Crowd Sale - Initial Coin Offerings

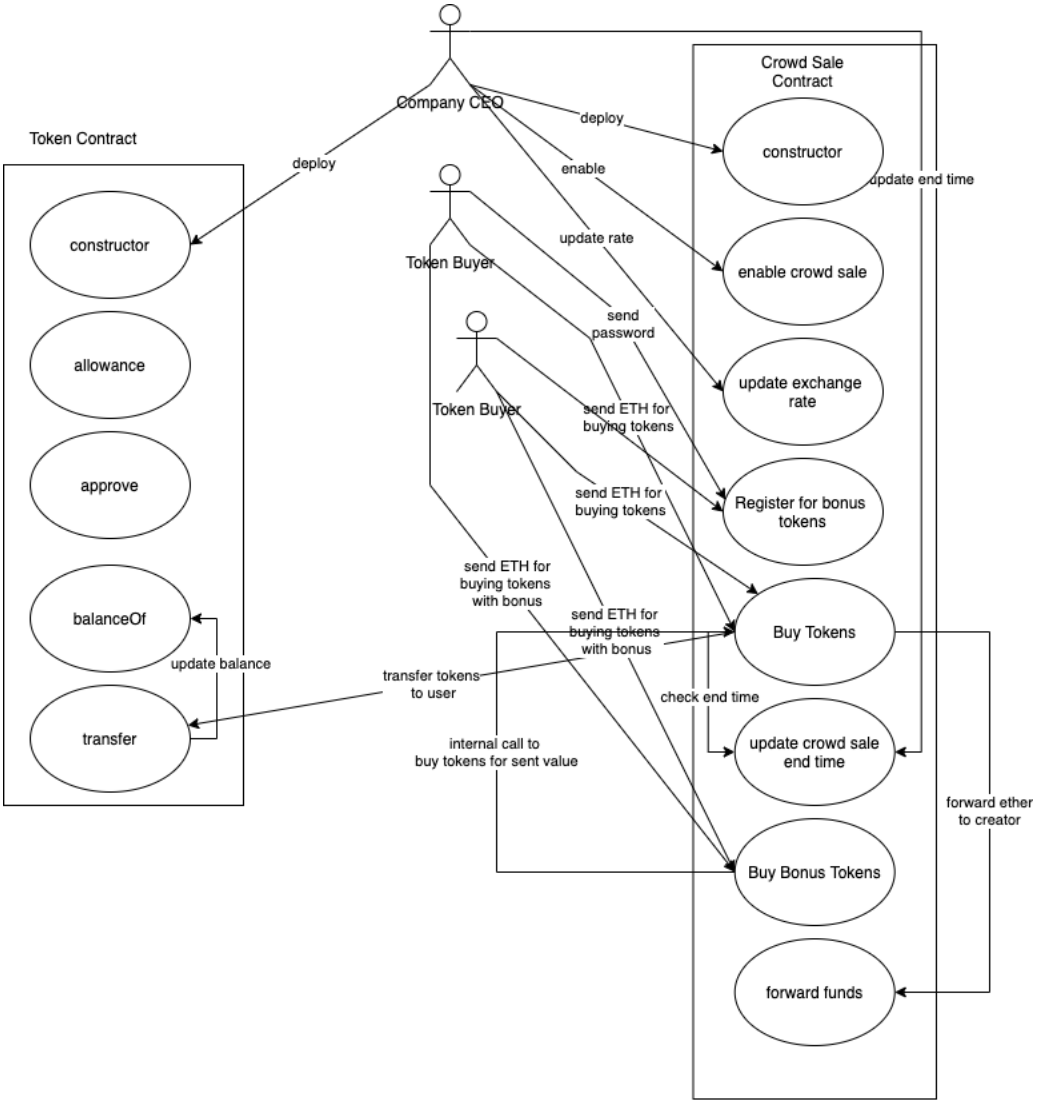
Abstract

In the world of blockchain which has endless possibilities, startups can use Blockchain to raise funds(Ether) to fund their development. The company will try to sell the token(cryptocurrency) it created in exchange for ethers. This way the company will raise funds. This project aims to develop a token(For Eg: BitCoin) and also develop a crowd sale contract which the people can call to buy tokens from the company. There are more additions to this where the owner/admin can cap the number of coins to be sold. Also, this crowd sale will be conducted in a timely manner where it will have start and end times of crowd sale and if the desired amount of ethers are not raised the once the crowd sale is ended, the tokens are taken back from the people in exchange for the ethers they spent(safe sale). If the desired amount of ethers are raised then we can transfer the ether raised to the wallet of the company.

DAPP Name: Crowd Sale - ICO
Clients: Companies that look to raise funds.

Name: Deepakanuraag Sathyanarayanan. Person Number: 50321237

2. Use Case Diagram



In this use case diagram the roles of actors are as follows,
Company CEO has following actiongs

1) Deploy Token Contract with total supply
2) Deploy Crowd Sale Contract with Crowd Sale Tokens with address of token contract
3) Update exchange rate for ex: 1 ETH = 500 tokens
4) Update crowd sale end time
5) Enable crowd sale

Token Buyers has following actions

1) Able to buy tokens at the listed exchange rate
2) Able to get bonus token while buying if you have registered for bonus tokens by sending 0.2 ETH
3) Able to transfer the tokens to any other address like any other cryptocurrency For Eg: BITCOIN

This basically is the implementation of crowd sale, where the company CEO will receive ETH by selling its custom built ERC20 tokens.
Approve comes from IERC20 interface and approval from an address is only required if an address tries to transfer tokens not its own, then it needs approval from that address however we will not require since crowdSaleContract will have predetermined number tokens already sent to it.

3. Contract Diagram

Token Contract

//data
mapping (address => uint256) balances
mapping (address=> mapping(address=> uint256)) allowed
uint256 totalSupply;
mapping (address => uint256) internal _balances;

// modifiers require (balanceOf(sender)>amount) // should be greater
//functions function transfer(address to,uint256 value) function approve(address spender,uint256 value) function allowance(address owner,address spender) function balanceOf(address sender)

Continued on next page

Crowd Sale Contract
//data bool isCrowdSaleEnabled; address creator; uint exchange_rate; bool isConfigSet;

// modifiers
modifier onlyCreator();
require(hashedPasswordForUser == passwordMap[msg.sender]);
require (_endTime >= now,'end time should be greater than now')
require (_exchange_rate >0 ,'exchange rate should be higher than zero)
require (!isConfigSet);
require (isCrowdSaleEnabled); // check if crowd sale is enabled
require (msg.value!=0) // check if ethers are sent for buying tokens

// functions function buyTokens() function enableCrowdSale(bool) function updateExchangeRate(uint256) function updateCrowdSaleEndTime(uint256) function forwardFunds() function getRate() function registerForBonusTokens() function buyTokensWithBonus()

4. State Diagram