

Auction Contract

- Smart Contract for a Decentralized Auction, **an ebay alternative**;
- The Auction has: **an owner** (the person that sells a good or service), **a start and an end time**;
- The owner can cancel the auction if there is an emergency and can finalize the auction after its end time;
- People are sending ether when they are calling a function named `placeBid()`. We'll register in a mapping the sender address and the sent value;
- Users are incentivized to bid the maximum they're willing to pay, but they are not bound to that full amount, but rather to the previous highest bid plus the **increment**. **The contract will automatically bid up to a given amount**;
- The `highestBindingBid` is the selling price and the `highestBidder` the person who won the auction;
- After the Auction ends the owner gets the `highestBindingBid` and everybody else **withdraws** their own sent amount;

Auction Contract - placeBid() function

bids[0x123...] = 40

bids[0xabc...] = 70

bidIncrement = 10

highestBidder = 0xabc...

highestBindingBid = 50

0x123... sends 100

Bids[0x123...] = 40 + 100 = 140

highestBindingBid = min(55 + 10, 70) = 65