PEBERS Architecture

Arpit Umap
Student , Computer Engineering
Trinity Academy of Engineering
Pune , India
arpitumap123@gmail.com

Prof. Alfiya Shahbad
Professor , Computer Engineering
Trinity Academy of Engineering
Pune, India
alfiyashahbad.tae@kjei.edu.in

***Abstract*—Ride Sharing has seen a steep rise in popularity in metropolitan cities to avoid wastage of resources, traffic jams and congestions. This gave birth to a whole new trend which ride aggregator services went on to capitalize by providing the option of sharing cabs to its users at a lower price.Although the current ride hailing services have revolutionized the transportation industry in today's world, they are extremely centralized.**

In this paper we propose our system PEBERS: Practical Ethereum Blockchain based Efficient Ride Hailing Service, in which we demonstrate how decentralized system based on consortium blockchain can be developed to keep track of ride data. In this context we explore smart contracts to build and deploy functionalities such as create ride, auto deposit transfer, cancel and complete ride methods of our decentralized ride hailing application. Our experiments show the driver smart contracts proposed in our system consume less Gas and hence prove that its an efficient system than many other existing systems with respect to the expenses of passengers and profitability of drivers. This is an effort towards providing smarter transportation for a society moving towards smart city concept.

***Index Terms*—Keywords: Blockchain, Decentralize, Ride hailing, Smart contract**

I. INTRODUCTION

Ridesharing is a facility that arranges one-way transportation on short notice through mobile apps and websites. To make the overall ride more affordable and environment friendly, the system groups users going in the same direction together and then splits the cab fare. The industry is booming, with clients ready to pay on-demand providers for convenience and a lower fee. However, there are challenges too. The identification and subsequent addressable of these challenging factors are of utmost importance to cab service providers, before consumers lose interest [3].

Currently, cab service aggregators are using a centralized methodology to carry out their day-to-day operations. The policies, rules and regulations, terms and conditions that both the user and the driver must follow vary from company to company. Furthermore, the booking of cabs requires mediators or third-party businesses to carry out the payment process. With more parties involved, this proves to be problematic with the creation of a lack of transparency [3] . These disadvantages have led to an extensive study of the blockchain technology and subsequently several proposals of ride-sharing architecture built atop the blockchain.

A. WHAT IS BLOCKCHAIN?

Blockchain is a public, immutable ledger for tracking resources, recording transactions and building trust. Anything asset (tangible or intangible) can be tracked and traded on a blockchain network, with the main advantage being the reduced risk as well as significant cut in costs for all parties. Every sector in every field is built on data.Most businesses operate solely because of the transfer of information, the faster this happens, the better. Blockchain is perfect for the movement of data because it can provide prompt, shared and completely transparent information that will be kept on an incontrovertible ledger that can be retrieved only by those who are authorized to do so. The most important feature of blockchain is that all users share a single view of the truth, so each member can see all the particulars of a transaction from the very beginning, giving members greater confidence while also increasing efficiency and giving rise to a plethora of applications blockchain can be used for. To further reduce transaction time, a set of rules known as the smart contract is stored on the blockchain and executed automatically. It is used to define conditions for transfer, include terms for different bonds and so on [3] .

In this paper, we explore partly private network of block chain called consortium blockchain with few authorized nodes to develop an efficient ride hailing decentralized application. The idea of partly private blockchain is unique, in that selected nodes are authorized to take part in consensus and add blocks to the chain. The write operation is limited to 2 these nodes only while the read can be performed by anyone in the netwrok. Additionally, user data privacy motivates us to choose private rather than public blockchain. This feature makes it possible to restrict the visibility of information on the network but yet make it decentralized. Furthermore it decreases the cost of establishing a blockchain unlike traditional public blockchain [2] .

B. PEBERS Model

PEBERS model, utilizes fog computing nodes as authorized nodes. Fog computing nodes are road side units with storage, computing and communication capabilities. These nodes are semi trusted and are distributed area wise in the network. This feature will overcome centralized server concept and bring about benefits such as location awareness, low latency to our

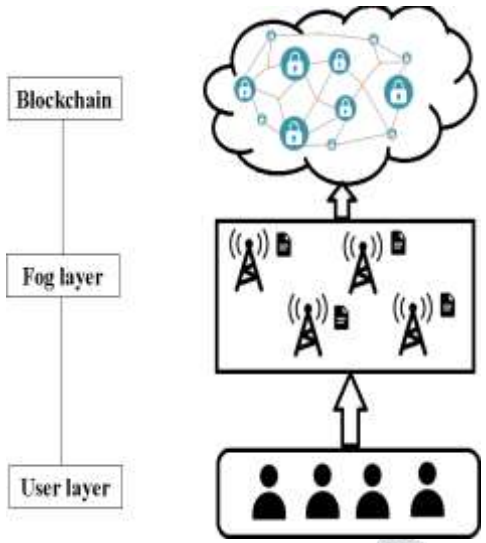

Fig. 1. System Overview

design. Furthermore, they serve as the agents to find and match passengers and drivers.

The main contributions of this paper are summarized as:-

- We build and deploy a prototype of driver smart contract that will be deployed individually by each driver in the network and we investigate the expenses incurred for both drivers and passengers by employing such a design.
- We propose to construct a consortium blockchain formed by fog computing nodes as shown in Fig 1. into our system PEBERS to record ride transactions in a verifiable and immutable ledger. Fog computing nodes help in minimizing delay in response.
- One of the most crucial decisions is to determine which participants will be the miners. We propose to make use of delegated proof of stake consensus algorithm in which there are no miners but validator nodes who are assigned leader role randomly. With this we ensure there is no forking of blockchain at any given time [2] .

II. RELATED WORK

This section is a study of few of the most popular existent proposals of Peer-to-Peer Ride Sharing Architecture.

1) B-Ride acquaints a reputation model which rates drivers built on prior behavior, allowing riders to select based on the collection of interactions of the drivers. The confirmation is done using zero-knowledge proof to protect rider/driver privacy. To ensure fair payment, a pay-as-you-drive philosophy is presented.
2) Green Ride promotes social commitments by implementing decentralization where it facilitates carbon emission reduction. It consists of two structures; centralized code that will dwell on Google Cloud App Engine and the decentralized GRTs. It leads to businesses, colleges, and government agencies reducing their annual carbon footprint.
3) O-Ride, a privacy-preserving system optimizes SHE so that bandwidth requirements and processing overhead are lessened using ciphertext packing and transformed processing. It includes features such as credit-card payment, contacting drivers in the event of missing belongings, and traceability in the event of criminal activity.
4) Block-VN is a distributed vehicle network architecture. It examines how the network of vehicles evolves with paradigms. The department of vehicles transmits details to the revocation authority each time a vehicle registration is issued. The revocation authority then informs the distributed blockchain of all information about ordinary and miner vehicle nodes [3].

TABLE I
MAIN NOTATIONS USED IN THIS PAPER

Notation	Description
Di = {d1, d2, d2, d4..dn}	set of n drivers
Pi = {p1, p2, p3, p4..pm}	set of m passengers
PKdi, SKdi	keys of driver Di
PKpi, SKpi	keys of passenger Pi
repdi	Reputation of driver Di
reppi	Reputation of passenger Pi
Iddi	Driver Di's id
Idpi	passenger Pi's id
Idfi	fog node's id
ldi	location of the driver Di
lpi	location of the passenger Pi
REQdi	Request message from the driver Di
REQpi	Request message from the passenger Pi
Opi	Origin of passenger
GOpi	Get-off location of the passenger
distpi	passenger's total distance of travel
rideCostdi	price per mile quote from the driver

III. BACKGROUND

A. Smart contracts and Solidity

Smart contracts are neither smart, nor contractual by nature. They can be defined as contractual constraints deployed on the blockchain that act as an immutable agreement, and can receive or execute transactions. Smart contracts are developed using a Turing-complete scripting language called Solidity, after developing the smart contract it is compiled into EVM byte code and then deployed on the blockchain.

B. The Ethereum Blockchain

The Ethereum blockchain is a consortium blockchain that enables people to develop and deploy their own decentralised applications. It also offers a programming language called solidity which enables anybody to develop smart 3 contracts and decentralised applications. Users can create an Ethereum

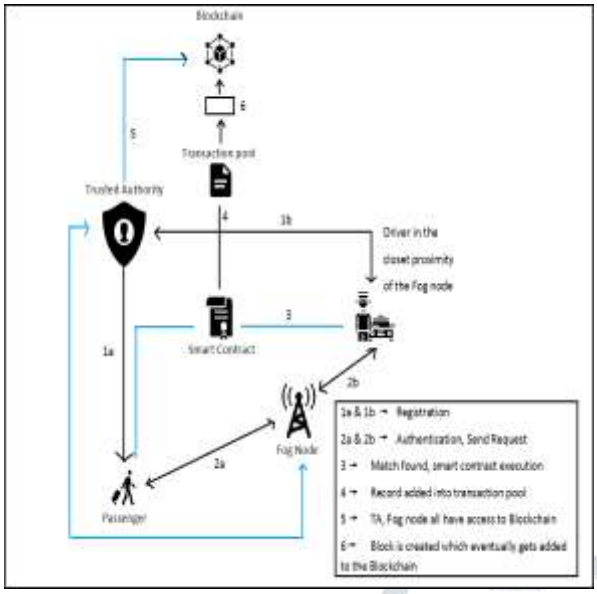

Fig. 2. System Architecture

account which is assigned with an address. Every computational step of transaction made has an associated Gas price.

C. Cryptocurrency

Cryptocurrency is a term given to the digital form of money, it is a decentralised monetary digital asset which uses cryptography to ensure secure exchange or transactions. The Ethereum blockchain has its own cryptocurrency called Ether. When users create Ethereum accounts they load ethers to transact [2].

IV. PROPOSED SYSTEM MODEL

A. System Components

Fig. 2 consists of functional components of our system namely Trusted Authority (TA), Fog computing nodes, n number of drivers and m number of passengers. Each of the entity's responsibilities are explained as follows:

1) Trusted Authority: TA should initialise the whole system, and generate cryptographic keys for fog nodes, passengers, and drivers. When a user registers to the network, their identity is verified by this trusted authority. Then keys get generated as shown in Fig 2. It keeps tracks of every user and their corresponding key pairs in encrypted format.
2) Fog computing nodes: Fog computing nodes are road side units with computation, communication and network capabilities. We use a set of fog nodes as a few semi trusted nodes in the network. They are assumed to be installed on area wise basis. Additionally they serve as agents to find and match passengers with drivers.
3) Drivers: Drivers register with the network and receive a pair of private key SKdi and a public key PKdi along with a unique iddi that defines their identity.
4) Passenger: Passengers also register to the network. They will have a matching private SKpi and public key PKpi set with a unique idpi [2].

V. METHODOLOGY

1) Registration phase: Registration occurs at the system users layer which encompasses all the different entities of the ride hailing application, including drivers and passengers. Firstly, users sign-up to the application. This phase is executed only once in the process. During this step, users can choose either driver or passenger as their account category. We assume that Di Pi = to eliminate complex scenarios. Notations have been summarized in the table 1.
Users register with the Trusted Authority(TA) as shown in Fig 2. Our assumption is that all the communication happening between TA, passengers, and drivers is through secured protocol. Designing a secure protocol is out of the scope of this paper. TA provides the key pair to the user in a secure manner at the time of registration after a successful verification. TA is trusted by all parties. After passing the identity authentication by a TA, a legitimate driver di obtains its public key PKdi and private keys SKdi and a unique id Iddi from cryptographic module of TA server. It can be represented as below:
$$di \rightharpoonup \{Iddi, SKdi, PKdi\}$$
Passengers also sign up to the application using the mobile devices through a secure protocol. A legitimate passenger Pi receives his own PKpi and SKpi keys and identity idpi from cryptographic module of TA. It can be represented as below [2] :
$$pi \rightharpoonup \{Idpi, SKpi, PKpi\}$$
After users have been registered and credentialed, they can use their digital identities ie Idpi, Iddi to access the services.
2) Authentication phase: This phase is to authenticate the registered users through the nearest fog computing node. When a user first accesses the application, our Authentication smart contract checks if the user address is valid. After successful authentication, users can start using the service.
3) Deploying driver smart contract: A driver is uniquely identified by his iddi in the network. The registered driver deploys the smart contract on the network and deposits one ether initially on the smart contract storage. Algorithm 1 represents our driver smart contract.
4) Ride Requesting phase: In this phase, passenger submits a ride request to the nearest fog computing unit. A registered passenger pi looking for a ride, holding an identity idpi and a reputation value reppi creates a request REQpi with her current location Opi , destination location GOpi, total distance of travel distpi as follows. This request is received by the nearest fog computing node. Passenger request REQpi can be represented as below [2]:

REQpi ={idpi,Opi, GOpi, distpi }

5) Matching protocol by fog unit: When a request is received by a fog node holding id Idfi it tries to perform one-to-many dynamic matching based on the location of the passenger and returns the list of currently available driver details.
Interested drivers routinely query the topologically near-est fog computing units for ride requests from passen-gers. Each driver within the desired region of passenger, first checks the REQpi from passenger and sends their response by specifying their price per mile quote. Driver response contains these details [2]:

RESdi ={ iddi, repdi, rideCostdi }

6) Accepting ride: If the passenger pi does not want to accept anyone from the list of waiting drivers, fog node discards the request after a certain time out period. Otherwise, it sends an Acknowledgement message to the driver di that is selected by passenger pi. Driver di acknowledges back to the notification.
7) Pick Up: Finally, pi can communicate with matched di for follow-up communications, such as a specific pickup location Opi, and a get off location GOpi.
On pickup, the passenger deposits fixed number of ethers into driver smart contract storage and passes encrypted passenger details. On ride completion, driver and passenger both sign the transaction which transfers the cost of ride to driver automatically. Rest of the deposit is returned to passenger address. Users leave a rating to each other which gets accumulated in the reputation parameter.
8) Consensus mechanism: Blockchain is based on Ditributed Ledger technology(DLT) that stores data in a decentralized manner. ie, each trusted node has its own replica of the ledger. Because of its distributed nature, there has to be a consensus protocol to ensure the consistency of the ledger. We assume that there is a record pool containing all the records of every ride made and is to be added on the block chain.

- At a given time, every fog node pulls some records from the transaction pool and verifies the correct-ness of the transaction by validating the signatures of driver di and passenger pi
- The number of transaction verified by each fog computing node is associated with its stake. For every time period, one random fog computing node becomes the leader Lfi, and has the right to generate a new block out of the transactions it has verified. The probability of a fog node idfi being selected as a leader node is proportional to its stake recorded in previous block.
- The selected leader fog computing node generates a new block Bi consisting of a block header, fog node's identity idfi,number of records of transaction that are validated, 5 that is counted as its stake and its signature fi.This block gets added to the blockchain and is now permanently written to the distributed ledger and then the network is notified to sync.

9) Ride Cancellation: If a passenger or driver decides to drop off from the ride they can call a cancel function of the smart contract. However if the ride is cancelled after selecting the driver, some amount of ethers from the deposit is transferred to the driver account to respect their time. Similarly, if the driver is cancelling the ride for some reason, ethers from the initial deposit will be transferred to passenger's account [2].

Algorithm 1: Algorithm for createRide contract [2].

```
1.  constructor (_createRide) payable
    // constructor is "payable" so it can receive the initial deposit
2.  require(msg.value == 1 ether);
3.  set owner = msg.sender
4.  rideState ⟶ null
5.  driverState ⟶ idle
6.  rideCost ⟶ fixedrate;
7.  Function newRide() payable public
    //makes a new ride contract instance
    // with new passenger address
8.  if (contractInstance.driverState == 'Idle') (msg.value == 1 ether) then
9.      rideState='inTransit'
10.     driverState='busy'
11.     passengerAddress= msg.sender;
12. end if
13. Function rideCompleted ( rideCost, distpi) public
14. set driverState = idle
15. set rideState = rideCompleted
16. const cost = call RideCost( rideCost, distpi)
    // function to calculate the ride cost
17. if (cost != 0) then
18.     transfer(balance.driveraddress)
19. end if
20. Function rideCancellation () public
    // if passenger cancels the ride
    // transfer smart contract
    // balance to driver
21. if (contractInstance.rideState == Cancelled) msg.sender !=owner then
22.     transfer(balance.driveraddress)
23. else
24.     transfer(balance.passengeraddress)
    // transfer a deposit from the driver to the contract
25. end if
```

VI. APPLICATION OF BLOCKCHAIN CONCEPTS IN RIDE -SHARING

1) Blockchain based firmware update scheme: Autonomous vehicles manufacturers form a consortium blockchain

ensuring high availability and quick delivery of products and updates with low computational cost which is resistant to a DoS attack. Attribute-Based Encryption (ABE) generates an access policy that ensures that only approved autonomous vehicles may download and install new updates while also utilizing a smart contract to assure the validity and integrity of firmware updates. Due to the limited time required for cryptographic computations and the transfer time, the scheme can be implemented during the contact time of two moving autonomous vehicles.

2) Use of Zero Knowledge Proof Module: In a volatile environment a zero-
knowledge proof protocol is utilized. In exchange for proofs of distribution from receiver AVs, each distributor can trade an encrypted version of the update. The smart contract guarantees the delivery of the decryption key, which will be revealed after the proofs are collected. Based on the received proof, the smart contract also increases the distributor's reputation.
3) Use of Incentives and Rewards: A reward mechanism is designed to incentivize autonomous vehicles to distribute Firmware updates for consortium blockchain by maintaining a credit reputation for each distributor account in the blockchain.
4) Use of Smart Contract: Consider a Blockchain-based service that provides smart contract templates for drivers and passengers. The two parties will choose a "basic" smart contract template initially (for example, the transfer of goods or people; rewarding the driver in fiat currency or cryptocurrency; payment in cash, or through reward points;) The parties will then agree on the transaction's specifics (For instance, the precise fee to be paid; the choice to carry more people or not;) Individuals will no longer require a third party to complete the transaction since the Smart Contract template will ensure that either both sides of the transaction are fulfilled, or none at all.
5) Use of Pseudonymity: Pseudonymity is defined as the usage of aliases (pseudonyms) for confidentiality for the purpose of either shielding one's identity, achieving self-sovereignty, or for privacy and security concerns. Privacy in Blockchain refers to the preservation of anonymity and the unravelling of transactions. Transaction anonymity entails that it is impossible to connect each transaction to a unique user. Consequently, the user makes use of a unique address for every single new transaction. Unravelling makes the assumption that Blockchain addresses and transactions are not linked to the real user identities [3].

VII. CONCLUSION

In this paper we have proposed a scheme to use consortium blockchain technology for ride hailing services combined with smart contracts. Crucial feature is our driver smart contract that is individually deployed by each driver. We evaluated and ensured that it costs less for driver and passenger by employing this design. Further more we showed how a initial deposit from both parties brings trust in the system. Exploring decentralized application development on blockchain platform proves to be valuable.

REFERENCES

[1] V´azquez, E. and Landa-Silva, D., 2021. Towards Blockchain-based Ridesharing Systems. In ICORES (pp. 446-452).
[2] Kudva, S., Norderhaug, R., Badsha, S., Sengupta, S. and Kayes, A.S.M., 2020, February. Pebers: Practical ethereum blockchain based efficient ride hailing service. In 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT) (pp. 422-428). IEEE.
[3] Riddhi Gupta , Riya Gupta , Sonali Shripad Shanbhag, 2021, A Survey of Peerto- Peer Ride Sharing Services using Blockchain, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) Volume 10, Issue 08 (August 2021)