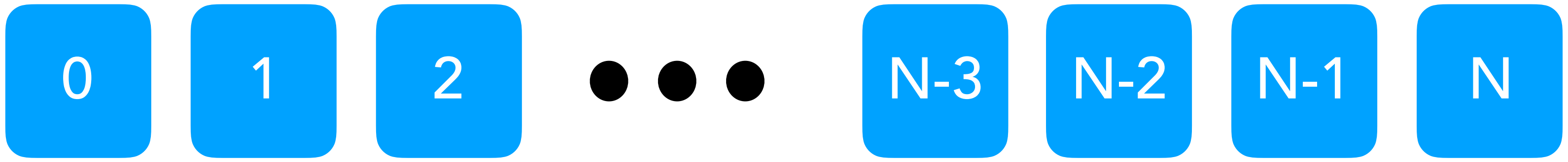
One or more txns are aggregated into a block.



```
Transition(State[N],(T1,T2..Tn)) =

Transition(State[N],Block) =

State[N+1]
```

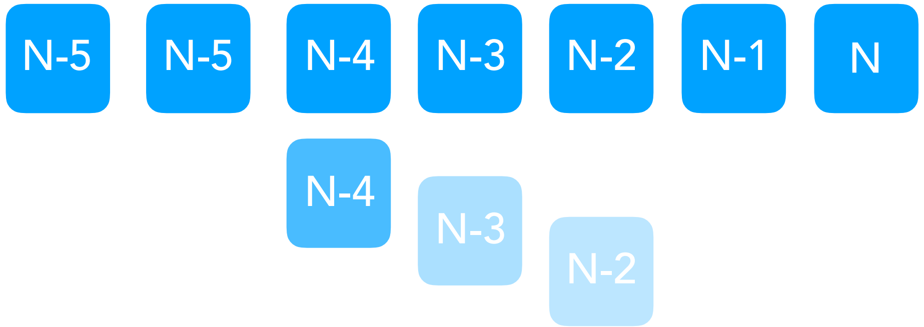
Proof of Work

```
for; nonce++
hash((T1,T2..TN),nonce) =? validAnswer
```



Latest Block
Latest State

- Miner provides valid Proof of Work solution
- Time period between blocks.
- Any number of peers can compete to generate a valid block
- Miners are rewarded: internal currency, fees
- All peers validate blocks before linking to previous valid block.
    - is proof of work valid?
    - is transaction processing done right?
- Network converges on same longest chain
- All peers have same copy of blockchain database
- Block size or execution steps are limited.



Ethereum Virtual Machine

EVM

Deterministic State Machine
Has an instruction set
Transaction cost
Execution (processsing & storage) cost

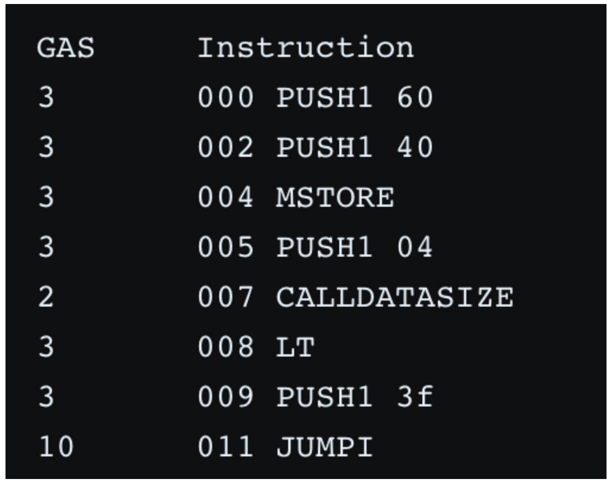
An Object or Contract

Compiled to assembly code
Deployed on Ethereum Network
Runs on EVM
Invoked by external actor
Also has an account
Cannot create txn

An Object or contract in Solidity

```
contract Asset{

     address owner;
     function Asset(){
          owner = msg.sender;
     }

     function transfer(address recipient)
{
          require(msg.sender == owner)
          owner = recipient
     }
}
```

Compiled code



```
GAS     Instruction
3       000 PUSH1 60
3       002 PUSH1 40
3       004 MSTORE
3       005 PUSH1 04
2       007 CALLDATASIZE
3       008 LT
3       009 PUSH1 3f
10      011 JUMPI
```