

## Implementation of a Secure Ride-Sharing DApp Using Smart Contracts on Ethereum Blockchain



Sathya A. Renu\*, Barnali Gupta Banik

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Deemed to be University, Hyderabad, Telangana 500075, India

Corresponding Author Email: [sathya.renu@klh.edu.in](mailto:sathya.renu@klh.edu.in)

<https://doi.org/10.18280/ijssse.110205>

### ABSTRACT

**Received:** 16 March 2021

**Accepted:** 15 April 2021

#### Keywords:

*blockchain, smart city, smart contract, DApp, Ethereum, cryptocurrency, shared economy, greedy algorithm, graph theory*

The idea of sharing economy gives rise to unique ideas and develops innovative businesses. This article aims to relate the smart city concept by introducing the smart transport system and explores the opportunities of adopting blockchain technology in ride-sharing services. Blockchain technology is a distributed, decentralized public ledger that allows peer-to-peer transactions in a secured way without any third party. This paper proposes a blockchain-based framework from the existing centralized framework for a ride-sharing service and implements the same as a decentralized application (DApp) based on smart contracts on Ethereum Blockchain. Using smart contracts facilitate the users with automated transactions, removes the intermediaries, and enables various activities to be carried out safely and securely. Implementation of smart contracts is done using the Solidity programming language. This DApp uses the min matching algorithm to match riders requesting rideshare to save total travel distance. With the overwhelming growth in the usage of cryptocurrencies, smart contracts usage in applications as proposed in this paper can transform the sharing economy.

## 1. INTRODUCTION

### 1.1 Background

For the past few decades, Blockchain is revolutionizing every major industry with distinguishing properties and not just with the financial sectors. Researchers are trying to explore the problems where blockchain can fit in and provide exceptional services. Crowdfunding, Insurance, Healthcare, Governance, Energy are few major domains where blockchain is already progressing. For the past few years sharing economy has seen exponential growth [1, 2]. For example, Ola, Uber, Airbnb, Lyft will witness the same [3]. It enabled users to share their assets with others who do not possess them and require them. Sharing of resources will reduce wastes and promotes a lot of people to participate. However, most transactions of sharing economy applications happen on a centralized platform with a third party. The transaction can be an information exchange or payments. Centralized mode of transactions is susceptible to threats like password hacks, misusing of customer data and has increased cost and efforts in maintaining it. One of the best contributors to sharing economy can be shared transport services i.e. taxi rides. Currently, Uber, Ola, and Lyft are leading their way in ride-sharing systems. Almost all ridesharing applications use a centralized approach. But, having a centralized system in ride-sharing services has many downsides like surge fee, convenience fee, GST, payment gateway charges, etc. Thus there's a need for better approaches. One such solution to overcome the downside of the centralized approach is a decentralized, peer-to-peer approach in the ride-sharing process.

This paper aims to provide a deeper understanding of blockchain technology and give considerations of how to integrate it into their business strategies. The paper humbly claims two-fold contribution:

- i. A blockchain-based ride-sharing architecture is proposed for safe and secure traveling.
- ii. A prototype of the proposed architecture is developed as a Decentralized Application (DApp) on the Ethereum blockchain.

### 1.2 Blockchain

Blockchain is a peer-to-peer, distributed ledger technology that progressed from Bitcoin [4], a cryptocurrency developed by Satoshi Nakamoto in 2008. The overwhelming success of Blockchain is due to its distinct properties like distributed, decentralized, immutable, secured, and shared. Every transaction in the blockchain is recorded and anyone in the network can access and verify the transactions. As it follows a peer-to-peer (P2P) model blockchain eliminates the need for intermediaries. Adapting Blockchain in a smart city can lead to various benefits like security, transparency, trust-free, and automation. A smart city aims in improving the standard of living and creating a sustainable environment through Information and Communication Technology (ICT) [5]. The conjunction of Smart city and Blockchain can improve the performance, efficiency, security, and smartness of the environment. The most commonly used distributed ledger technology is Blockchain. It organizes the data into several blocks interconnected with each other. Blockchain systems are of three different types: public, private, and consortium [6]. A public blockchain is permission-less and anyone being in the

network can see and make transactions, validate them and manage the shared ledger. Bitcoin and Ethereum are examples of the public blockchain. Whereas private blockchain is a decentralized yet centralized network possessed by a single entity. A consortium blockchain is also a private network that handles cross-organizational transactions.

### 1.3 Ethereum

Ethereum is a public and permissionless blockchain introduced by Vitalik Buterin. Ethereum is built with Turing complete programming language and overcomes several limitations of Bitcoin’s scripting language difficulty. Transactions on Ethereum are done by Smart Contracts. Smart contracts are nothing but executing some set of predefined rules when certain conditions are met [7]. Transactions in Ethereum are cryptographically signed instructions. The fee for these transactions is paid as Ether. Ether is the built-in currency of Ethereum. Apart from payment Ether is also used as a pricing instrument for running DApps in the system. Ethereum defines cost for every computation in the form of Gas. These instructions are executed by Ethereum Virtual Machine (EVM) running in every node of the network. Some of the potential domains of Ethereum are Insurance, File Storage, Market Predictions, FinTech, etc. However, one of the significant use-cases of Ethereum is Decentralized applications (DApps) which will be discussed in depth in this paper.

### 1.4 Smart contract

A smart contract based on the Ethereum blockchain is a cryptographic bundle that takes and processes inputs, writes outputs, and stores information. The public can access the output only when some predefined conditions are met [7]. Kevin Delmolino et al., have explained a step-by-step procedure of implementing a smart contract in their Safe Smart Contract paper [8]. The smart contracts generate bytecode at compilation and are stored in blockchain. These byte codes are run by EVM which is running in each node of the network [9]. Solidity is the most common programming language in creating smart contracts. Solidity is derived from C++, JavaScript, and Python. Smart contracts enable the solidity code to run as decentralized applications on a blockchain system. Smart contracts are considered the backbone of decentralized applications [10].

### 1.5 Decentralized Applications (DApp)

DApp is a decentralized application hosted on a P2P blockchain network. Ethereum hosted DApps on different domains like Insurance, Energy, Finance, Health, and many more. But, many of the DApps are only partially decentralized. As per [11], the properties of DApp are characterized as,

- Open Source: DApp codes are open source and audit from a third party is possible.
- No single-point Failure: Due to its decentralized nature no central point of failure.
- Consensus: To enable transparency consensus among the nodes is inevitable.
- Currency Support: Internal cryptocurrency is the major factor that runs the ecosystem of any decentralized application.

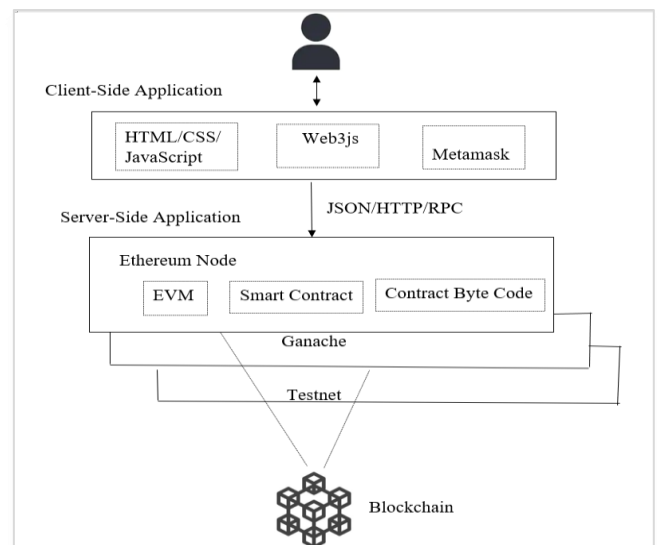
Key differences between centralized web applications and decentralized blockchain applications are shown in Table 1 below.

**Table 1.** Web App Vs DApp

Traditional Web Applications	Decentralized Applications
Follows client-server architecture	The client interacts through smart contracts.
Databased connected to the backend server and is accessed by the client	Backend code runs on a decentralized P2P network.
Needs central authority to set roles and permissions	No central authority
Security and Privacy are the concerns	Secure, immutable, and autonomous.

#### 1.5.1 DApp architecture

DApp is two-tier architecture, one tier being the front-end client-side application and the other being the back-end server-side tier where the smart contract is deployed in the blockchain network. Figure 1. below shows the general architecture of the decentralized application and the interaction between a client and a server-side application. The tools that are used in developing the ridesharing Dapp are discussed in the later sections.



**Figure 1.** DApp architecture

As mentioned above any DApp has a backend and a front-end application. Unlike the centralized applications where the backend code runs on centralized servers, a DApp’s backend code is distributed through a decentralized peer-to-peer network. The front-end code of a DApp can be written in any language and API calls can be made to the backend.

Ethereum was the first blockchain-based platform to create a Turing complete language for writing smart contracts and a DApp development platform. The standard language for building DApp on the Ethereum platform is solidity. In this project, the Ethereum Truffle suite was used to develop and deploy smart contracts.

Truffle suite The Truffle suite comprises several components that are needed for developing a Dapp. It facilitates to development, compilation, deploy and test the smart contracts. It also allows the creation of front-end user interfaces for

Dapps. The three major components of the Truffle suite are 1. Truffle framework, the developmental framework where the smart contracts are created, deployed, and tested. 2. Ganache, the simulator where the Dapp's Graphical User Interface (GUI) is developed and tested. 3. Drizzle: the front end library for Dapp development.

Web3.js The client-side application is just like any other web application built using HTML, CSS, and Javascript. Web3.js is a set of libraries that enables the Dapp browser to interact with the blockchain. It also helps in reading and writing data from smart contracts, transacting ethers from one account to another, and so on. We have used Metamask in our project that includes its web3 provider in the browser. Metamask is a chrome browser extension that allows users to manage their private key and Ethereum accounts in a secured manner. It uses the Ethereum accounts to interact with websites that use web3.js.

Ethereum Virtual Machine (EVM) In an Ethereum network, every Ethereum node runs its EVM and is involved in running the same instructions of smart contracts over the entire Ethereum network.

## 2. EXISTING CENTRALISED MODEL OF TAXI SHARING PLATFORMS

### A. Pricing

Let's take a look at Uber and Ola where they offer transactions between the driver and the user. They help the customers to make a safe payment and flexibility in finding the cabs and other amenities while traveling. This creates a dependency between the customer and the driver and assists the intermediaries to increase their prices. According to [12], these intermediaries charge 10-20 percent higher than the actual charges to the customers.

### B. Privacy and Security

As per [4], Uber even uses algorithmic predictions to find how much the customers are willing to pay more than the services they receive which is highly a security breach. Also, having all customer details in centralized servers can invite hackers for numerous attacks.

### C. Lack of Transparency

Most of these ridesharing companies adapt surging price concepts where based on the demand they may charge the customers high especially in metro cities. The majority of the customers do not understand how these operations take place. There's no transparency between the customers and the organizations.

### D. Safety standards

Nowadays, the most important issue faced by people who travel through Uber, Ola, or any riding platforms is safety. The safety of both driver and the customers are at stake. Despite spending a huge amount in background verification processes this problem persists. The working process of the centralized ride-sharing model is shown below in Figure 2.

The process model used in the centralized storage model is shown in Figure 2. The steps involved in the process are as follows.

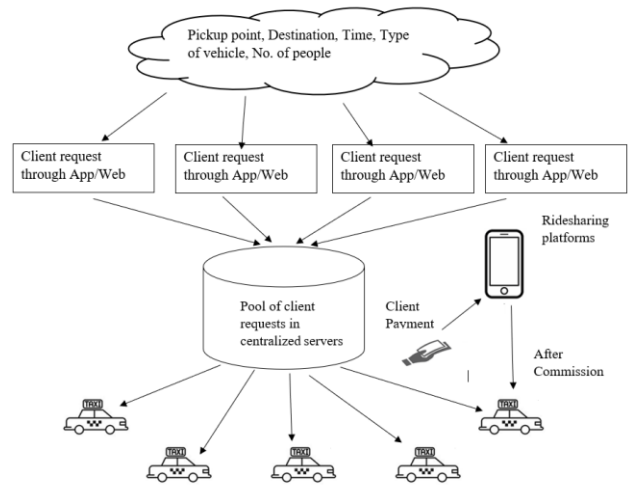


Figure 2. Existing ride-sharing framework

1. The user looks for the cab through a web or mobile app of the cab providing platforms.
2. The user enters the details like pick-up location, drop point, time, no. of persons, type of car, and mode of payment.
3. The request is queued in a transaction pool and is routed to the nearby driver.
4. The driver's and user's information are exchanged with each other.
5. Once the trip is completed, the user will pay through the app and the cab provider receives and will send the driver's charge after reducing a considerable amount of 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.

## 3. DECENTRALISED BLOCKCHAIN SOLUTION

To eliminate the issues mentioned in section 2, of the current centralized model. A decentralized approach is suggested. Blockchain as we know is a peer-to-peer decentralized, public ledger that allows transactions to happen without any intermediaries. And any transaction on a blockchain network is transparent, immutable, and secured. Therefore, adopting blockchain in ridesharing services removes the middle man and makes the transaction between the drivers and users at a reasonable cost. This attracts more users onboard and benefits them financially. S. E. Chang et al., in his work Application of Blockchain Technology to Smart City Service, mentioned that adopting Blockchain Technology in ridesharing services will make us closer to the concept of Smart City [13] as it embeds the technology within the city and adds different social dimensions for efficient allocation of resources, providing high-value services and improving the quality of life for the public. Some of the benefits provided by blockchain technology compared to the centralized model are listed below [14].

- A. *Less Cost:* Apparently, blockchain removes the intermediated for the information exchange and thereby slashing the costs involved in the transaction to a greater

extent. Moreover, it enables the user to trust the data on blockchain than an unknown third-party.

- B. *Transparency*: Although Blockchain is anonymous it provides transparency of the transactions at the same time. Therefore, it is always possible to trace back to the information if something goes wrong.
- C. *Safety*: Authenticity of data is assured as all the data blockchain are cryptographically protected. It is impossible to have any fraudulent data over the network.
- D. *Transparency*: In the blockchain, the data is available over the network and the raiding parties can have a transparent view of each other.
- E. *Safe Payments*: Users will be able to make a safe payment directly to the drivers using smart contracts, another unique feature of blockchain.

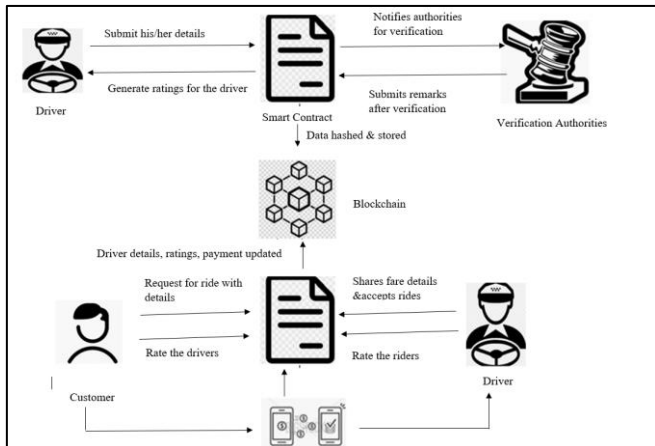
#### 4. PROPOSED RIDESHARING FRAMEWORK

A decentralized P2P framework using Blockchain is proposed based on an existing framework. A decentralized application (DApp) is developed which will act as a front-end interface and is built on a decentralized Ethereum blockchain at the backend. In this framework, both the user and the driver get registered in the blockchain network with the necessary details. This meta-data information is attached to each of their profile which would be viewed by every node of the network. Three user roles were created as a part of this framework one being the driver, the other the user, and some legal authorities for verification. The working process is as given below [15].

1. The driver creates his profile in the network by submitting his driving license and other necessary documents. MongoDB is used as a storage mechanism and the hashed value is stored in the blockchain.

2. The Legal authorities who are also part of the network will be notified once the driver gives the information. Background verification will be done and the remarks would be added to the driver's profile. A smart contract is used to generate the review or ratings of the driver based on which the rider can decide to take a ride with the driver.

3. Similarly, users who want to take a ride should also get registered in the network by providing some essential information like name, phone, etc., and background verification will happen similarly. This assures the safety of drivers.



**Figure 3.** Proposed blockchain-based framework for ridesharing

4. After the verification is done, the user can request cabs by providing his/her location details and other trip details. The driver on such requests calculates the cost of the ride using various parameters like the distance to be traveled, type of car, and a fixed price per kilometer and shares to the users' request.

5. The rider can now choose the ride as per their comfort.

6. Once the ride is completed payment can be automated from one wallet to the other as crypto payments.

Figure 3 summarizes the decentralized P2P ridesharing model.

#### 5. IMPLEMENTATION

##### 5.1 Environment set-up

###### A. Tools and Libraries

Several tools and libraries are available for DApp development. However, the tools and libraries used in developing this ride-sharing DApp are briefly discussed below in Table 2.

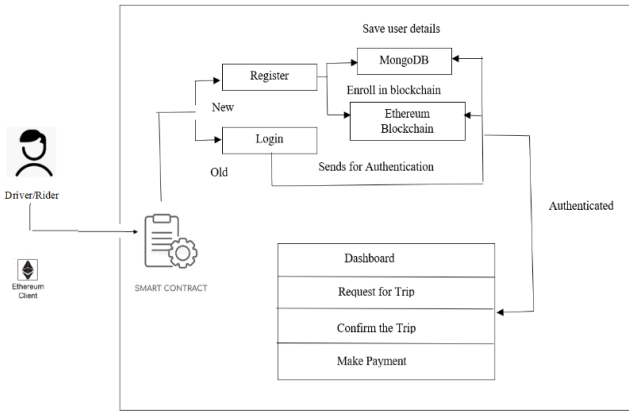
**Table 2.** Tools and libraries used in ridesharing DApp

Tools and Libraries	Descriptions
Truffle Framework	Allows creating DApps on Ethereum network. Provides a suite of tools to write Smart Contracts using Solidity. Provides a framework for testing the contracts. Gives tools to deploy Smart Contracts in the blockchain. Used to develop client-side applications inside truffle.
Ganache	Runs a local instance of Ethereum. Provides sample accounts that can be used for the development and testing process.
Solidity	Object-oriented programming knowledge to write and implement Smart Contracts in Ethereum blockchain.
Web3js	Open Source JavaScript library. Provides API to interact with local Ethereum nodes.
NodeJs Server/npm MetaMask	Backend server for DApp development. Chrome browser extension that acts as Web3 wallet. Inserts Web3js libraries into the browser so that reading writing can be done on Ethereum blockchain
MongoDB	Core data is saved in blockchain however, to facilitate the user details and displaying dashboard details we use MongoDB.
Web Front End	HTML/CSS/JavaScript which highly uses Web3Js to access the blockchain nodes.

###### B. Process Flow of Ridesharing DApp

To develop the prototype of the proposed framework we make use of the local Ethereum framework, MetaMask, Web3js, Nodejs, and MongoDB. Two kinds of stakeholders are identified for this DApp i.e. the driver and the rider. Each user has different roles and responsibilities that are offered using different dashboards of the DApp. Driver Dashboard shows, i) The pick-up and Drop details of the user ii) The ride

fare and iii) Payment Status whereas, the Rider Dashboard shows, i) Pick up and Drop details and ii) Ride Fare. Whenever a user detail is entered, the details are stored in MongoDB and the metadata of the credential is pushed into the blockchain. The DApp contains a front end which at the back end runs a decentralized platform. The workflow of the DApp is shown in Figure 4.



**Figure 4.** The ridesharing DApp workflow

### C. Ride Matching Problems

One of the major problems of the P2P ridesharing system is the ride-matching problem. Any ridesharing system's ability to direct drivers to passengers in the most efficient way is a distinguishing attribute. Any flexible ridesharing system must be prepared to provide an optimal solution to ride-matching problems. Some of the algorithms which give an optimal solution to ride-matching are greedy heuristic optimization, meta-heuristic optimization, Exact formulation and heuristic solution, decomposition algorithm, and dynamic programming [16]. The DApp implemented solves the matching problem by using a min matching algorithm to match riders requesting rideshare to save total travel distance.

### 5.2 Matching algorithm

Matching is a concept in mathematics, graph theory in specific. Matching is a group of edges without any common vertices in an undirected graph. For this ride-sharing problem, an undirected graph is formed with passengers being the nodes and their sharing plan being the edges [17]. With a maximum matching with minimum weight algorithm, it is possible to find the best sharing plan with minimal total distance. The algorithm calculates the distance between every two passengers is defined in different scenarios. The Manhattan distance between two passengers is calculated as,

$$(A(a_1, a_2), B(b_1, b_2), d(A, B) = |a_1 - b_1| + |a_2 - b_2| \quad (1)$$

The five different scenarios in which the distance between each two passengers  $d(A, B)$  is defined are:

- i. Pick up A and then pick-up B then drop A and then drop off B
- ii.

$$d(AB) = d(1) = d(A1B1) + d(B1A2) + d(A2B2) \quad (2)$$

iii. Pick up A then Pickup B then drop B and then drop A

$$d(AB) = d(2) = d(A1B1) + d(B1B2) + d(B2A2) \quad (3)$$

iv. Pick up B and then pick up A then drop A and then B

$$d(AB) = d(3) = d(B1A1) + d(A1A2) + d(A2B2) \quad (4)$$

v. Pick up B then pick up A then drop B and then drop off A

$$d(AB) = d(4) = d(B1A1) + d(A1B2) + d(B2A2) \quad (5)$$

vi. A and B travel on their own

$$d(AB) = d(5) = d(A1A2) + d(B1B2) \quad (6)$$

To assure the quality of sharing vehicles, the algorithm does not allow people to share vehicles if their total distance exceeds 25% more than their travel. Repeat the process below for all the passengers and find the distance.

For passenger(A)

if  $d(A1B1) + d(B1A2) > 1.25d(B1A1)$ :  
set  $d(1)$  to  $+\infty$

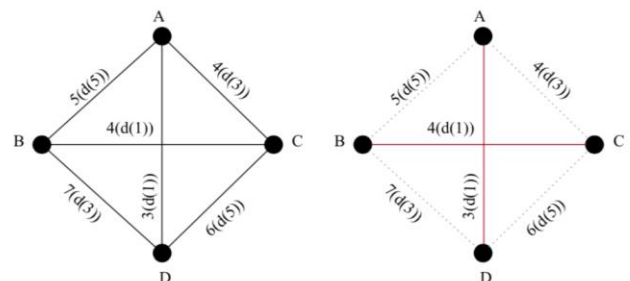
if  $d(A1B1) + d(B1B2) + d(B2A2) > 1.25d(B1A1)$ :  
set  $d(2)$  to  $+\infty$

if  $d(A1B2) + d(B2A2) > 1.25d(B1A1)$ :  
set  $d(4)$  to  $+\infty$

The weight of any two passengers can be:  $d(AB) = \min\{d(1), d(2), d(3), d(4), d(5)\}$ .

### 5.3 Minimum matching

A complete graph is formulated to denote passengers and their suitable way of pooling each other once the weights of the edges are defined. The pseudocode of the minimum matching algorithm is given in Table 3. Using one of the scenarios (1) mentioned above, the matching of passengers for ridesharing is shown below. A, B, C, and D are the four passengers and are assigned some weights. The edges are marked with the way of pooling [18]. For all passengers with minimum total weight, the maximum match is pair A with D utilizing scenario 1, and pair B and C using scenario 1. The digramatic representation of rider matching problem based on a greedy algorithm is given in Figure 5.



**Figure 5.** Riders matching problems using Greedy algorithm

**Table 3.** Minimum Matching Algorithm used to match rides with minimum distance

```

Input: a stream of data b
Output: distance between the passengers
  Get_dist(b)
  Convert stream to data object
  Get Data frame from data object
  Convert Passenger count to numeric type
  Filter the data for passenger less then or equal to
2
  Get distance from the filtered data
  return distance
Input: distance between the passengers
Output: minimum pooling for data points
  min_matching_graph(dist)
  create networkx graph object Get
  Get Shape from dist object
  for all i in list (n1,n2...ni) do
  Add node to graph G
  update Position map with distance
  for all i in list (n1,n2...ni) do
  for all j in list (i1,i2.....ij) do
  
```

```

Add edge for graph
Get label from graph object Get
Print labels
return nodes and label
  
```

**D. Experimental results**

The developed decentralized application is tested by running different experiments over a custom network where the Ganache was running. However, it can be tested using Rinkeby (Ethereum’s) test network as well. The cost for creating the contracts and calling the contracts is given in Table 4 below.

Several ridesharing systems are available having their own merits and demerits. A comparison of the existing centralized model with blockchain based model is tabulated in Table 5.

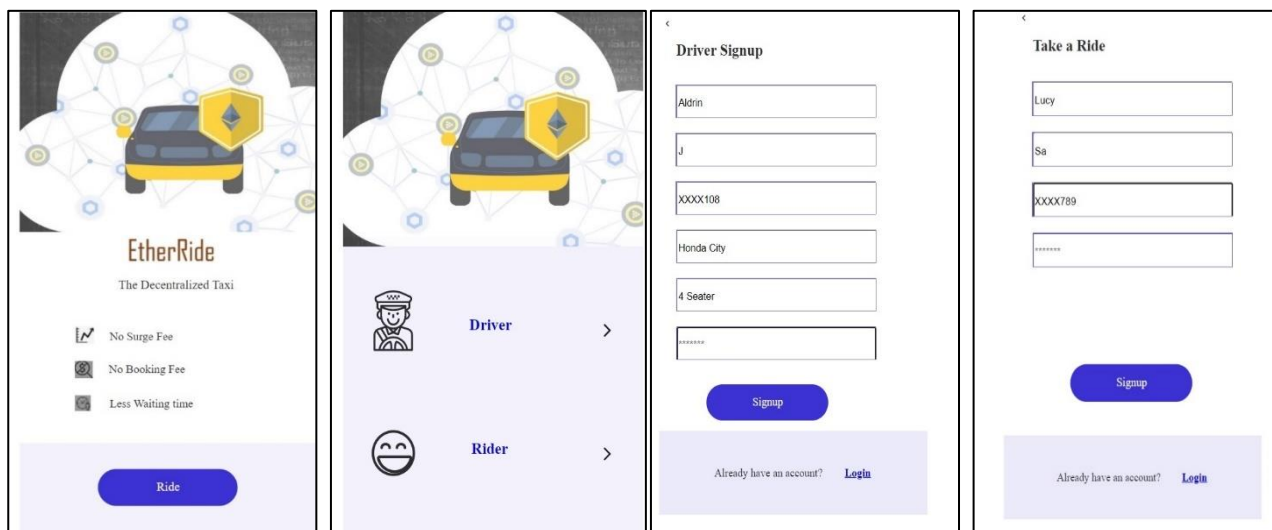
**Table 4.** Smart contract deployment cost

Contract Name	Execution Cost (Gas)	Transaction Cost (Gas)
Rider Contract	1014987	1387779
Driver Contract	1680974	2269274
Authentication	671302	939570

**Table 5.** Comparison of blockchain-based ride-sharing and other ride-sharing systems

System	Architecture	Privacy	Trust	Payment	Transparency
Existing Ride sharing systems	Centralized	Not guaranteed	Partial	Surge Fee	No transparency
SRide[19]	Centralized	Guaranteed	Not guaranteed	Not Fair	No transparency
Arcade City [20]	Blockchain	No privacy	No trust	Not fair	Transparent
EtherRide	Blockchain	No privacy	Trustable	Fair	Transparent

**E. Output**



**6. CONCLUSION AND FUTURE WORK**

The main goal of this paper is to look at the revolutionary technology Blockchain and its use in the shared economy, which can serve as a framework for the smart city idea. This article presents an existing framework for decentralized, P2P, blockchain-based ridesharing services and proposing a more improved version for the same. Further, to support this ride-sharing framework, a decentralized application (DApp) is developed. It will act as a front-end user interface assisted by blockchain. Ethereum, a permissionless public blockchain is used in this DApp and the transactions and information

exchange over the network is automated using smart contracts.

To summarize, blockchain can be used to create a system in which smart contracts incorporated in digital code are maintained in decentralized and transparent databases. The data in these databases are considered to be mutable. Every process and task is expected to have a digital record that will be identified, validated using a digital signature. We can have an ecosystem where no more intermediaries are needed.

Indeed, blockchain leads to a transformation of business models and governance but practically it is still many years away. Blockchain is not a disruptive technology intended to eradicate the traditional business models by providing low-

cost solutions. Rather, it can be seen as a foundational technology capable of laying new frameworks for economic and social issues. Blockchain isn't a band-aid solution to a common technological problem. Although it can help with the transition, a clear plan based on proof of concepts for opportunities must be established. However, while blockchain will have a huge effect, it will take decades for it to pervade our socioeconomic infrastructure. As waves of technical and structural change make headway, acceptance will be incremental and steady, rather than abrupt.

Although blockchain could be used on its own, it would be more likely to have a greater impact while integrated with other technologies like the Internet of Things, Artificial Intelligence, and Big data. This could lead to better solutions for location-based automotive services. In the future we intend to extend the work by i) analyze the cost and performance of the application developed. ii) to explore the technology from a data processing perspective as blockchain technology is a trust-free system and enables users to trust data. Although the overall data quality is improved in blockchain technology it is essential to know the data processing abilities while integrating the blockchain with a larger software system. iii) Analyse the data processing workloads on different types of blockchain.

## REFERENCES

- [1] Chang, S.E., Chang, C. (2018). Application of blockchain technology to smart city service: A case of ridesharing. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, pp. 664-671. [https://doi.org/10.1109/Cybermatics\\_2018.2018.00134](https://doi.org/10.1109/Cybermatics_2018.2018.00134)
- [2] Ramachandran, G.S., Radhakrishnan, R., Krishnamachari, B. (2018). Towards a decentralized data marketplace for smart cities. 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, pp. 1-8. <https://doi.org/10.1109/ISC2.2018.8656952>
- [3] Marchi, A., Parekh, E.J. (2016). How the sharing economy can make its case. *McKinsey Quarterly*, 1: 112.
- [4] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Available: <http://www.ildgroup.si/uploads/product/20/bitcoin.pdf>.
- [5] Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., David, B. (2015). A literature survey on smart cities. *Science China Information Sciences*, 58(10): 1-18. <https://doi.org/10.1007/s11432-015-5397-4>
- [6] Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *Proc. IEEE BigDataCongress'17*, Honolulu, HI, USA, pp. 557-564. <https://doi.org/10.1109/BigDataCongress.2017.85>
- [7] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *Ethereum white paper*.
- [8] Delmolino, K., Arnett, M., Kosba, A., Miller, A., Shi, E. (2015). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In: Clark J., Meiklejohn S., Ryan P., Wallach D., Brenner M., Rohloff K. (eds) *Financial Cryptography and Data Security*. FC 2016. *Lecture Notes in Computer Science*, vol 9604. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-662-53357-4\\_6](https://doi.org/10.1007/978-3-662-53357-4_6)
- [9] Wood, G. (2018). *Ethereum: A secure decentralized generalized transaction ledger*, Byzantium version. available at: <https://ethereum.github.io/yellowpaper/paper.pdf>.
- [10] Ibba, S., Pinna, A., Seu, M., Pani, F.E. (2017). Citysense: Blockchain-oriented smart cities. In *Proceedings of the XP2017 Scientific Workshops*, XP '17, pp. 12:1-12:5. <https://doi.org/10.1145/3120459.3120472>
- [11] Raval, S. (2016). *Decentralized applications: Harnessing Bitcoin's blockchain technology*. O'Reilly Media, 1 ed.
- [12] Huckle, S., Bhattacharya, R., White, M., Beloff, N. (2016). Internet of things, blockchain and shared economy applications. *Procedia Computer Science*, 89: 461-466. <https://doi.org/10.1016/j.procs.2016.09.074>
- [13] Chang, S.E., Chang, C. (2018). Application of blockchain technology to smart city service: A case of ridesharing. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, pp. 664-671. [https://doi.org/10.1109/Cybermatics\\_2018.2018.00134](https://doi.org/10.1109/Cybermatics_2018.2018.00134)
- [14] Yuan, Y., Wang, F. (2018). Towards blockchain-based intelligent transportation systems. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, pp. 2663-2668. <https://doi.org/10.1109/ITSC.2016.7795984>
- [15] Victor, F., Zickau, S. (2018). Geofences on the blockchain: Enabling decentralized location-based services. 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, Singapore, pp. 97-104. <https://doi.org/10.1109/ICDMW.2018.00021>
- [16] Schrieck, M., Safetli, H., Siddiqui, S., Pflügler, C., Wiesche, M., Krcmar, H. (2016). A matching algorithm for dynamic ridesharing. *Transportation Research Procedia*, 19: 272-285. <https://doi.org/10.1016/j.trpro.2016.12.087>
- [17] Baza, M., Lasla, N., Mahmoud, M., Srivastava, G., Abdallah, M. (2019). B-ride: Ride sharing with privacy-preservation, trust and fair payment atop public blockchain. *IEEE Transactions on Network Science and Engineering*. <https://doi.org/10.1109/TNSE.2019.2959230>
- [18] Li, M., Zhu, L., Lin, X. (2019). Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing. *IEEE Internet of Things Journal*, 6(3): 4573-4584. <https://doi.org/10.1109/JIOT.2018.2868076>
- [19] Aïvodji, U.M., Huguenin, K., Huguet, M.J., Killijian, M.O. (2018). Sride: A privacy-preserving ridesharing system. *Proc. of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pp. 40-50. <https://doi.org/10.1145/3212480.3212483>
- [20] Arcade city. [Online]. Available: <https://arcade.city/>, accessed on 20 July 2020.