BLOCKCHAIN TECHNOLOGY

A peek into Ethereum

Ledger

A ledger is a system of records for a business that records asset transfer between participants.

Date	Visits
5/1/2018	2
5/2/2018	3
5/3/2018	5
5/4/2018	1
5/5/2018	2
5/6/2018	1
5/7/2018	3
5/8/2018	5
5/9/2018	1
5/10/2018	1
5/11/2018	1
5/12/2018	2

Date	Visits
5/1/2018	2
5/2/2018	3
5/3/2018	5
5/4/2018	1
5/5/2018	2
5/6/2018	1
5/7/2018	3
5/8/2018	5
5/9/2018	1
5/10/2018	1
5/11/2018	1
5/12/2018	2

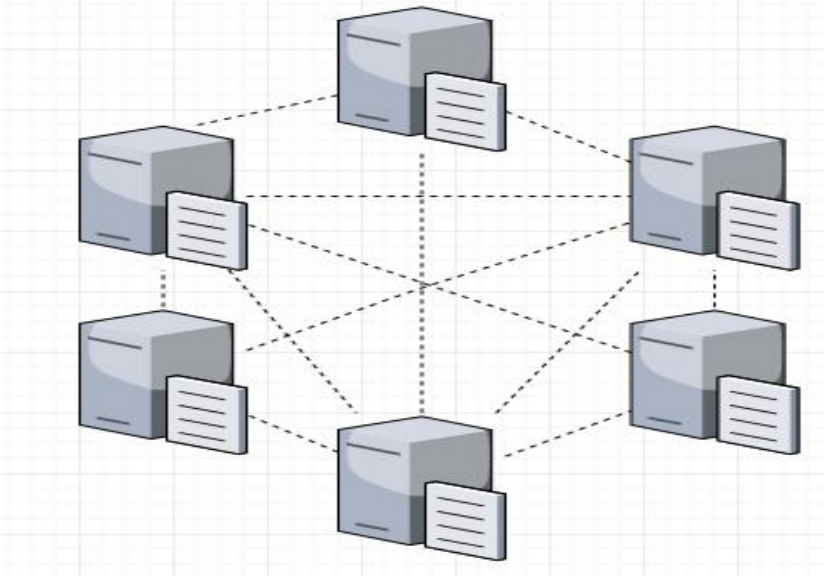
1	Date	Visits	Cumulative freq
2	5/1/2018	2	2
3	5/2/2018	3	5
4	5/3/2018	5	10
5	5/4/2018	1	11
6	5/5/2018	2	13
7	5/6/2018	1	14
8	5/7/2018	3	17
9	5/8/2018	5	22
10	5/9/2018	1	23
11	5/10/2018	1	24
12	5/11/2018	1	25
13	5/12/2018	2	27

Immutability through hashing

Sr.No	Date	Visits	Cumulative Frequency	Hash
1	5/1/2018	2	2	4247bf...
2	5/2/2018	3	5	f9d41c...
3	5/3/2018	5	10	0f2da6...
4	5/4/2018	1	11	8ef43c...
5	5/5/2018	2	13	562453...
6	5/6/2018	1	14	1da9ca...
7	5/7/2018	3	17	5bc31b...
8	5/8/2018	5	22	39760a...
9	5/9/2018	1	23	67e322...
10	5/10/2018	1	24	65082e...
11	5/11/2018	1	25	1ba8d3...
12	5/12/2018	2	27	6hsa21...

Decentralization

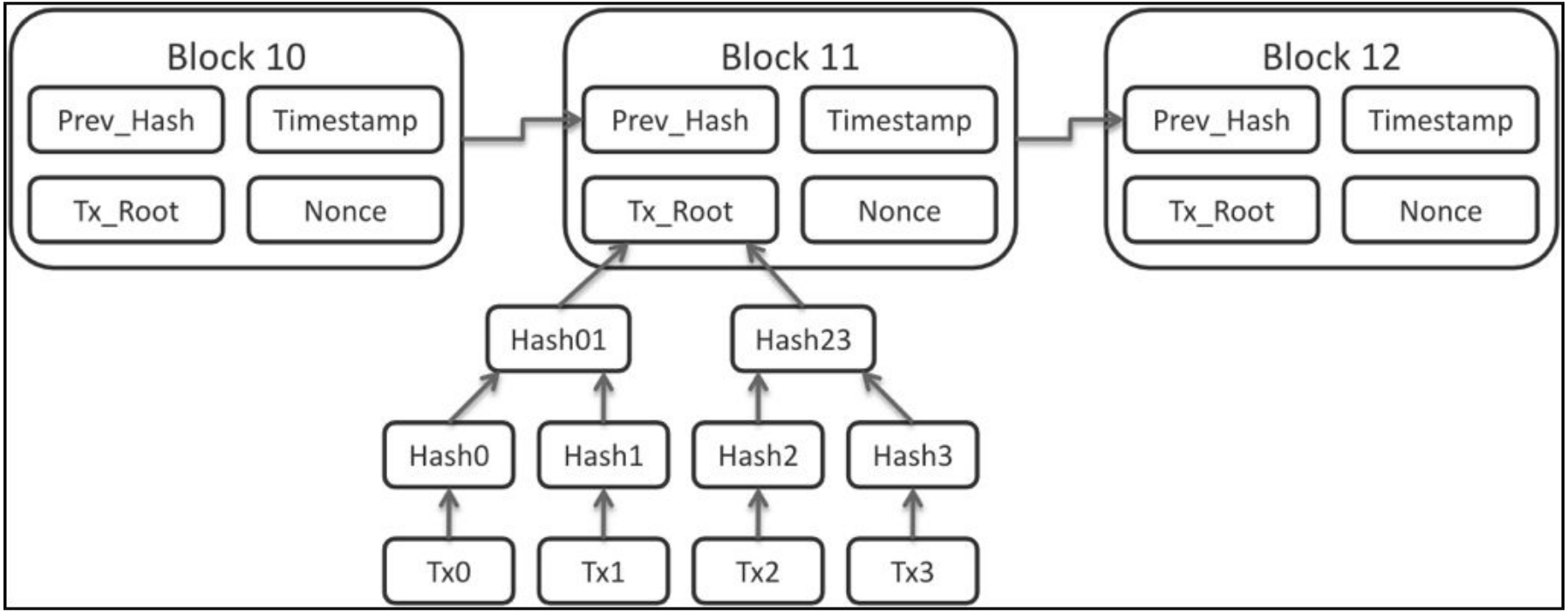
Sr.No	Date	Visits	Cumulative Frequency	Hash
1	5/1/2018	2	2	4247bf...
2	5/2/2018	3	5	f9d41c...
3	5/3/2018	5	10	0f2da6...
4	5/4/2018	1	11	8ef43c...
5	5/5/2018	2	13	562453...
6	5/6/2018	1	14	1da9ca...
7	5/7/2018	3	17	5bc31b...
8	5/8/2018	5	22	39760a...
9	5/9/2018	1	23	67e322...
10	5/10/2018	1	24	65082e...
11	5/11/2018	1	25	1ba8d3...
12	5/12/2018	2	27	6hsa21...



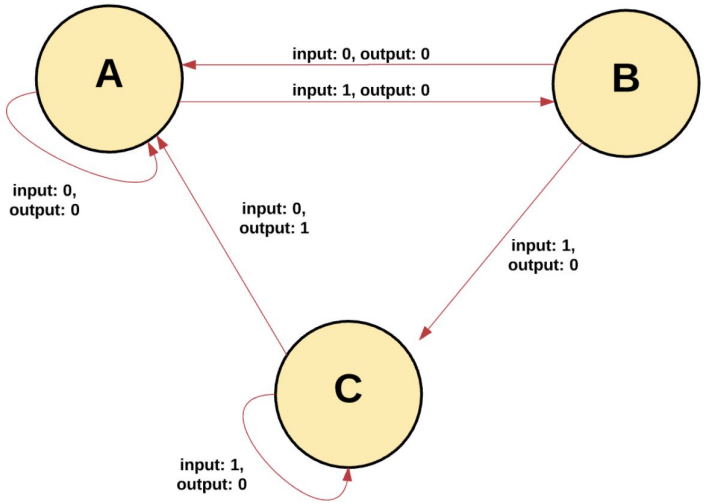
A chain of blocks

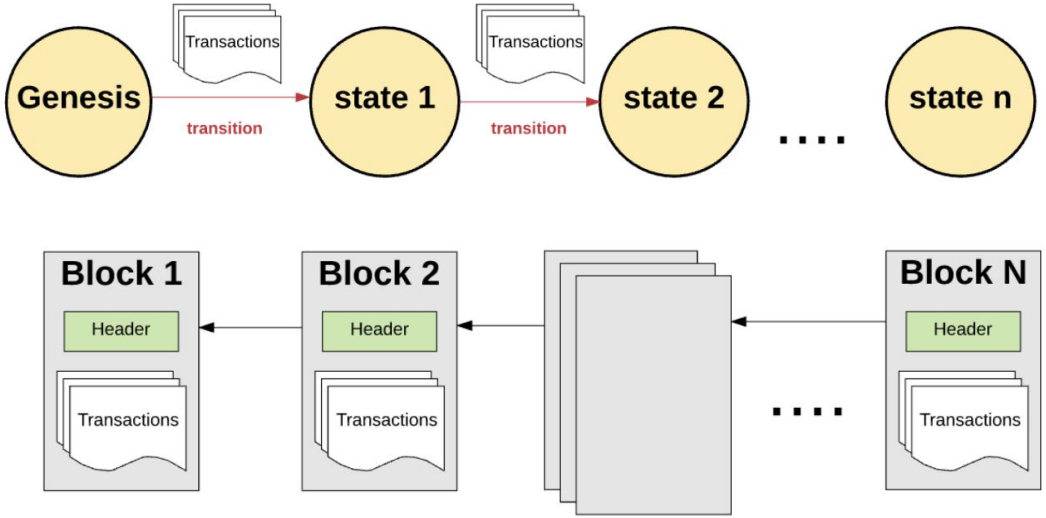
Sr.No	Date	Visits	Cumulative Frequency	Hash
Block No. 16				
1	5/1/2018	2	2	4247bf...
2	5/2/2018	3	5	f9d41c...
3	5/3/2018	5	10	0f2da6...
4	5/4/2018	1	11	8ef43c...
Block Hash of 16 = Hash(Block Hash of 15, Merkle Hash of 16)				
Block No. 17				
5	5/5/2018	2	13	562453...
6	5/6/2018	1	14	1da9ca...
7	5/7/2018	3	17	5bc31b...
8	5/8/2018	5	22	39760a...
Block Hash of 17 = Hash(Block Hash of 16, Merkle Hash of 17)				
Block No. 18				
9	5/9/2018	1	23	67e322...
10	5/10/2018	1	24	65082e...
11	5/11/2018	1	25	1ba8d3...
12	5/12/2018	2	27	6hsa21...
Block Hash of 18 = Hash(Block Hash of 17, Merkle Hash of 18)				

The Blocks (merkle tree)

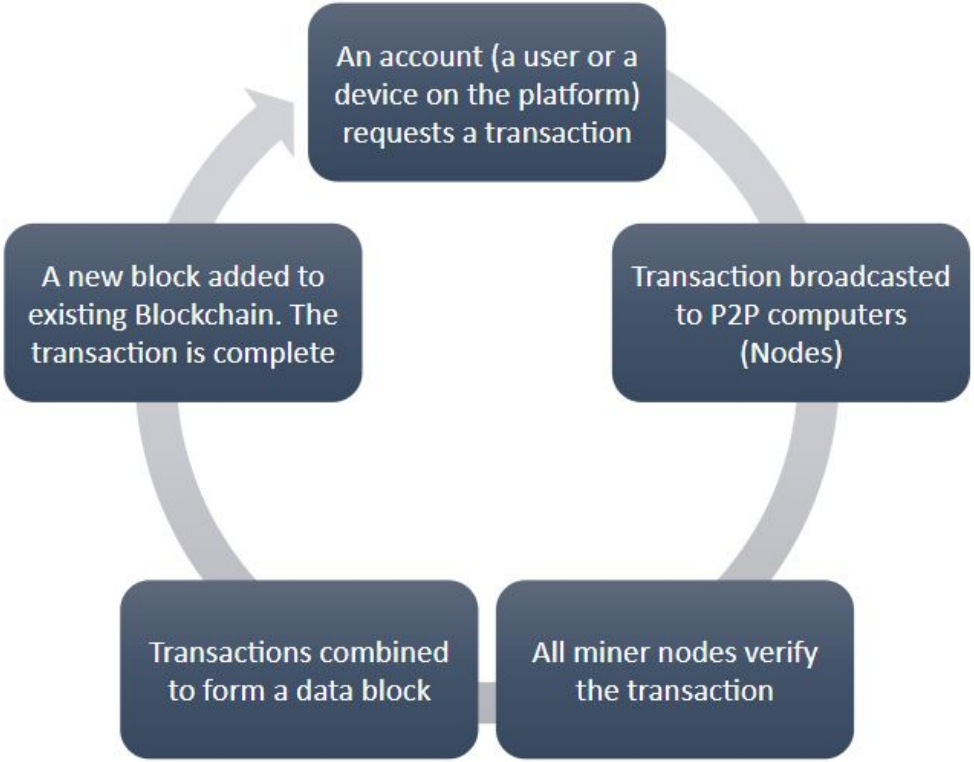


Ethereum: Transaction-Based State Machine





Transaction lifecycle



Debugging

Q. A simple bid function that takes the value(amount) from message and updates the highest bidder while returning the previous amount back to the previous highest bidder.

Possible sol.

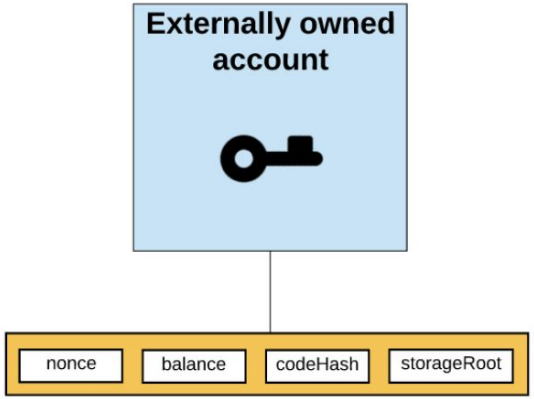
```
pragma solidity ^0.4.15;
contract Auction {
 address currentFrontrunner;
 uint currentBid;
 //Takes in bid, refunding the frontrunner if they are outbid
 function bid() payable {
   require(msg.value > currentBid);
   //If the refund fails, the entire transaction reverts.
   //Therefore a frontrunner who always fails will win
   if (currentFrontrunner != 0) {
     //E.g. if recipients fallback function is just revert()
     require(currentFrontrunner.transfer(currentBid));
   }
   currentFrontrunner = msg.sender;
   currentBid         = msg.value;
 }
}
```

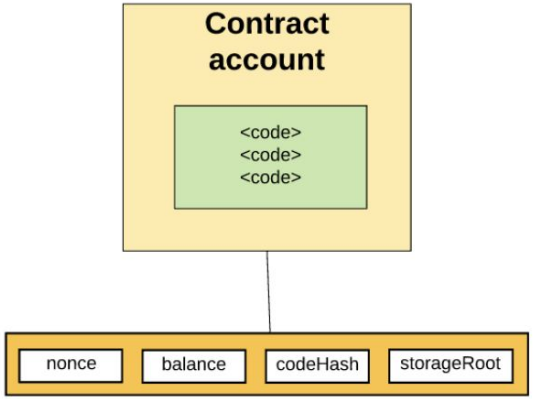
Refactored sol.

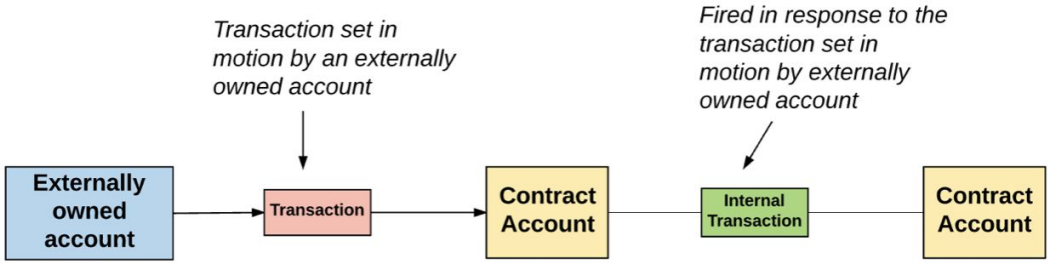
```
contract Auction {
 address currentFrontrunner;
 uint    currentBid;
 mapping(address => uint) refunds; //Store refunds in mapping to avoid DoS
 //Avoids "pushing" balance to users favoring "pull" architecture, be careful with loops too
 function bid() payable external {
   require(msg.value > currentBid);
   if (currentFrontrunner != 0) { refunds[currentFrontrunner] += currentBid; }
   currentFrontrunner = msg.sender;
   currentBid         = msg.value;
 }
 function withdraw() external { //Allows users to get their refund from auction
   //Do all state manipulation before external call to
   //avoid reentrancy attack
   uint refund = refunds[msg.sender];
   refunds[msg.sender] = 0;
   msg.sender.send(refund);
 }
}
```

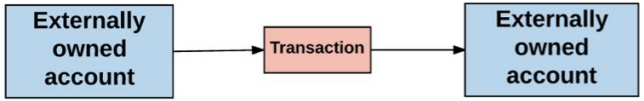
Terminologies

Account

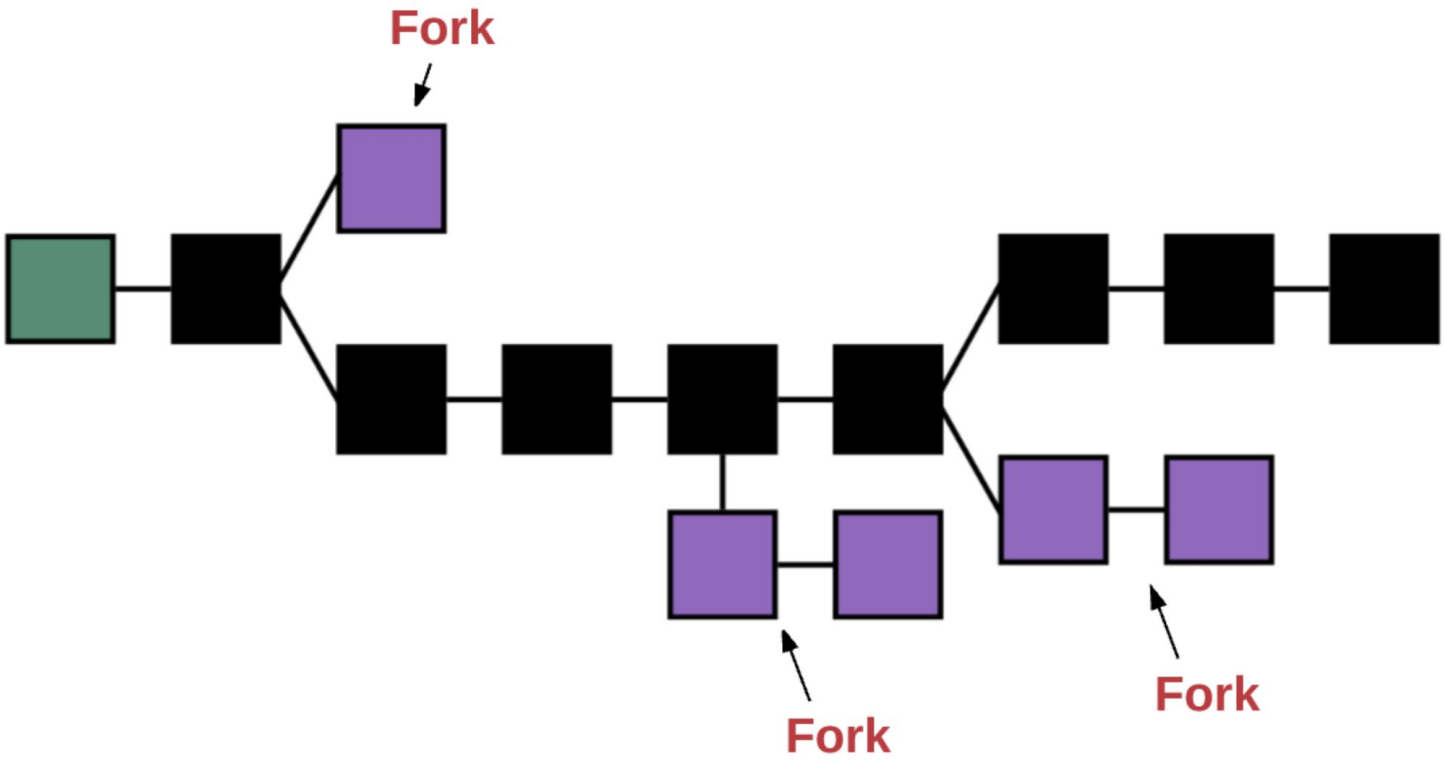

Externally owned account
nonce
balance
codeHash
storageRoot


Contract account
<code>
<code>
<code>
nonce
balance
codeHash
storageRoot


Transaction set in motion by an externally owned account
Fired in response to the transaction set in motion by externally owned account
Externally owned account
Transaction
Contract Account
Internal Transaction
Contract Account


Externally owned account
Transaction
Externally owned account

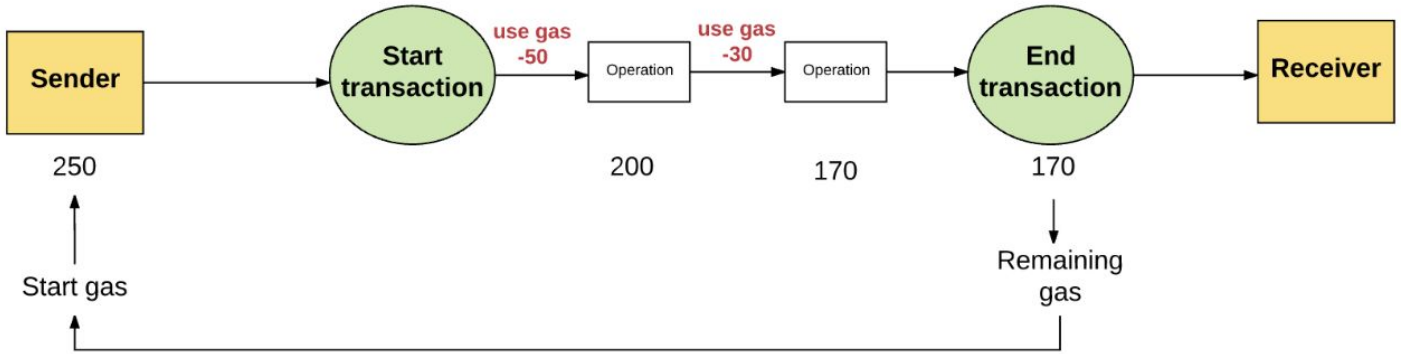
Fork



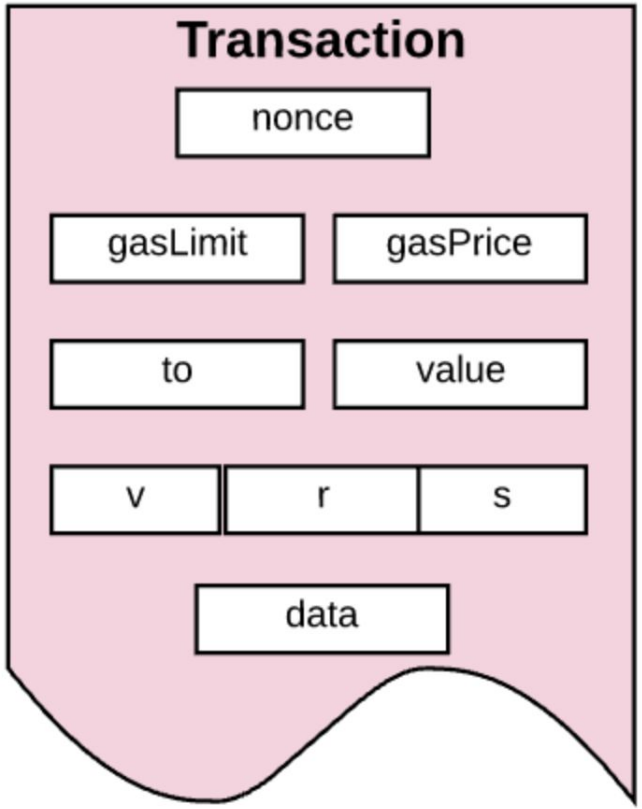
GHOST Protocol = "Greedy Heaviest Observed Subtree"

Gas

Gas Limit	x	Gas Price	=	Max transaction fee
50,000		**20 gwei**		**0.001 Ether**



Transaction


Transaction
nonce
gasLimit
gasPrice
to
value
v
r
s
data

WALLET

- The interface / client / wrapper / holder that you use to manage your account(s).
- Example: MyEtherWallet.com, MEWconnect, MetaMask, a hardware wallet (i.e. Ledger, Trezor, BitBox, Secalot, etc..), a Multisig Wallet Contract.

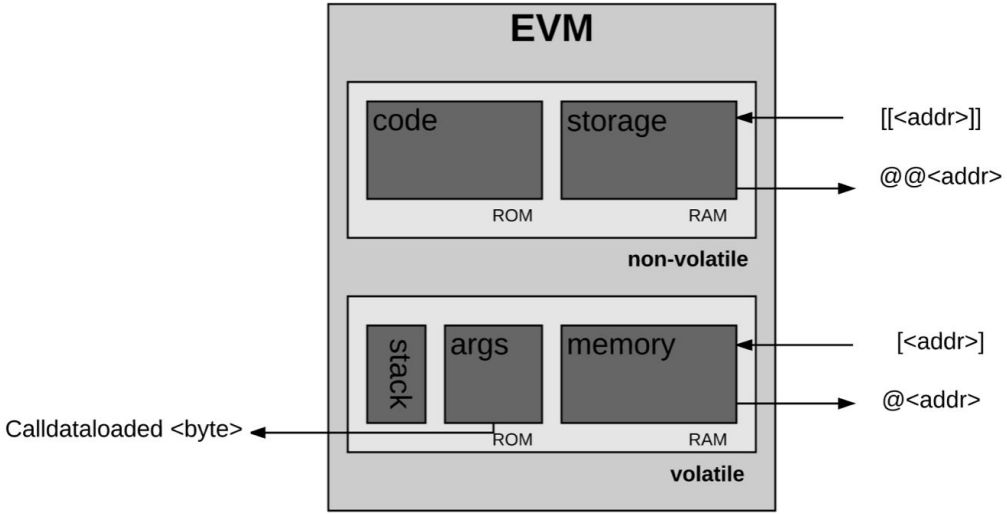
Explorer

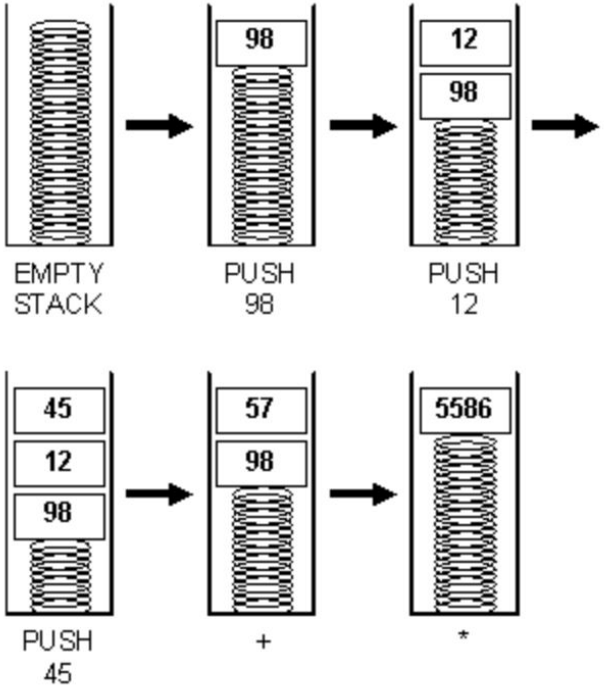
Bitcoin v/s Ethereum

EVM

How EVM works?

- LIFO manner
- Turing complete machine but is intrinsically bound by gas.





Before executing a particular computation, the processor makes sure that the following information is available and valid:

- System state
- Remaining gas for computation
- Address of the account that owns the code that is executing
- Address of the sender of the transaction that originated this execution
- Address of the account that caused the code to execute (could be different from the original sender)
- Gas price of the transaction that originated this execution
- Input data for this execution
- Value (in Wei) passed to this account as part of the current execution
- Machine code to be executed
- Block header of the current block
- Depth of the present message call or contract creation stack

At the start of execution, memory and stack are empty and the program counter is zero.

How does Ethereum query data?

ETHEREUM

Smart Contracts (Remix IDE)

Ganache-cli (Previously testrpc)

Framework for development

Wallets (Metamask)

Geth