K B A

Building an Auction Dapp

Auction Description

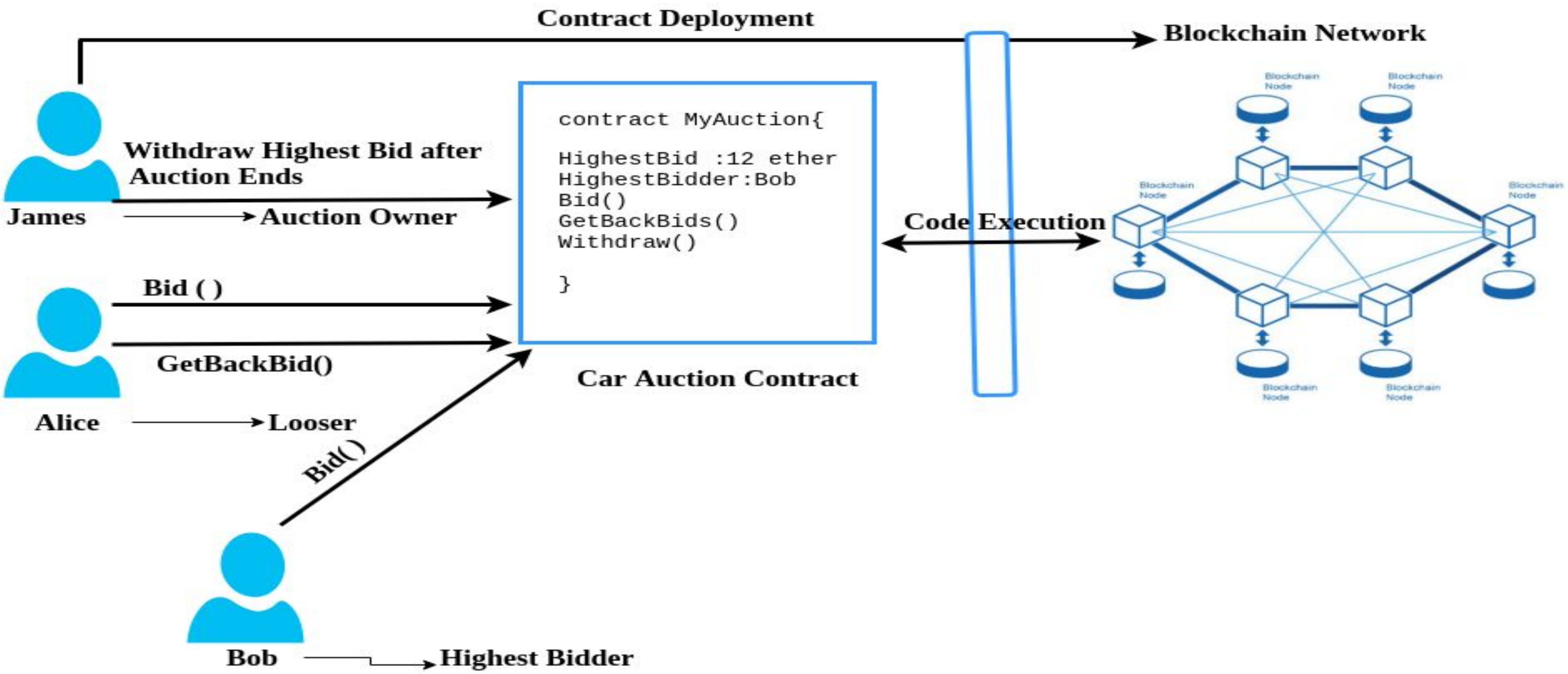
A vehicle’s owner deploys the contract to the blockchain and becomes the auction owner. The auction is open immediately after the contract deployment, and, once the bidding period is over, the highest bidder wins the auction, and the other participants get back their bids.

Problem Statement

Implement a smart contract for Auction.

- The contract should have exactly one owner.
- Clients can bid vehicle for time slots provided by owner
- once the bidding period is over, the highest bidder wins the auction, and the other participants get back their bids.
- once the bidding period is over, the highest bidder wins the auction, and contract owner gets highest bid

Auction



```
contract MyAuction{

HighestBid :12 ether
HighestBidder:Bob
Bid()
GetBackBids()
Withdraw()

}
```

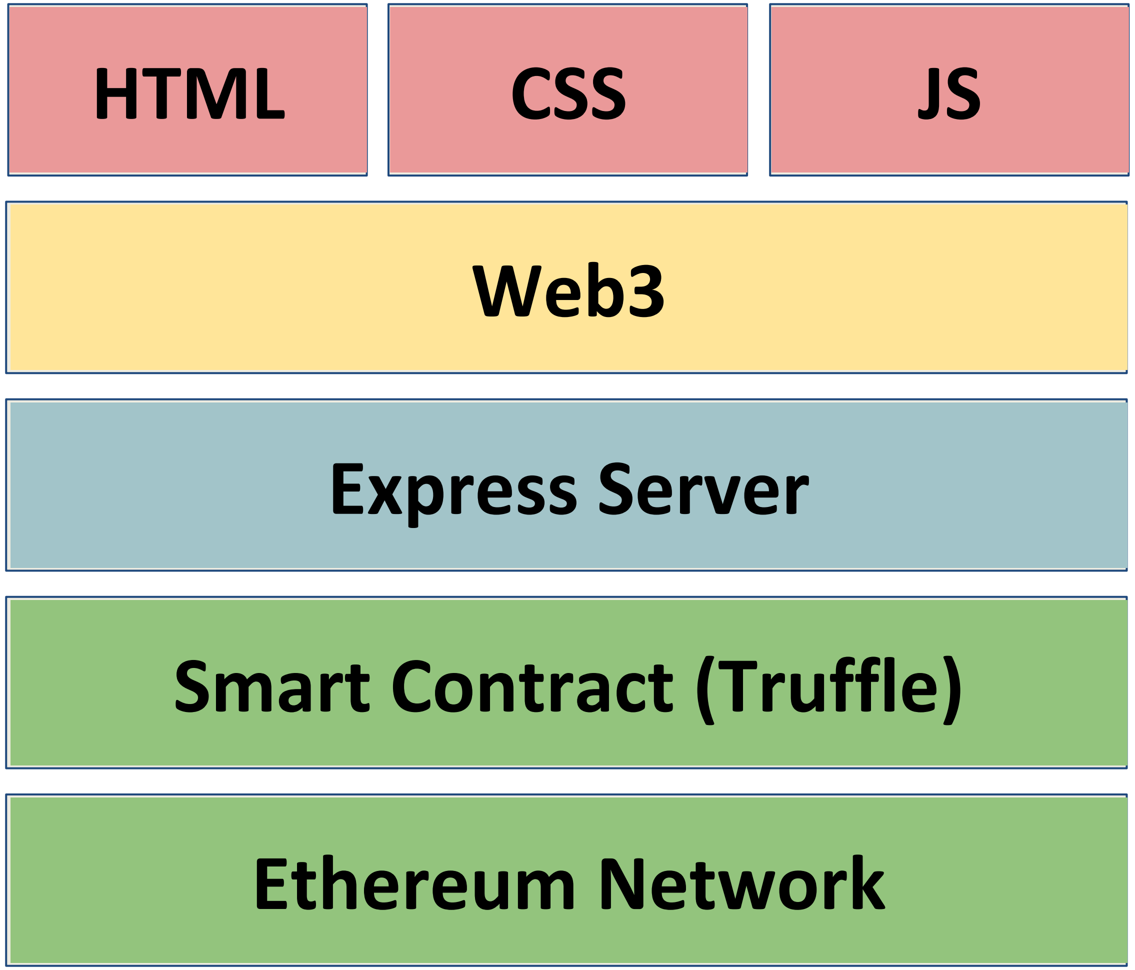
Contract Skeleton

```
pragma solidity ^0.5.0;
contract Auction{
   //Declare All State Variables here

   //Define a constructor for your contract
   constructor( public {

}

//Define a structure for Vehicle Details
   struct car{
       string Brand;
       string Rnumber;
       address owner;
   }
   car public Mycar;



   //Mapping that accepts the bidder's address as the key, and with the value type being the corresponding bid

   mapping(address => uint) public bids;
   //Checks whether the bid is can be done
   modifier bid_conditions(){


   }
```

Contract Skeleton

```
 //makes the contract ownable
   modifier only_owner(){

   }
//Define Bidding function
   function bid() public payable bid_conditions returns (bool){


   }
 //Withdraw function for loosers
   function getAmount() public returns (bool){
             }

  //Withdraw Bid amount to owner address

   function withdraw() public only_owner returns (bool){



   }
```

Dapp Architecture



HTML

CSS

JS

Web3

Express Server

Smart Contract (Truffle)

Ethereum Network

THANK YOU