Ridesharing DApps - A Study on Peer-to-Peer Ridesharing on Ethereum

Tushar S Menon[1], Aviral Srivastava[2], Aditya[3], Dr. Radhika K R[4]

Department of Information Science and Engineering, BMS College of Engineering, Bengaluru, Karnataka, India

Abstract: ***Ridesharing is an effective method to resolve traffic congestion and also reduce pollution due to excess vehicles on-road. However, the centralized nature of the current ridesharing systems is not ideal for the user. The lack of transparency in the system as well as risk of data security is a big demerit for such a system. To keep the third-party involvement minimal, a trustless, decentralized peer-to-peer ridesharing DApp is being proposed using a private Ethereum blockchain. Credibility of ride sharing systems can be improved by implementing blockchain technology. Blockchains are decentralized databases where every single piece of information is stored on systems everywhere which can be retrieved and traced freely by anyone on the network. The system will no longer be trust-based but simply based on concrete proof that exists which is built into the ledger. In a blockchain-based system, a rider will anonymously post a ride request. A driver can accept the request and provide their id details and quote. The rider can choose if the transaction is fair and accept the quote and begin his ride. Various other concepts such as time-locked deposit and proof-of-elapsed distance have been introduced to ensure further security for driver and rider. The primary goal of such a system is to develop a reliable and transparent ride sharing system where users do not have to worry about their privacy.***

Keywords: blockchain, decentralization, encryption, peer-to-peer, ridesharing, intermediaries, DApp, Ethereum

1. Introduction

The proper operation of modern society is heavily reliant on a well-functioning transportation system. Any community's transportation system is crucial. With the fast introduction of new mobility services in recent years, intelligent transportation systems have changed traditional transportation supply. Ride-sharing is one of these services that is gaining popularity. During peak hours, the high proportion of solo drivers contributes to traffic congestion. Lower occupancy in vehicles that take up a lot of road space are one of the key causes of traffic congestion. The impact of traffic congestion on the economy and society is expected to grow considerably in the coming decades as cities expand.

Ridesharing, in the sense of carpooling, has developed as a form of transportation that has the potential to relieve traffic congestion by boosting average vehicle occupancy rates and reducing the number of vehicles on the road during peak commuting hours. Ride-sharing is an on-demand transportation service that attempts to promote sustainable transportation by reducing car usage, increasing vehicle occupancy, and increasing ridership on public transportation. Ride-sharing promises to reduce pollution, travel expenses, and congestion as well as boost passenger vehicle occupancy and public transportation utilization.

Even though it is clear that Ride-sharing is beneficial in numerous ways, it was found that even the environmentally-conscious are hesitant to use a ridesharing system. The reason was found to be the centralized nature of the existing systems where the user has no say in the policies that will dictate their ride. Drivers and riders must adhere to the rules of a third-party, who aren't even part of their ride, to make decisions. The lack of transparency in the decision making from the third-party is a point of concern for users which leads to less use of ridesharing services by the general public.

Consumers may now book a private or shared car with a few taps of a mobile application, thanks to ride-hailing apps that are primarily based on the on-demand availability business model. These approaches have grown more convenient than hailing a regular cab because payment is taken automatically from users' accounts. The centralized authority sets the pricing and service policies, which are related to the events. However, there are several drawbacks to such a management system. The system looks to be less transparent, rigid, and centralized, with a single entity controlling all aspects of the system and dictating regulations and service conditions.

Furthermore, questions have been expressed about the safety and security of client information as well as their trans-national data when using such centralized platforms. The central server is expensive to maintain and manage, and it is extremely vulnerable to distributed denial of service assaults. In addition, due to high bandwidth use and processing overhead, the response time of requests from a remote corporate cloud server result in an unnecessary increased response delay in present centralized systems. As a result, the present centralized ride hailing systems' flexibility, data integrity, and stability are all questioned.

The service provider acts as a go-between for system users, facilitating communication and charging a commission for each successful shared journey. However, using a central server to run the service exposes the system to a single point of failure and assaults. If the service provider's security is breached, the service may be disrupted, and data may be leaked, edited, or even erased. For example, Uber has been dealing with a massive data breach involving 57 million consumers and drivers for more than a year. Uber has paid a total of 148 million dollars to settle a data breach probe. [7] It is evidently clear that a centralized system cannot promote trust in its users, as a result of which, users are wary of using a ridesharing system for their daily commute, instead preferring traveling by their own vehicle at a low occupancy rate.

Introduction of a decentralized architecture for ridesharing could solve this issue. A peer-to-peer system in which the users have the power to control the policy of the ride without involvement of any third party will be attractive to many. The other benefit of such a decentralized system will be the lack of a single point of fault, i.e., during a data breach only a single user's data will be at risk while the other user's data stay safe. Utilizing blockchain technology can help achieve such an architecture that is trustless, transparent and reliable in nature.

2. Blockchain, Ethereum and Smart Contracts

Blockchain technology has recently changed several aspects of the information technology business. Blockchain is a decentralized database that keeps track of all transactions. Sending money to someone in the network is an example of a transaction. Although Bitcoin was the first to employ blockchain technology, there have been uses in a variety of industries other than banking since 2009. The application concepts are still being developed, tested, and refined over time. Blockchain is gaining traction in the ride hailing platform by allowing people to contact directly with drivers who are prepared to carry them. They help decentralize the whole architecture and make it peer-to-peer. By promoting cooperative management between passengers and drivers, blockchain-based ride hailing companies might address the aforementioned concerns. Participants distribute transactional data across a vast network of nodes rather than deciding on a single trusted centralized authority. This eliminates the need for intermediaries to act as gatekeepers. Transaction information is stored in a distributed ledger that is accessible to all peers in the blockchain, making it more transparent.

What makes blockchain so powerful is that it is based on a timestamped list of blocks that record, share, and aggregate data about all transactions that have ever taken place on the blockchain network. The block is added to the blockchain when it is completely filled. Every block is encrypted and connected to the previous block's hash code, making it impossible for anybody to alter the blocks. As a result, transactions are shared, irrevocable, unchanging, and highly secure records. These are without a doubt the most fascinating elements of blockchain that aid in the development of trust. The immutability and highly secure nature of blockchain ensures that any user on the blockchain stays protected and the transactions made by them cannot be altered.

Ethereum [19] is a Blockchain application that serves as a public medium on which anybody can build apps without having to invest in the time and money required for Blockchain development. Smart contracts (SCs) are Ethereum-based contracts that may be programmed in a variety of programming languages, including Java. However Solidity remains one of the more popular languages to create smart contracts. The Ethereum Protocol was created by the Ethereum Foundation and is a more advanced kind of crypto-currency that includes Blockchain escrow, gambling markets, and financial contracts. Since its inception in 2019, Ethereum's smart contract technology has drawn broad interest from government agencies, financial organizations, and technological firms.

Vujičić et al. (2018) explain in their paper how Ethereum works and how efficiently it implements smart contracts while drawing comparison to another popular blockchain Bitcoin.

Bitcoin as a cryptocurrency has gotten a lot of attention because it was one of the earliest implementations. They represent the very foundation of modern cryptocurrency development blockchain implementation with a focus on smart contracts.

Each transaction in a Bitcoin network has a hash value that represents transaction identification as well as a collection of inputs and outputs. In the whole blockchain, each transaction output can only be utilized once as an input. Each transaction is evaluated. Double-spending is caused by attempting to reference the same output twice, which is prohibited in the network. All transactions and ownerships in the Bitcoin network are recorded on the distributed ledger. This peer-to-peer network stores a copy of the ledger record on each node.

The Ethereum blockchain resembles the Bitcoin blockchain in appearance. The key distinction is that Ethereum blocks include the transaction list and the most current state in addition to the block number, difficulty, nonce, and other information. The previous state is applied to each transaction in the transaction list to establish a new state.

The block header in the Ethereum blockchain contains the Keccak 256-bit hash of the parent block's header, the mining fee recipient's address, hashes of the roots of state, transaction, and receipts tries, the difficulty, the current block gas limit, a number representing total gas used in the block transactions, timestamp, nonce, and several extra hashes for verification purposes.

Every node in the Ethereum network is controlled by EVM, which executes its commands. The nodes transform the smart contracts into EVM code, which they subsequently execute. Solidity is one of the most widely used programming languages for creating smart contracts as previously stated. The eligibility for ASIC mining is one of the most serious issues facing the Bitcoin network. Ethereum's proof-of-work method is Ethash, which is memory intensive and thus unsuitable for ASIC mining. The Ethash algorithm is a variation of the Dagger-Hashimoto method.

Ethereum is now moving on to Proof-of-Stake as its primary consensus algorithm. Proof-of-stake is a popular consensus algorithm that blockchains utilize to agree on the current state of the network and to reach distributed consensus. To validate a node in the blockchain, users must post their ETH as a stake. Validators are responsible for arrangement of transactions and creation of new blocks on the network so that all peers in the blockchain can agree on the it's state.

Today's most well-known and valued cryptocurrencies are Bitcoin and Ethereum. They are built on blockchain technology, which is designed to create a peer-to-peer network trust mechanism based on the consensus of the majority of nodes.

3. Decentralized Applications (dAPPS)

A decentralized application, often known as a dApp, is a software programme that runs on a distributed network. It is not hosted on a single server, but rather on a decentralized peer-to-peer network. Development of dApps are highly important for the popularization and development of blockchain technologies.

A Dapp is similar to any other piece of software you would use. It could be a website or a smartphone app. A dApp differs from a standard app in that it is created on a decentralized network such as Ethereum. In simpler words, the backend of a dApp is a blockchain network. In such architecture, a smart contract serves as the link to the backend.

Cai et al. (2018) in their paper discuss about the key performance criteria for an ideal dApp:

- Low Latency: The transaction delay should be minimal
- High Throughput: The dApp should be able to handle high transaction traffic on the network
- Fast Sequential Performance: Since all the transactions/operations must be processed and confirmed by all nodes in order to reach consensus in blockchain systems, the sequential performance of a dApp is governed by reaction latency from all nodes in the network. To handle enormous volumes, the blockchain platform that supports dApps requires fast sequential performance.

The authors also emphasize the need for low transaction fees to support the development of a dApp.

4. Centralized Ridesharing System and its Demerits

The current centralized system relies heavily on a third-party system to carry out all the processes like ride booking and payment transactions. The rider and driver have no say whatsoever in making the policies that dictate their trip.

The centralized authority sets the pricing and service policies, which are related to the events. However, there are several drawbacks to such a management system. The system looks to be less transparent, rigid, and centralized, with a single entity controlling all aspects of the system and dictating regulations and service conditions. Furthermore, questions have been expressed about the safety and security of client information as well as their trans-national data when using such centralized platforms. The central server is expensive to maintain and manage, and it is extremely vulnerable to distributed denial of service assaults.

The process model for a classic centralized ridesharing architecture can be described as follows:

1. The user searches for a cab using a cab provider's website or mobile app.
2. The user provides information such as pick-up location, drop-off location, time, number of passengers, automobile type, and payment method.
3. The request is directed to a nearby driver after being queued in a transaction pool.
4. The information of the driver and the user is communicated.
5. When the journey is finished, the customer pays using the app, and the cab company collects and sends the driver's bill after deducting a significant transaction fee.
6. All transactions or information exchange here happens through the service provider who is the central authority and has control over the data.

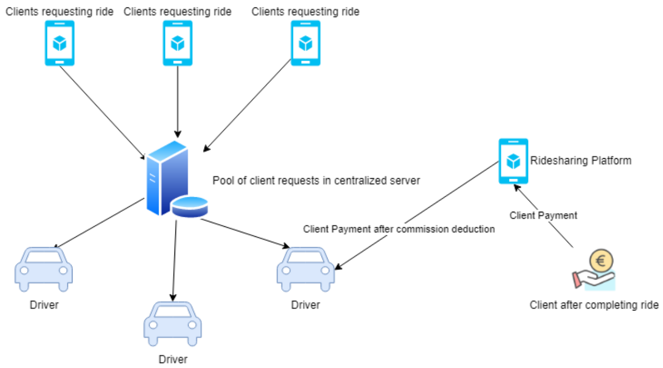


Figure 1: Centralized Ridesharing Model

The demerits of such a system:

- Costlier payments due to third-party involvement: The central party dictates the pricing policy of the ride. They also take a share away from the driver for their involvement in the affair.
- Insufficient transparency: Measures to provide transparency to the customers in the entire ride-making process is not implemented.
- Privacy and Security issue: Users submit key details about themselves to the third-party which can later abuse this knowledge.
- Employee Exploitation: Drivers can be forced to make a certain number of rides per day to stay employed. Drivers don’t have the comfort to decide their ride route or charges.
- Single point of fault due to trust-based centralization system: Since the entire system is centralized, any attack on the central figure, or any malicious behavior by the central figure will shut down the entire architecture.

5. Peer-to-Peer Ridesharing

A P2P ride sharing system is a ride booking architecture in which every decision policy is made by the peers of the network when they participate in the ride. Such a system has minimal third-party involvement. No central governing agency exists in such an architecture that would dictate the policies for each ride nor any mediating party that would resolve any dispute between two parties.

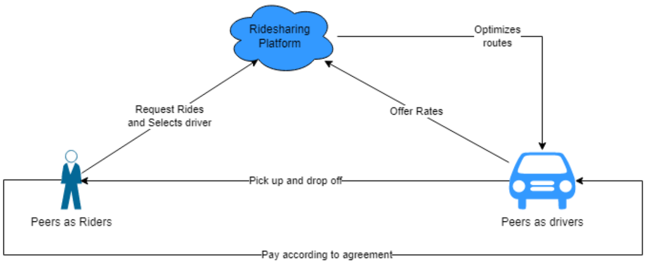


Figure 2: Peer-to-Peer Ridesharing Model

Ridhi Gupta et al. (2021) provide a better explanation of the concept in their paper. In a P2P Ridesharing Architecture, customers can find rides on the spur of the moment using a peer-to-peer ride sharing service, which allows them to communicate directly with neighbors via radio. Simply put, peer-to-peer transportation services enable drivers (peers) to provide on-demand transportation to people in need (peers) using their personal automobiles. Peer-to-peer networks may handle a wide range of hosts, including private automobiles and public transportation vehicles. These disparities in behavior have an impact on the agreements and, as a result, on travel options. Apart from the environmental benefits of fewer automobiles and hence less fuel use, such a system also offers up a whole new world of income prospects for car owners.

In their paper, Shrawani Silwal et al. (2019) discuss a distributed taxi-sharing system whose concepts are similar to a peer-to-peer ridesharing system. They emphasized that a faster processing time can be achieved by distributing the system, i.e., doing the complex computations at the vehicle level rather than at a central point. Wireless technology is used by both the cab and the passengers to send and receive requests and responses. When a new request is received, a permutation of all routes is computed and filtered according to the constraints. The shortest path is used to compute the distance cost.

The maximum service degradation factor, or the additional distance incurred as a result of ride sharing, must also be taken into account. When all of the constraints are met, the passengers are given a cost estimate. On the passenger side, an algorithm can choose the least expensive response.

Bozdog et al. (2018) built a fog-based decentralized ridesharing system which they named Ride Matcher. Participants having mobile devices (such as cellphones) join in a peer-to-peer method to share their rides in this system. Participants do not use a central database to find available rides, unlike a standard cloud-based service. Instead, they use short-range communication technologies like Bluetooth or Wi-Fi Direct to find other users who can give transportation that meet their requirements. The system, which runs on a mobile device, monitors the surroundings and attempts to discover rides that match a specified path, just like a person looking for a cab on the street.

The participating mobile nodes use this mesh network to advertise and find available rides. When two or more mobile nodes decide to share a ride, they form a ridesharing group. This happens on its own, without the need for any intervention.

Riders may communicate directly with drivers using the decentralized network, reducing the need for additional fees. People with a smartphone and secure modern automobiles have greater market opportunities because there are no intermediaries.

6. P2P Ridesharing Using Blockchain

In our study, we found various frameworks for peer-to-peer blockchain based ridesharing systems which utilize or apply various blockchain concepts for an efficient ridesharing architecture.

A basic framework for a blockchain based decentralized ridesharing architecture was explained by Ridhi Gupta et al. (2021) in their paper. They highlight a few key basic blockchain concepts that can be used to increase the privacy of the users in the blockchain while also maintaining transparency in transactions.

For the proposed framework the authors suggest the use of Blockchain Based Intelligent Transport Systems. A technology, application, or platform that employs applications to monitor, manage, or improve transportation networks is known as an intelligent transportation system (ITS). The Intelligent Transportation System relies heavily on data collection and processing. The data collecting and analysis system's outputs are then used to manage, control, and plan transportation. Vehicular Ad-hoc Networks, which allow moving IVs to be dynamically linked, help intelligent vehicles connect to the internet and analyze and share data in real time. We may use blockchain aspects such distributed ledgers, Merkel trees, Hash functions (SHA-256), and consensus mechanisms (proof of work technique) to create a more secure environment with user identity awareness using the Intelligent Vehicle Biometric Crediting (IV-BC) protocol.

It's a peer-to-peer networking system that provides a secure and reliable environment for vehicle communication, as well as an internal ledger and data access. Vehicular cloud computing (VCC) has had a substantial impact on traffic management by equipping intelligent vehicles with digital resources such as cloud computing, data storage, traffic guiding, and decision-making.

In the framework, following concepts were suggested:

1. Firmware update strategy based on blockchain: Autonomous vehicle manufacturers join a consortium blockchain to ensure high availability and speedy delivery of products and updates at low computational cost that is resistant to DoS attacks. Attribute-Based

Encryption (ABE) creates an access policy that ensures that only approved autonomous vehicles can download and install new updates, while also ensuring the validity and integrity of firmware updates through the use of a smart contract. The approach can be implemented during the contact period of two driving autonomous cars due to the low time required for cryptographic computations and the transfer time.

2. Use of a Zero-Knowledge Proof Module: Zero-Knowledge Proof protocols are a mechanism through which a validator validates a prover's claims by performing some computations without having key information about the prover. A zero-knowledge proof protocol is used in a volatile environment. Each distributor can trade an encrypted version of the update in exchange for proofs of distribution from receiver antivirus software. The smart contract ensures that the decryption key is delivered, which will be exposed after the proofs are collected. The smart contract also improves the distributor's reputation based on the obtained proof.
3. Using Incentive and Reward Systems: By preserving a credit reputation for each distributor account in the blockchain, a payment mechanism is intended to motivate autonomous vehicles to distribute Firmware updates for consortium blockchain.
4. Consider a Blockchain-based service that gives drivers and riders smart contract templates. Initially, the two parties will select a "simple" smart contract template (for example, transferring products or persons; paying the driver in fiat currency or cryptocurrency; payment in cash or through reward points;) The parties will next agree on the specifics of the transaction (for example, the exact fee to be paid; whether or not to carry more persons;) Individuals will no longer require the services of a third party to execute transactions since the Smart Contract template ensures that either both sides of the transaction are met, or none at all.

In another framework proposed by Renu et al. (2021), it was realized that there is a need for third party involvement in a decentralized ride sharing environment, but this involvement has to be restricted to only the initial verification stage of the drivers. Verifying a driver's details is an important measure to protect the riders against malicious behaviors and thus, will require a verifying authority's involvement. Based on an existing architecture, a decentralized P2P system using Blockchain is presented. A decentralized application (DApp) is being constructed as a front-end interface, with the backend being built on the decentralized Ethereum blockchain. Both the user and the driver are registered in the blockchain network with the essential information in this framework. This meta-data information is tied to each of their profiles, and every node in the network may see it. As part of this framework, three user roles were created: the driver, the user, and some legal authorities for verification.

The authors used the local Ethereum framework, MetaMask, Web3js, Nodejs, and MongoDB to create a prototype of the proposed framework. For this DApp, there are two types of stakeholders: the driver and the rider. Each user has multiple roles and duties, which are assigned through the DApp's many dashboards. The user's pick-up and drop-off details, as well as the ride fare, are displayed on the Driver Dashboard. The Driver Dashboard also displays the payment status, whereas the Rider Dashboard displays the user's pick-up and drop-off information, as well as the ride fee. When a user's information is entered, it is stored in MongoDB and the credential's metadata is pushed into the blockchain. The DApp has a front end and a decentralized platform running on the back end.

The authors identified that the ride-matching issue is one of the major issues with the P2P ridesharing system. The ability of any ridesharing system to efficiently direct drivers to passengers is a distinctive feature. Any flexible ridesharing system must be capable of providing the best possible answer to ride-matching issues. Greedy heuristic optimization, meta-heuristic optimization, Exact formulation and heuristic solution, decomposition algorithm, and dynamic programming are some of the methods that provide an optimal solution to ride-matching. The DApp tackles the matching problem by utilizing an algorithm to match riders requesting ridesharing in order to reduce overall journey distance.

Matching is a mathematical notion, specifically in graph theory. In an undirected graph, matching is a set of edges with no shared vertices. An undirected graph is created for this ride-sharing situation, with passengers as nodes and their sharing plan as edges. It is feasible to determine the optimal sharing plan with the shortest total distance using a maximum matching with minimum weight method.

The authors formulated different ride-matching scenarios involving two passengers, A and B. The authors identified five key passenger scenarios:

1. A is picked up first and then B, the A is dropped and then B
2. B is picked up first and then A, the B is dropped and then A
3. A is picked up first and then B, then B is dropped off first and then A
4. B is picked up first and then A, then A is dropped off first and then B
5. A and B travel separately

The algorithm calculates the distance between every two passengers defined in different scenarios using Manhattan's Distance concept.

The main drawback found in the previous two frameworks was:

1. Key rider information is still transparent to the peers. The driver before booking of ride can still know key details of the rider's route and can put this knowledge into malicious use
2. Less safeguard for riders against malicious acts from driver side
3. No safeguard for a driver if a rider does any malicious act

Since the framework is peer-to-peer, the lack of safeguards is a big concern since no governing authority exists in such a system to resolve disputes. The system proposed by M. Baza et al. (2019), which they named B-Ride, provides few measures to prevent malicious behavior from any peer present in the blockchain. The authors suggest a smart-contract-based blockchain-based ridesharing system to address the single point of failure vulnerabilities that plague traditional client-server systems. The authors claim that, in addition to being entirely dispersed and transparent, blockchain's openness creates a possible privacy risk because data can be publicly available. Despite the usage of anonymous authentication, the end users' privacy is not adequately protected.

This primarily necessitates reconciling two opposing goals:

1. The demand for a transparent system while maintaining users' privacy.
2. Ensuring accountability while remaining anonymous.

The major contributions of the paper were:

1. Cloaking Mechanism to hide riders route details: The writers employ cloaking to protect riders' trip privacy, so a rider provides a cloaked pick-up and drop-off location, as well as a pick-up time. Then, using an off-line matching mechanism, interested drivers check if the request falls on his shrouded route, and then deliver the exact trip data encrypted with the riders' public key. Based on some criteria, a rider can then choose the best-matched driver to share a trip with. This works as a distributed auction that is run on the blockchain for transparency.
2. Time-locked deposit protocol: The authors suggest a time-locked deposit protocol for ridesharing services based on zero-knowledge proof protocol to ensure confidence between a rider and a selected driver. The central concept is to design a claim-or-fine approach that works as follows: (i) A rider must post a smart contract with a deposit budget as proof of accepting a driver's offer, as well as a set of obfuscated locations. (ii) As a commitment to his offer, the picked driver should also deposit a budget to the contract. (iii) When the driver arrives at the pick-up site, he or she acts as a (prover) and submits a proof to the blockchain. The driver must show that the pick-up location is within a predetermined range of cells. (iv) Finally, a smart contract works as a (verifier) by checking the proof in a zero knowledge way and then awarding rewards to the driver if the proof is legitimate or fining the driver if the proof is invalid or not sent before the agreed-upon pick-up time.
3. Proof-of-elapsed-distance: The authors present a way for ensuring equitable payment between the driver and the rider without relying on trust. A driver must provide an elapsed distance to the rider at regular intervals, who must validate it by signing it with his private key. The smart-contract then transmits the fare to the driver once the rider presents proof-of-elapsed-distance (i.e., the elapsed distance with the driver's signature on it). The driver is paid as he or she drives in this manner. Meanwhile, if the rider ceases to provide proofs to the blockchain, the journey will be terminated instantly. Furthermore, the blockchain only stores elapsed distances, and no other critical information is exposed to the public.
4. Reputation points: Each driver has two reputation indices in B-Ride: (i) The first score rises every time a driver sends a legitimate proof of arrival to the pick-up location. (ii) As each trip is completed, the second score grows. Each driver will have a trust value in B-Ride based on the two indices, which riders will use to choose good drivers for their journeys. The reputation system provides an economic incentive for drivers to follow the rules.

Vazquez et al. (2021) provide a very similar architecture as B-Ride in their proposal. Their architecture, however, uses a location-based service provider to ensure a driver reached the starting point. The suggested methodology use smart contracts and a cryptography protocol to decentralize the automation of transactions between drivers and passengers.

The process involved in their proposed system is highlighted below:

1. A smart contract is used by the passenger to publish a ride request on the blockchain. Spatial cloaking, a technique for blurring a user's exact location into a spatial region in order to preserve privacy, is used to store the pickup and drop-off information. This cloaking mechanism is similar to the one implemented in the B-Ride Ridesharing system as discussed before.
2. The matching algorithm uses the blockchain to figure out which drivers' destinations correspond to the concealed spatial location.
3. The drivers who have been matched are invited to make a travel offer. To create a fair bid system, the price is posted to all matched drivers. Each driver's route is encrypted with the passenger's public key, ensuring that only the passenger can view the driver's whereabouts.
4. The passenger selects the preferable offer, publishes a ZKP-based smart contract with a deposit fee, and posts a smart contract with a budget to guarantee the journey to foster confidence between the passenger and the driver. As a commitment to the offer, the driver posts a smart contract with a deposit fee.
5. The smart contract verifies the transaction and using ZKP determine the outcome of the contract, i.e., release passenger/driver’s deposit fee in case passenger/driver fails to appear at pickup point or when arrival at pickup point is validated by both parties, passengers initial deposit is released to driver as initial payment.
6. Once the ride begins, a new smart contract delivers partial payments to the driver in 5 to 10-minute intervals as the journey progresses. This is known as proof-of-elapsed-time consensus algorithm.
7. When the ride is finished, the smart contract acts as a validator, ensuring that the journey was completed successfully, before paying the service fee to the driver.

The authors compared three algorithms in order to identify a match between passengers and drivers. A Distance Greedy Heuristic approach, a Time Greedy Heuristic approach, and a Data Structure approach were used. The following limits were set for the three algorithms: (1) vehicle capacity, (2) maximum passenger wait time, and (3) maximum detour for the ride request. The goal of the objective function is to reduce passenger timeout.

The Distance Greedy Heuristic simply seeks the nearest accessible vehicle to the trip origin. The Time Greedy Heuristic finds the vehicle with the least amount of waiting and journey time. The Data Structure method is more involved, since it creates nodes for ride offers before determining the shortest path for the trips. The response time for the time greedy algorithm was faster as the number of rides increased.

The blockchain based ridesharing systems discussed till now involved the rider initially implementing the smart contract. Sowmya Kudva et al. (2020), introduce different systems in which driver smart contracts are implemented that are individually deployed by each driver. They named this system PEBERS. The authors have also justified this by showing that the driver smart contracts proposed in the system use less gas (in simpler words, spend less computational cost), proving that it is a more efficient system than many other existing systems in terms of passenger expenses and driver profitability.

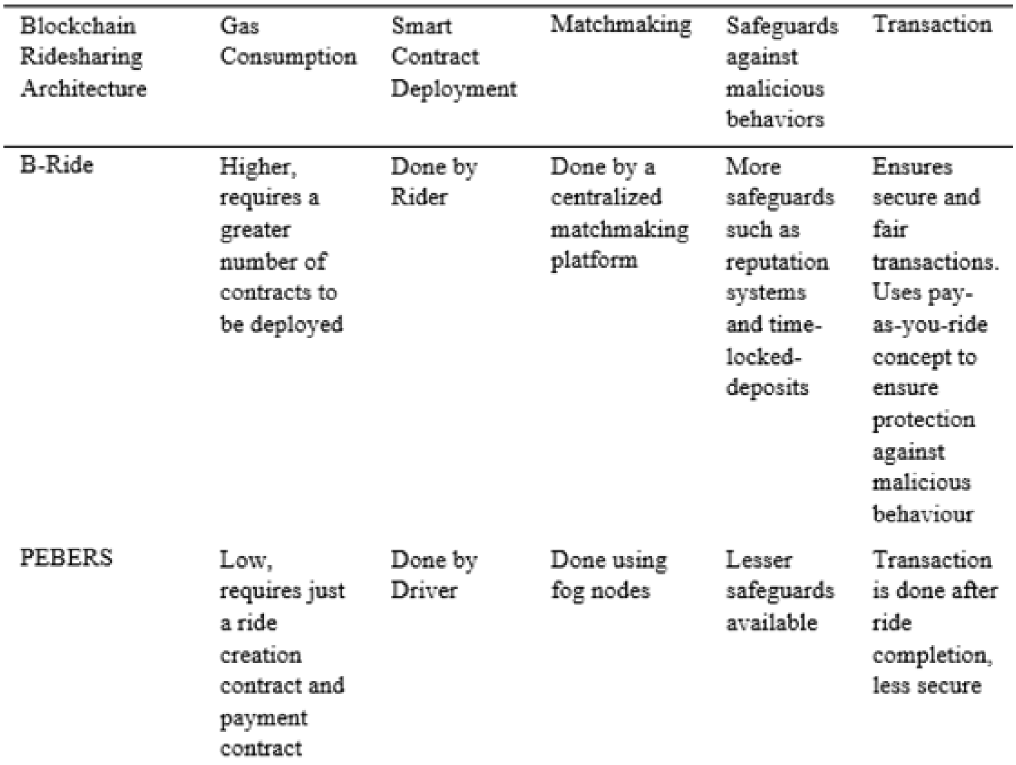
The authors propose a delegated proof of stake consensus mechanism, in which validator nodes are randomly given leader roles rather than miners. This assured that there would be no forking of the blockchain at any point in the future.

Fog computing nodes are also used as authorized nodes in the proposed model. Fog computing nodes are roadside equipment that provides storage, computation, and connectivity. These nodes are semi-trusted and are scattered over the network in terms of area. This functionality will eliminate the need for a centralized server and provide benefits such as location awareness and reduced latency to our architecture. They also act as matchmakers, bringing together passengers and drivers.

In the system, details of driver smart contracts deployment and utilization of fog computing for matchmaking can be summed up below:

1. In the network, a driver is uniquely identifiable by his id. The smart contract is deployed on the network by the registered driver, who first puts one ether in the smart contract storage.
2. The passenger requests a trip from the nearest fog computing unit.
3. When a fog node receives a request, it attempts one-to-many dynamic matching based on the passenger's location and delivers a list of currently available driver details.
4. Interested drivers check the topologically closest fog computing units for passenger transportation requests on a regular basis. Each driver in the passenger's selected region first reviews the passenger's request before sending a response with their price per mile offer. The driver's id, quote, and rep points are all included in the response.
5. If the passenger refuses to accept any of the waiting drivers, the fog node discards the request after a specific amount of time has passed. Otherwise, it sends an Acknowledgement message to the passenger's chosen driver. Driver responds with a positive response to the notification.

Table 1: Comparison between B-Ride and PEBERS



Blockchain Ridesharing Architecture	Gas Consumption	Smart Contract Deployment	Matchmaking	Safeguards against malicious behaviors	Transaction
B-Ride	Higher, requires a greater number of contracts to be deployed	Done by Rider	Done by a centralized matchmaking platform	More safeguards such as reputation systems and time-locked-deposits	Ensures secure and fair transactions. Uses pay-as-you-ride concept to ensure protection against malicious behaviour
PEBERS	Low, requires just a ride creation contract and payment contract	Done by Driver	Done using fog nodes	Lesser safeguards available	Transaction is done after ride completion, less secure

S. Khanji et al. (2019) propose a hybrid blockchain based application - GreenRide-to offer realtime and prompt ridesharing services in cities that suffer from traffic jams and poor public transport infrastructure. The proposed system is semi-decentralized in nature.

The architecture of GreenRide utilizes the decentralization and distribution nature of blockchain to create the GRT to reward users for their carbon emissions reduction. GRT or Green Ride Tokens is a cryptocurrency created for the application by forking the private Ethereum Network.

The GreenRide Token (GRT) is an ERC-20-compliant cryptocurrency. It features an endless supply of tokens; the service provider can mint as many tokens as they like as long as the token is linked to kilogrammes of CO2 saved per ride. GRT smart contracts are written in Solidity and define all abstract functions required to perform GRT balance-related transactions such as minting, transferring, and approving. On the private Ethereum network, these smart contracts are compiled and deployed. The only decentralized element of GreenRide where GRT circulation is managed on a private blockchain network.

GreenRide is a hybrid blockchain application that aims to solve problems with the public blockchain network by allowing only pre-selected nodes to authorize (mine) the next block in the blockchain. As a result, token exchange is instantaneous and scalability is not an issue. However, for security and legal reasons, GreenRide users' data will be stored in a Google Firebase database that can be readily connected with the corporate database, and so employees' data will not be stored outside the corporate premises. Users will no longer be hesitant to provide their route information as a result. The programme, on the other hand, is run on a private network to avoid any regulatory issues in nations where cryptocurrencies are prohibited.

7. Conclusion

Existing ride-sharing platforms, while efficient and popular, may not be better in terms of pricing models, user safety, transaction transparency, and data security. All of these challenges can be addressed with blockchain-based systems, which provide more inventive functionality with greater simplicity of use and control. Riders may communicate directly with drivers using the blockchain's decentralized network, reducing the need for additional fees. People with a smartphone and secure modern automobiles have greater market opportunities because there are no intermediaries. Due to blockchain's potential to establish accountability, passengers may examine how a ride-sharing business operates. Smart contracts empower stakeholders to use blockchain-enabled peer-to-peer leasing of autos for two parties directly involved, based on predetermined criteria.

As a result, accurate pricing is always provided, and the system develops credibility and openness. By establishing an acceptable ranking for riders, the limits ensure that drivers do not engage in any illegal behavior.

References

[1] Riddhi Gupta, Riya Gupta, Sonali Shripad Shanbhag, 2021, A Survey of Peer-to-Peer Ride Sharing Services using Blockchain, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 08 (August 2021)

[2] Silwal, Shrawani & Gani, Md Osman & Raychoudhury, Vaskar. (2019). A Survey of Taxi Ride Sharing System Architectures. 144-149. 10.1109/SMARTCOMP.2019.00044.

[3] Bozdog, Vladimir & Makkes, Marc & Halteren, Aart & Bal, Henri. (2018). RideMatcher: Peer-to-Peer Matching of Passengers for Efficient Ridesharing. 263-272. 10.1109/CCGRID.2018.00041.

[4] Renu, S. A., Banik, B. G. (2021). Implementation of a secure ride-sharing DApp using smart contracts on Ethereum blockchain. International Journal of Safety and Security Engineering, Vol. 11, No. 2, pp. 167-173. https://doi. org/10.18280/ijsse.110205

[5] M. Baza, N. Lasla, M. M. E. A. Mahmoud, G. Srivastava and M. Abdallah, "B-Ride: Ride Sharing With Privacy-Preservation, Trust and Fair Payment Atop Public Blockchain, " in IEEE Transactions on Network Science and Engineering, vol. 8, no. 2, pp. 1214-1229, 1 April-June 2021, doi: 10.1109/TNSE.2019.2959230.

[6] Vazquez, E. and Landa-Silva, D. (2021). Towards Blockchain-based Ride-sharing Systems. In Proceedings of the 10th International Conference on Operations Research and Enterprise Systems-ICORES, ISBN 978-989-758-485-5; ISSN 2184-4372, pages 446-452. DOI: 10.5220/0010323204460452

[7] Kudva, Sowmya & Norderhaug, Renat & Badsha, Shahriar & Sengupta, Shamik & Kayes, A. S. M. . (2020). PEBERS: Practical Ethereum Blockchain based Efficient Ride Hailing Service. 10.1109/ICIoT48696.2020.9089473.

[8] R. Joseph, R. Sah, A. Date, P. Rane and A. Chugh, "BlockWheels-A Peer to Peer Ridesharing Network, " 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 166-171, doi: 10.1109/ICICCS51141.2021.9432188.

[9] S. Khanji and S. Assaf, "Boosting Ridesharing Efficiency Through Blockchain: GreenRide Application Case Study, " 2019 10th International Conference on Information and Communication Systems (ICICS), 2019, pp. 224-229, doi: 10.1109/IACS.2019.8809108.

[10] Metcalfe, William. (2020). Ethereum, Smart Contracts, DApps. 10.1007/978-981-15-3376-1_5.

[11] H. Wang, L. Wang, Z. Zhou, X. Tao, G. Pau, and F. Arena, “Blockchain-Based Resource Allocation Model in Fog Computing, ” Applied Sciences, vol. 9, no. 24. MDPI AG, p. 5538, Dec. 16, 2019. doi: 10.3390/app9245538.

[12] Zikratov, Igor & Kuzmin, Alexander & Akimenko, Vladislav & Niculichev, Viktor & Yalansky, Lucas. (2017). Ensuring data integrity using blockchain technology. 534-539.10.23919/FRUCT.2017.8071359.

[13] Bathen, Luis & Flores, German & Jadav, Divyesh. (2020). RiderS: Towards a Privacy-Aware Decentralized Self-Driving Ride-Sharing Ecosystem. 32-41.10.1109/DAPPS49028.2020.00004.

[14] Chang, Shuchih Ernest & Chang, Chi-Yin. (2018). Application of Blockchain Technology to Smart City Service: A Case of Ridesharing. 664-671.10.1109/Cybermatics_2018.2018.00134.

[15] Vakilinia, Iman & Vakilinia, Shahin & Badsha, Shahriar & Arslan, Engin & Sengupta, Shamik. (2019). Pooling Approach for Task Allocation in the Blockchain Based Decentralized Storage Network. 10.23919/CNSM46954.2019.9012719.

[16] Cai, Wei & Wang, Zehua & Ernst, Jason & Hong, Zhen & Feng, Chen. (2018). Decentralized Applications: The Blockchain-Empowered Software System. IEEE Access.6.53019-53033.10.1109/ACCESS.2018.2870644.

[17] W. Al Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmary, and K. Akkaya, “Towards secure smart parking system using blockchain technology, ” Proc. of 17th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las vegas, USA, 2020.

[18] K. Christidis and M. Devetsikiotis, “Blockchains and smart contracts for the internet of things, ” Ieee Access, vol. 4, pp. 2292-2303, 2016

[19] Buterin, V., 2013. Ethereum white paper. GitHub repository, 1, pp.22-23.

[20] Chow, C. -Y. (2008). Cloaking Algorithms for Location Privacy, pages 93-97. Springer US, Boston, MA.

[21] Madhuram. M, Ashu Kumar, Pandyamanian. M (2019). Cross Platform Development using Flutter. International Journal of Engineering Science and Computing, April 2019

[22] Vujičić, Dejan & Jagodic, Dijana & Ranđić, Siniša. (2018). Blockchain technology, Bitcoin, and Ethereum: A brief overview. 1-6. 10.1109/INFOTEH.2018.8345547.