CarConTracks - Car Rental DApp Whitepaper

EECE 571G

Dr. Zehua Wang

Negar Yassaie, Sara Gargoum, Mateusz Faltyn

Introduction

As COVID-19 pandemic restrictions continue to lift across Canada and much of the rest of the world, the commuting landscape looks very different than it had before March 2020. With many large knowledge work workspaces, such as university and tech campuses, closing for intermittent periods, working from home eliminated the need for a daily commute for many individuals. Knowledge workers who were able to shift most, if not all, of their work to home were posed with an interesting question: without a commute to work, what is the need for purchasing a vehicle?

The overall trend is clear: individuals want to increasingly rent, not buy, vehicles [1], [2]. Global vehicle sales have been decreasing since their peak in 2016 with a decrease of around 13.8% between 2019 and 2022 [3]. Additionally, global vehicle production has fallen from around 97 million per year in 2017 to 78 million per year in 2020 [3]. With the rise of app-based "on-demand" transportation options such as ride-hailing, carsharing, and scooter-rentals, individuals living in ever increasingly urban areas are finding that renting a car is both the more affordable and accessible mode of transportation.

"On-demand" ride-hailing services such as Uber and Lyft have steadily seen a rise in customer usage in recent years. Similarly, "on-demand" car-sharing services such as Modo and evo have spread to many urban areas across the world. However, all of these services suffer from issues of centralization such as security issues, privacy issues, and central points of failure; there is a need for decentralized solutions. Our aim in this project is to fill this gap in the market by creating a peer-to-peer car rental decentralized application.

Description

CarConTracks will be an Ethereum-based [4] DApp that allows individuals with the appropriate legal qualifications (i.e., age and driver's license) to rent a car for a specific duration of time. Instead of signing in to an app of a centralized carsharing service provider such as Modo or evo, users of CarConTracks will be able to rent a car in a peer-to-peer manner anywhere across the globe where cars are located.

Stakeholders

The following are stakeholders of CarConTracks:
1. Renters - are users of the CarConTracks DApp who wish to rent a car for a specific period of time (consumer).
2. Rentees - are users of the CarConTracks DApp who wish to put their car up for rent for a specific period of time (provider).

Business Model

CarConTracks has a simple commission-based business model: Rentees provide vehicles to renters while receiving 80% of the transaction value after gas fees. This commission-based business model has been tried and tested throughout a number of companies such as Uber and Lyft to great success across the world. All expenses related to vehicle operation will be covered by the renter during the specified rental period or the rentee outside of a specified rental period.

Approach

Architecture

Figure 1 displays the system architecture of the CarConTracks DApp.

Figure 1. CarConTracks Architecture

Assumptions

To simplify the DApp architecture, the following assumptions were made:
1. Two types of users: renters and rentees.
2. Renter must be 18 years or above of age.
3. Renterser must provide a valid license number.
4. Rental duration is one day.
5. Renter can only have at most one rental request.
6. Renter cannot make another rental until they cancel/complete the current rental.
7. A historical record of users must be saved.
8. Car rentals are offered at a fixed price.
9. Rentee can only offer one car for rent at a time.

Basic Functionalities

1. Rentee will make their car available for rent and post their car information.
2. Renter will be asked to enter their name, age, and valid license number for verification purposes.
3. Renter is able to locate the nearest available car.
4. Renter will be able to identify the model description of the nearest available car.
5. Renter is able to create a rental order for said nearest available car.
6. Renter is able to confirm the rental order.
7. Renter is able to cancel the rental order.
8. Renter is able to return the car after the rental duration is complete.
9. Renter can manually update the web interface in addition to the interface automatically updating at a reasonable rate.

Transactions

1. Renter must deposit ether into the smart contract to rent the nearest available car (gas fees apply).
2. Rentee must receive ether placed into the smart contract if the rental is successful (gas fees apply).
3. Renter must receive ether placed into smart contract if they cancel their rent order (gas fees apply).
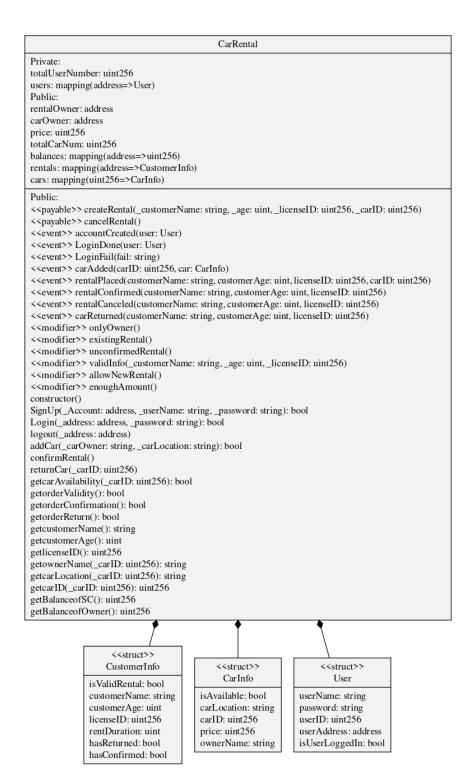
Smart Contract UML

```
                                    CarRental
─────────────────────────────────────────────────────────────────────────────
Private:
totalUserNumber: uint256
users: mapping(address=>User)
Public:
rentalOwner: address
carOwner: address
price: uint256
totalCarNum: uint256
balances: mapping(address=>uint256)
rentals: mapping(address=>CustomerInfo)
cars: mapping(uint256=>CarInfo)
─────────────────────────────────────────────────────────────────────────────
Public:
<<payable>> createRental(_customerName: string, _age: uint, _licenseID: uint256, _carID: uint256)
<<payable>> cancelRental()
<<event>> accountCreated(user: User)
<<event>> LoginDone(user: User)
<<event>> LoginFail(fail: string)
<<event>> carAdded(carID: uint256, car: CarInfo)
<<event>> rentalPlaced(customerName: string, customerAge: uint, licenseID: uint256, carID: uint256)
<<event>> rentalConfirmed(customerName: string, customerAge: uint, licenseID: uint256)
<<event>> rentalCanceled(customerName: string, customerAge: uint, licenseID: uint256)
<<event>> carReturned(customerName: string, customerAge: uint, licenseID: uint256)
<<modifier>> onlyOwner()
<<modifier>> existingRental()
<<modifier>> unconfirmedRental()
<<modifier>> validInfo(_customerName: string, _age: uint, _licenseID: uint256)
<<modifier>> allowNewRental()
<<modifier>> enoughAmount()
constructor()
SignUp(_Account: address, _userName: string, _password: string): bool
Login(_address: address, _password: string): bool
logout(_address: address)
addCar(_carOwner: string, _carLocation: string): bool
confirmRental()
returnCar(_carID: uint256)
getcarAvailability(_carID: uint256): bool
getorderValidity(): bool
getorderConfirmation(): bool
getorderReturn(): bool
getcustomerName(): string
getcustomerAge(): uint
getlicenseID(): uint256
getownerName(_carID: uint256): string
getcarLocation(_carID: uint256): string
getcarID(_carID: uint256): uint256
getBalanceofSC(): uint256
getBalanceofOwner(): uint256
```

```
      <<struct>>            <<struct>>            <<struct>>
     CustomerInfo             CarInfo               User
──────────────────     ──────────────────     ──────────────────
isValidRental: bool     isAvailable: bool      userName: string
customerName: string    carLocation: string    password: string
customerAge: uint       carID: uint256         userID: uint256
licenseID: uint256      price: uint256         userAddress: address
rentDuration: uint      ownerName: string      isUserLoggedIn: bool
hasReturned: bool
hasConfirmed: bool
```

Figure 2. CarRental UML

Future Work

Advanced Functionalities

Time-permitting, we will implement the following features:
1. Google Maps interface via the Google Maps API to display the nearest available car.
2. Verification function to confirm whether a user has a valid license number.
3. User compliance features to the DApp.

Tokenomics

Time-permitting, we will launch a new ERC-20 token entitled the Car Rental (CARR) token. In accordance with the tokenomics theory proposed by van Oordt in 2016 [5], the CARR token must take three major factors into account in order to maintain long-term price stability: 1) token utility, 2) total supply regulation, and 3) token adoption.

Token Utility

The CARR token will be used to post car information to initiate a loan as well as rent a car on CarConTracks. In essence, it will be the medium of exchange between the renter and the rentee. Without acquiring the CARR token, users will not be able to utilize the DApp.

In essence, the flow of CARR tokens will follow this path. 1) a renter buys N CARR tokens using software such as Metamask. 2) a renter selects a vehicle to rent for a duration of time and transfers $N - G$ (G = gas fees) tokens to the rentee. 3) In accordance with the aforementioned business model, the rentee will receive $R = 0.8 (N - G)$ of the CARR tokens from the renter. 4) The rentee can now become a renter with their newly acquired CARR tokens or sell their CARR tokens for profit.

Total Supply Regulation

In order to not price out new renters or rentees, the total supply of the CARR token will be regulated using a model similar to that of other utility tokens such as the Basic Utility Token launched by Brave Software [6]. The exchange rate of the CARR token will be proportional to the number of car rentals completed while being inversely proportional to the number of tokens not in use during a specific time period. In essence, the CARR token will increase in value if the number of users transacting on the CarConTracks DApp decreases. Holding CARR tokens will equal the discounted expected future exchange rate minus the risk premium for the expected uncertainty in the future value of the CARR token. Taken together, this model will make up the supply and demand of the CARR token and help stabilize the token value.

Token Adoption

Due to the tokenomics described in the previous paragraph, new users will be incentivized to join the platform as the CARR token will increase in value if the number of car rentals drops in

number. Since CarConTracks is a peer-to-peer car rental service, renters will have access to cars in areas wherever rentees are interested in putting their cars up for loan.

References

[1] M. Garaus and C. Garaus, "The Impact of the Covid-19 Pandemic on Consumers' Intention to Use Shared-Mobility Services in German Cities". *Front. Psychol. 12:646593*. doi: 10.3389/fpsyg.2021.646593

[2] K. M. Kiran Raj and K. G. Nandha Kumar, "Impact of Covid-19 Pandemic in the Automobile Industry: A Case Study", *International Journal of Case Studies in Business*, IT, and Education (IJCSBE), vol. 5, no. 1, pp. 36–49, Feb. 2021, doi: 10.5281/zenodo.4505772.

[3] "Topic: Automotive industry worldwide", Statista, 2022. [Online].

[4] V. Buterin, "A next-generation smart contract and decentralized application platform", Whitepaper, 2014

[5] W. Bolt and M. van Oordt, "On the Value of Virtual Currencies", *Tech*, Working Paper, 2016.

[6] Brave Software. "Basic Attention Token (BAT): Blockchain-Based Digital Advertising", Whitepaper, 2021.