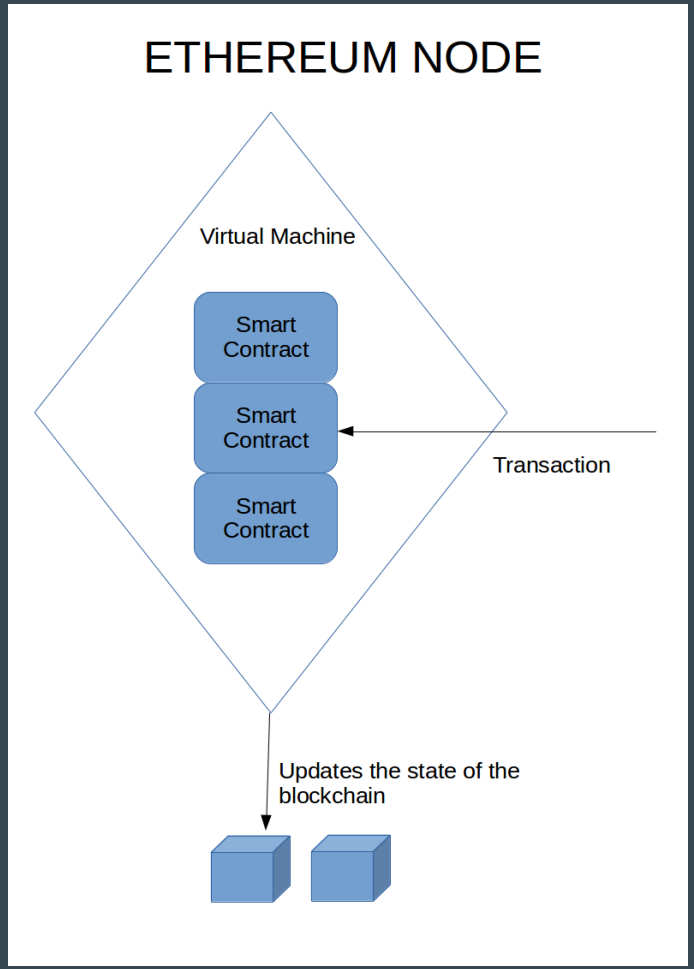
# Smart Contracts

- Contracts lives on the Ethereum blockchain
- They have their own Ethereum address and balance
- They can send and receive transactions
- They are activated when they receive a transaction, and can be deactivated
- The Ethereum Virtual Machine runs a turing complete language
- They have a fee per CPU step, with extra for storage
- The user can run the application on their local block chain

# Ethereum Node



# Ethereum Programming Languages

Smart contracts can be written in

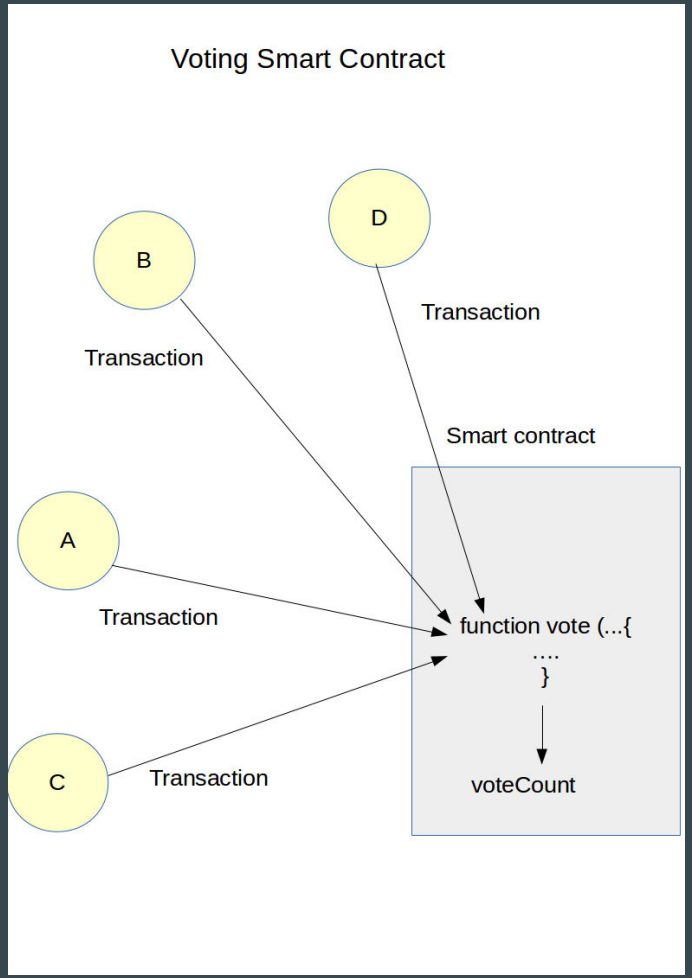
Solidity (a **JavaScript-like** language)
Serpent (a **Python-like** language),
Mutan (C-like)
LLL (**Lisp**-like).

They are compiled into bytecode before being deployed to the **blockchain**.

# An Example Smart Contract - A voting application

The state of the contract (voteCount) is maintained on the blockchain along with the smart contract

After a certain time the smart contract will end the election and publish the results



```
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;

    // Create a new ballot
    function Ballot(uint8 _numProposals) {
        chairperson = msg.sender;
        voters[chairperson].weight = 1;
        proposals.length = _numProposals;
    }

    // Give a single vote
    function vote(uint8 proposal) {
        Voter sender = voters[msg.sender];
        if (sender.voted || proposal >= proposals.length)
return;
        sender.voted = true;
        sender.vote = proposal;
        proposals[proposal].voteCount += sender.weight;
    }

    function winningProposal() constant returns (uint8
winningProposal) {
        uint256 winningVoteCount = 0;
        for (uint8 proposal = 0; proposal <
proposals.length; proposal++)
            if (proposals[proposal].voteCount >
winningVoteCount) {
                winningVoteCount =
proposals[proposal].voteCount;
                winningProposal = proposal;
            }
    }

}
```

ethereum

# Ether buys GAS to fuel the EVM

Every opcode instruction executed by the EVM uses up Gas.