



Smart Contract Testing

Course Overview

Tools

Truffle

Hardhat

Mocha

contract()

describe()

Project

Ebay Smart Contract

Testing

Course Prerequisites

Blockchain

Solidity

JavaScript (ES6)

Web3.js

Smart Contract Testing

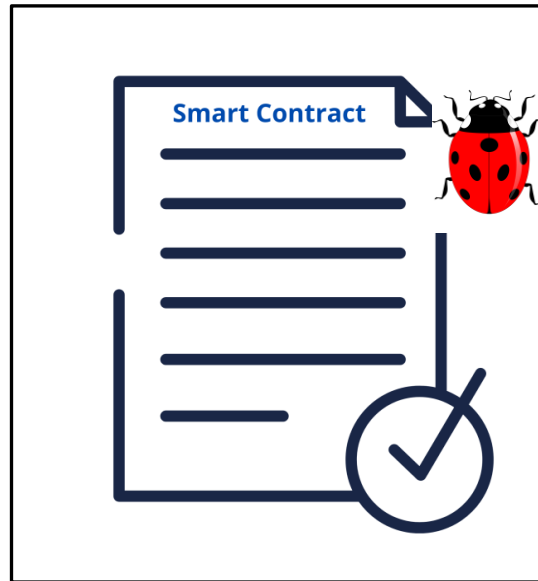
To point out defects

To increase reliability

To avoid extra cost

To gain client trust

Smart Contract Testing



Blockchain



Tools For Testing

Truffle

Hardhat

Tools For Testing

Truffle

Hardhat

Testing in Truffle

- Truffle comes with a testing framework by default to automate the testing of our contracts.
- Two ways to write tests in truffle :

JavaScript

Solidity

Testing in Truffle

- Truffle comes with a testing framework by default to automate the testing of our contracts.
- Two ways to write tests in truffle :

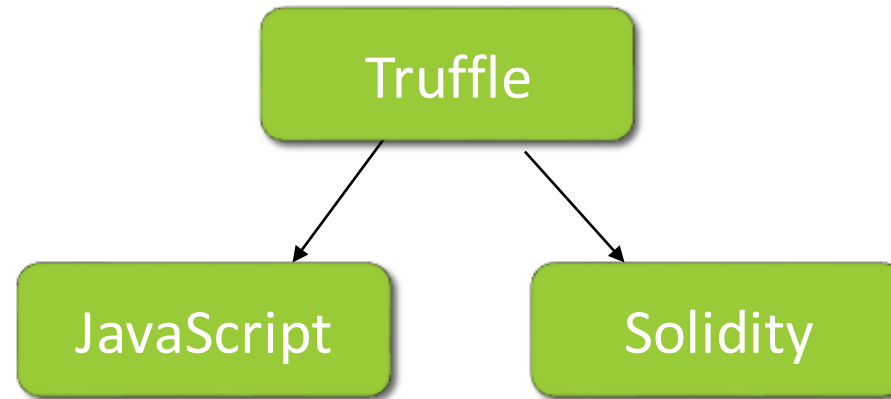
JavaScript

Solidity

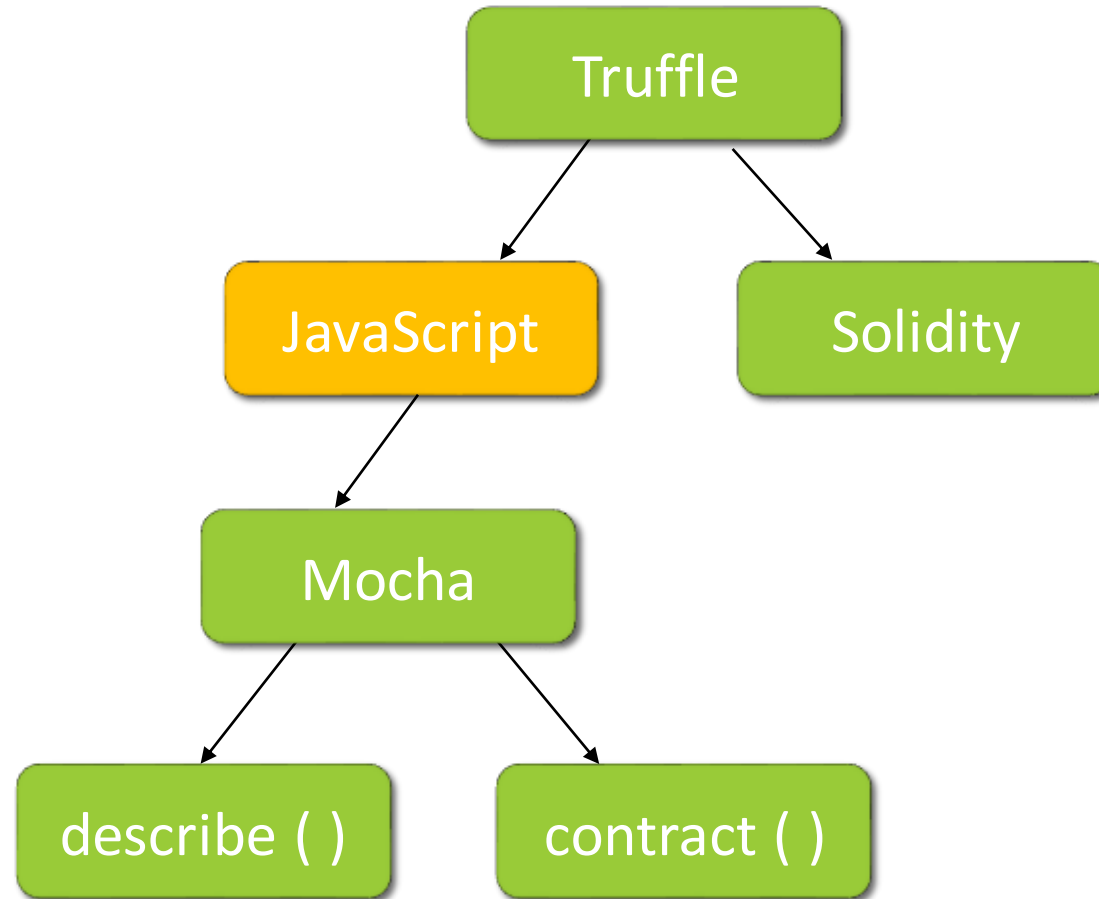
Test Using JavaScript

- Truffle's JavaScript testing framework is built on top of **mocha**.
- Mocha is a **JavaScript framework** to write tests.
- What makes truffle tests different from mocha is the **contract ()** function.
- What actually mocha uses is a **describe ()**.

Summaries



Summaries



Creating Instance Of A Contract

```
const helloTesting = await HelloTesting.deployed()
```

helloTesting



HelloTesting.sol

Describe()

Hardhat with ether.js

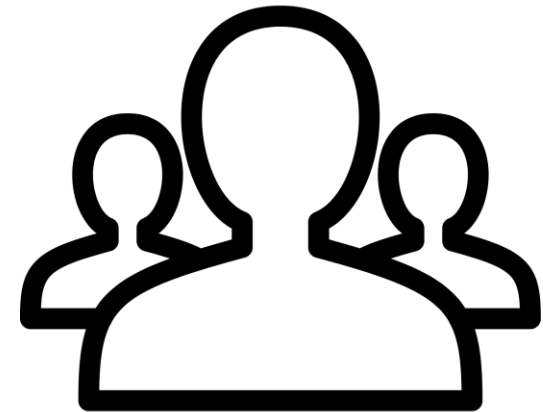
Hardhat vs Truffle Tool



Testing

BY CODE EATER

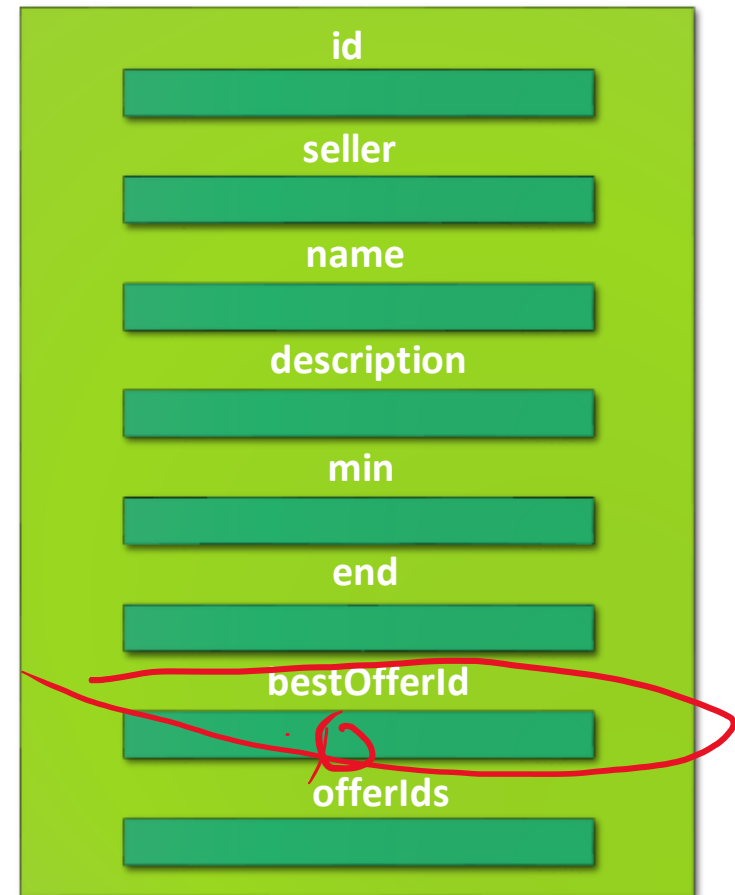
Algorithm



Auction Structure

auction · best offerId ~~50~~

```
struct Auction {  
    uint id;  
    address payable seller;  
    string name;  
    string description;  
    uint min;  
    uint end;  
    uint bestOfferId;  
    uint[] offerIds;  
}
```



Offer Structure

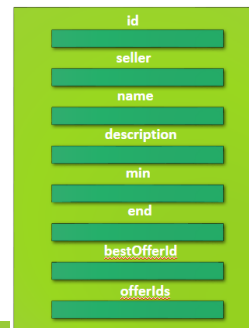
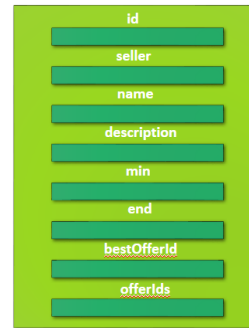
```
struct Offer {  
    uint id;  
    uint auctionId;  
    address payable buyer;  
    uint price;  
}
```



Mapping

```
mapping(uint => Auction) private auctions;
```

uint	Auction
0	→
1	→





← auctions[1]
← auction



```
function createAuction(string calldata _name,string calldata _description, uint _min) external{
    require(_min>0,"minimum must be greater than 0");
    uint[] memory offerIds = new uint[](0);

    auctions[newAuctionId] = Auction(newAuctionId,payable(msg.sender),_name,_description,_min,0,offerIds);
    auctionList[msg.sender].push(newAuctionId);
    newAuctionId++;
}
```

auctions

uint	Auction
1	1, 0xa, car, beautiful, 10, 0, 
2	2, 0xa, mobile, nice, 5, 0, 

auctionList

address	uint[]
0xa	 

mapping(uint=>Auction) public auctions;

mapping(address=>uint[]) private auctionList;

```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

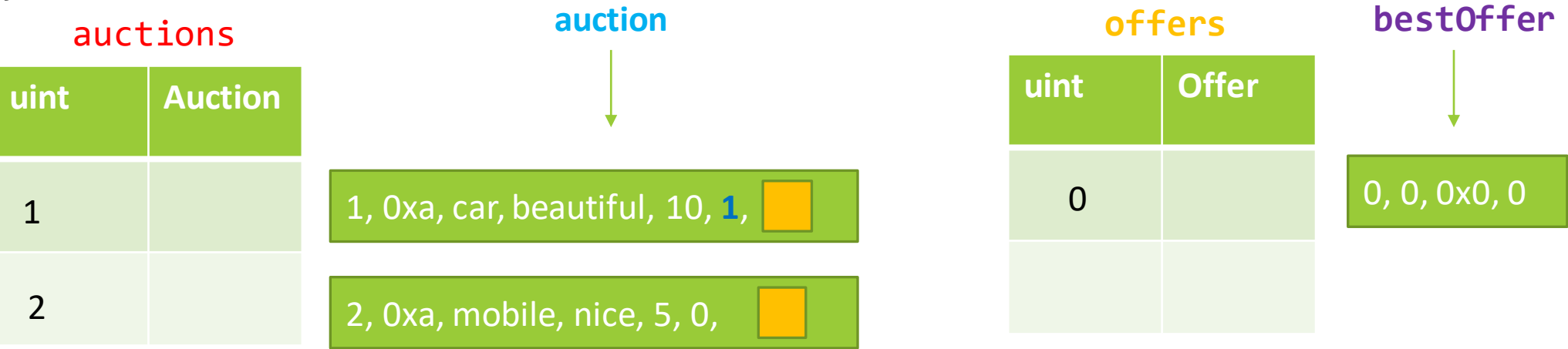
    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```



```
mapping(uint=>Offer) private offers;
```

```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```

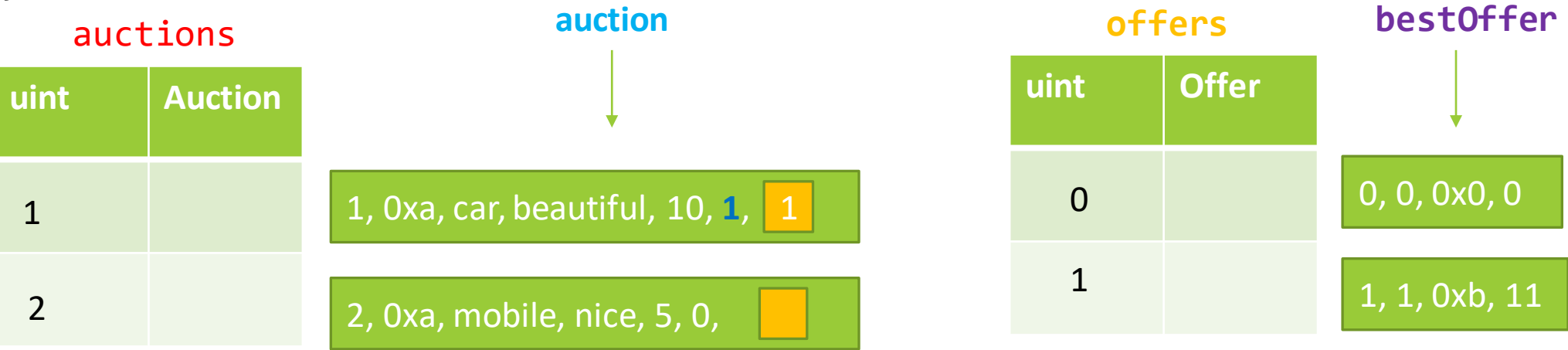


```
mapping(uint=>Offer) private offers;
```



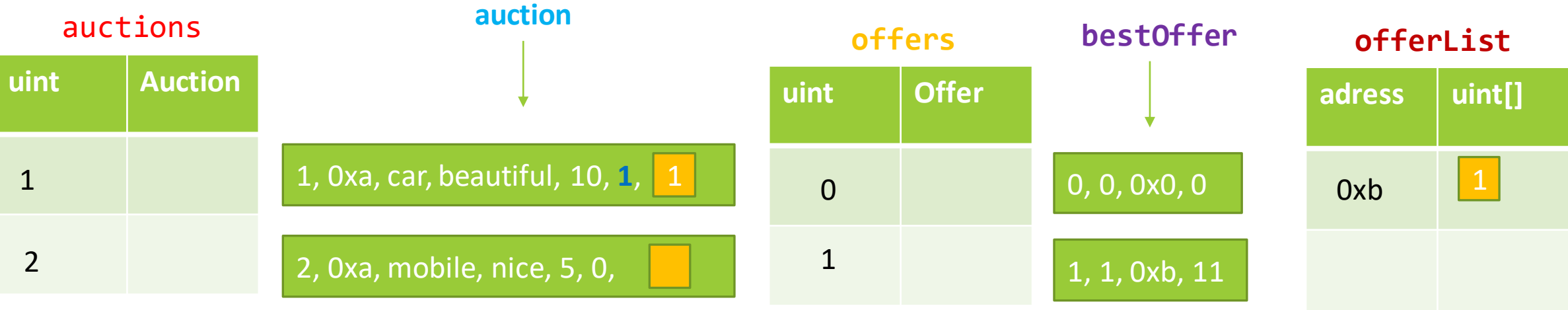
```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```



```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

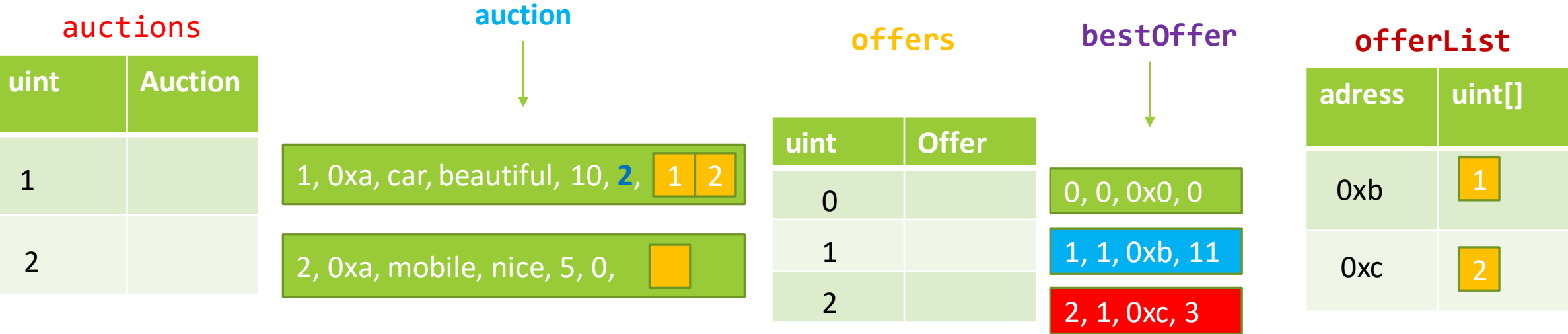
    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```



mapping(uint=>Offer) private **offers**;

```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

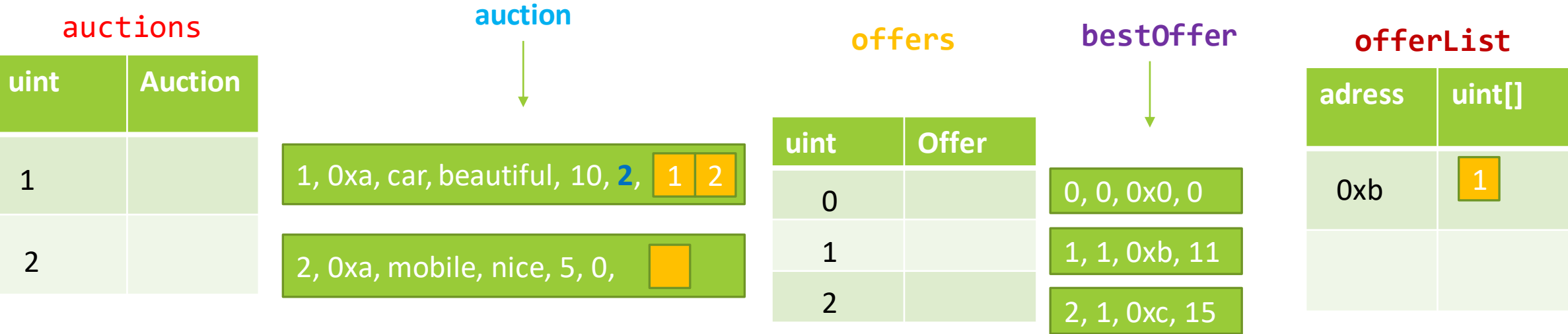
    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```



mapping(uint=>Offer) private offers;

```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

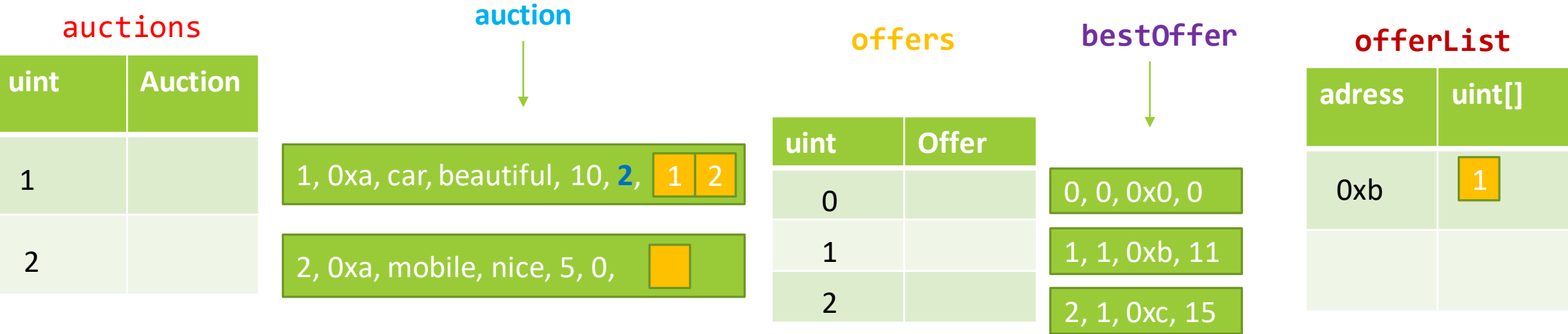
    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```



mapping(uint=>Offer) private **offers**;

```
function createOffer(uint _auctionId) external payable {
    Auction storage auction = auctions[_auctionId];
    Offer storage bestOffer = offers[auction.bestOfferId];
    require(msg.value>=auction.min && msg.value>bestOffer.price,"msg.value must be greater than the minimum and the
best offer");
    auction.bestOfferId = newOfferId;
    auction.offerIds.push(newOfferId);

    offers[newOfferId]=Offer(newOfferId,_auctionId,payable(msg.sender),msg.value);
    offerList[msg.sender].push(newOfferId);
    newOfferId++;
}
```



mapping(uint=>Offer) private **offers**;

Congratulations

Thank You

Please Like and Subscribe !

Instagram – @codeeater21

Discord – Blockchain Developer Community (Link in Description)

E-Ticket

Order: 859925187947268794



IMPORTANT INFORMATION

Mandatory check-list for passengers :

1. Please download and activate the Aarogya Setu app on your phone.
2. Certify your health status through the Aarogya Setu app or the self-declaration form.
3. Remember to do web check-in before arriving at the airport. Please do carry an e-boarding pass on your mobile. Alternatively, you can carry the printout of the boarding pass.
4. Please reach at least 2 hours prior to flight departure.
5. No meal service will be available on-board.
6. Face masks are compulsory. We urge you to carry your own.
7. You are requested to print and paste the baggage tag attached to your booking confirmation. Alternatively, you can write your name, PNR and flight number on an A4 sheet and affix on your bag.

Trip 1: Bangalore to Bhubaneswar

Bangalore to Bhubaneswar

AirAsia I5-1567 Tue, 11 Oct 2022

BLR 06:50 Tue, 11 Oct 2022 Kempegowda Int'l Airport	 1H50M ECONOMY	08:40 BBI Tue, 11 Oct 2022 Biju Patnaik Int'l Airport
--	----------------------	--

Baggage Info - Check-in: 15KG, Cabin:7KG

Meal info: Meal will not be served in E and O Class booking

Please check with airline for terminal information

TRAVELLERS	AIRLINE PNR	TICKET NO	SEAT NO
Mr. PREETAM KUMAR SAHU	Y51G7G	Y51G7G	



Best Price Gauranteed

Find a lower price? We'll refund
20% for the difference



Dedicated Support Team

We are always trying our best to assist you.
Reach us at cs@happyeasygo.com



Super Instant Offer

Konw more!
Visit www.happyeasygo.com

E-Ticket

Order: 859925187947268794



About This Trip

- Use your Order ID %s for all communication with HappyEasyGo about this booking.
- Check-in counters at all airports close 60 minutes before departure.
- Your baggage shouldn't weigh more than the limits mentioned above.
- Carry your photo identification and print this ticket. You will need them as proof of identity while checking-in.
- We advise you to go to 'Edit Booking'/'Manage Booking' section on Indigo/SpiceJet official site to shorten the waiting periods of your rescheduling requests.
- Terminal is subject to change, please check with airline before going to airport.
- Due to ongoing Covid-19, a web check-in is mandatory for all passengers taking flights. Personal info including email and mobile number must be completed.
Click: <http://hegurl.com/1PI2Ax5n> .

Download our mobile Apps



Fare Breakup

Airline charge	Rs. 6,215
Convenience fee	Rs. 400
Reverse of convenience fee	Rs. -400
Discount(Other discount)	Rs. -320
Tax	Rs. 0
Meals	Rs. 0
Seats	Rs. 0
Baggages	Rs. 0
Grand total Rs. 5,895	



Best Price Gauranteed

Find a lower price? We'll refund 20% for the difference



Dedicated Support Team

We are always trying our best to assist you.
Reach us at cs@happyeasygo.com



Super Instant Offer

Konw more!
Visit www.happyeasygo.com

INVOICE



Invoice Date: 09/Sep/22
Invoice No: HEGFD20220162005

CIN No.

U63030DL2017PTC312927

Pan No.

AAECH0804J

GST No.

06AAECH0804J1ZW

Client Name:NA

State:NA

Client Address:NA

GSTIN No:NA

PAX NAME

PREETAM KUMAR SAHU

Trip 1

Trip ID: 859925187947268894

PNR NO

Y51G7G

BOOKING ON

09/Sep/22

TRAVEL DATE

11/Oct/22

FLIGHT NO

I51567

SECTOR

BLR-BBI

TICKET NO

Y51G7G

Airline charge	Rs. 6,215
Convenience fee	Rs. 400
Reverse of convenience fee	Rs. -400
Discount(Other discount)	Rs. -320
Tax	Rs. 0
Meals	Rs. 0
Seats	Rs. 0
Baggages	Rs. 0

Grand total Rs. 5,895

Note: This is an electronically generated invoice and does not require a physical signature or stamp.