Auction

Core Idea

The core idea is to distribute the auction mechanism in phases with each phase being implemented using a smart contract. We define a few states in the auctioning process to ensure fairness in the process: OPEN, CLOSED, FINISHED. It prevents the owner and users from executing functions that cannot be executed in that particular state.

Highest Bid phase

We deal with the case of an open auction where a product is sold with several prospective bidders who wish to buy the product by bidding the highest amount. Initially, there is no bid, so a bidder can submit a bid. The current bid is always considered the highest bid.

If another bidder wants to bid over the current bid, he must put a bid higher than the current bid. This is an auction where the product is sold to the highest bidder. There is no point in bidding on a value smaller than the current bid as you won't be able to buy that product anyway.

In the situation of a higher bid than the current bid, the current bid stores the new higher bid and this is broadcast to all the bidders. Since it is an open auction, we need to inform all the bidders who have put the highest bid and how much is that. This helps the prospective bidders make an informed choice whether to bid higher or not.

There will be no option of revoking a submitted bid. In the case of a bidder who has submitted a bid and won the auction does not want to abide by it by withdrawing, such an option will not be entertained. A submitted bid cannot be revoked.

The current highest bid is always stored and any other bid lower than that is discarded. An individual can submit bids as they please as long as they are bigger than the highest bid. The blockchain stores the smart contract takes the highest bid and registers it.

The bidder does not need to pay any fees to submit a bid, he/she will anyhow pay for the product in the case of winning the auction.

The bidding for the highest bid is completely open and no secret bids will be entertained. When a new highest bid is made it must be broadcasted to all the prospective bidders. Secret bids can hamper the fairness of the auctioning process. Open means that secrecy will not be entertained.

Thus, in the case of any new higher bid, a broadcast is mandatory. This will ensure that the open auctioning process is fair, transparent, and impartial.

In this phase the auction is OPEN.

Winner announcement

In this phase the auction is CLOSED.

The winner of the auction is the bidder with the highest bid. The blockchain smart contract always stores the current highest bid with the address of the bidder. As the auctioning process is closed, the current highest bid is the highest bid of the auction.

As the contract already stores the highest bid of the auction, no further computation is required. The address of the highest bidder is already computed. He is the winner of the auction.

The payment for the product which is the highest bidding price is given to the owner in exchange for the product which is being bought. This will essentially complete the auctioning process.

It terminates with the FINISHED state as all processes in the auction are complete.

We also implement a change state mechanism to prevent changes that are prohibited which ensures that no action is taken to spoil the auction. The only valid state change is from OPEN to CLOSED to FINISHED. All other state changes are invalid and need to be disregarded.

Smart Contract Implementation

```
// Create a smart contract
contract Auction{

  // Bid value to be replaced with highest bids
  uint256 current_bid = 0;

  // To store the address of the current bidder
  // As there is no bidder till now, an arbitrary value is stored
  address address_current_bidder = 0x0000000000000000000000000000000;

  // OPEN = 1, CLOSED = 2, FINISHED = 3
  // Only valid order is 1 -> 2 -> 3
  // Initially Auction is OPEN
  uint Auctionstate = 1;

  // Function for bidder to put the highest bid
  function bid(uint256 val, address bidder){

    // Requirements

    // Auction must be OPEN to submit bid
    require Auctionstate == 1;
```

```
        // Bid must be higher than the current highest bid
        // Initial bid will always be accepted as current_bid is set to ZERO at the beginning
        // Any bid will be more than ZERO, so it will be accepted
        require val > current_bid;

        // Set the current bid to the highest bid so far by the bidder
        current_bid = val;

        // Set the current address to the address of the highest bidder
        address_current_bidder = bidder;

        // Inform all the prospective bidders of the highest bid so far and its bidder
        // It will help them decide to put a higher bid or not
        emit (current_bid, address_current_bidder);
    }

    // Announce the auction winner
    function auction_winner(){

        // As the auction winner is being declared, we cannot consider more bids
        // Auction is CLOSED
        changestate(2);

        // Auction must be CLOSED to declare the winner
        require Auctionstate == 2;

        // Product sold to the highest bidder
        // Payment of the auction money to the owner
        payable(owner().transfer(current_bid));

        // Announce the winner of the auction to all the bidder
        // Auction winner is displayed
        return (current_bid, address_current_bidder);

        // As the auction winner has been declared and the payment has been made to the owner
        // Auction is FINISHED
        changestate(3);
    }

    // Valid change of the Auction state
    function changestate(uint state){

        // Either we can move 1 -> 2 or 2 -> 3
        require (Auctionstate == 1 && state == 2) ||
                (Auctionstate == 2 && state == 3);
        // Change the current Auction state
        Auctionstate = state;
    }
}
```