Intelligent Systems with Applications 16 (2022) 200135

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

Enhancing Blockchain-based Ride-Sharing Services using IPFS

Nesma Mahmoud [*], Asmaa Aly, Hatem Abdelkader

Information Systems Department, Faculty of Computers and Information, Menoufia University, Egypt

ARTICLE INFO

Keywords:
Ride-sharing
Blockchain
Smart contracts
Interplanetary File System (IPFS)

ABSTRACT

Ride-sharing services (RSSs) using centralization methods experience various challenges like single point-of-failure, privacy violation, lack of security, and distributed denial of services (DDoS) attack, etc. So, blockchain-based RSSs mitigate such problems through decentralization. Relying on the blockchain only leads to problems such as increase in application response time, chain size, and a high computational cost due to the increase in data storage in blockchain and thus increase the service costs to end users. Additionally, the blockchain lacks to scalability of data because of the inability to store large-sized data and accommodate the grows of ride-sharing data. To overcome these problems, a novel decentralized ride-sharing system that exploits blockchain and Interplanetary File System (IPFS) is proposed. The goal of the proposed system is to move all ride-sharing data outside the blockchain and replacing it with a small hash. The blockchain manages the application state and users. In addition, it automates processes through smart contracts. While the IPFS stores data for blockchain in immutable and integral way. Wherefore, the proposed ride-sharing system integrates IPFS with blockchain for RSSs to retain the provided assurance by the blockchain and provide efficient service to end users. Experimental results proved the applicability and efficiency of RSS based on blockchain and IPFS which provides efficient storage of ride-sharing data, immutable history, and generally better efficiency in a decentralized manner.

1. Introduction

Ride-sharing is one of the intelligent transportations systems applications (Yuan & Wang, 2016; Mollah et al., 2021). It achieves a huge popularity in major cities due to its several advantages such as congestion reduction, maintain the environment by reducing carbon dioxide emissions and saving time of users. Ride-sharing also represents a prominent example of sharing economy (Baza et al., 2019; Khanji & Assaf, 2019). This economy promotes economic sharing activities in a peer-to-peer (p2p) manner (Yuan & Wang, 2018). Recently, ride-sharing services (RSSs), also known as carpooling or ride-hailing, became an alternative transportation service that allow people to use personal cars more wisely. They enable drivers or people owing private cars to share their vacant seats with other riders. RSSs provide many great benefits to the individual and the community. Examples of such benefits include increasing rates of occupancy, travel costs sharing, and reducing fuel consumption, carbon emissions and air pollution (Sánchez et al., 2016; Baza et al., 2019). Across the world, many providers offer online RSSs such as Careem, Uber, Lyft Line, and Blablacar. According to (Ride sharing market, 2021), the ride-sharing market was valued at USD 182.12 billion in 2018 and is expected to reach USD 212.60 billion by 2026.

An RSS seeks to match drivers using their ride offers i.e., planned trips, with suitable riders using their ride requests, i.e., desired trips. It requires that users —drivers and passengers—to share with the service provider their trip information, including pickup time, location, and destination. This service provider represents a middleman which facilitates the communication between users and charges a commission for each successful shared ride. However, running this service by a central authority and centralized platform, makes the system vulnerable to many problems including single point-of-failure, less transparency, inflexibility, dictation of policies and service conditions. Moreover, central server maintenance and management are costly and highly vulnerable to distributed denial of service attacks (DDoS) (Yuan & Wang, 2016; Baza et al., 2019; Kudva et al., 2020). In addition, the security of the service provider could be exposed to hacking. Consequently, the service can be interrupted, and the data can be disclosed, altered, or even deleted. For example, Uber has witnessed a huge data leakage of 57 million users in late 2016. It has paid 148 million dollars just to settle an investigation to its data breach (Uber China data breach,

[*] Corresponding author.
E-mail address: nesma.1115@ci.menofia.edu.eg (N. Mahmoud).

https://doi.org/10.1016/j.iswa.2022.200135

Available online 9 October 2022

2018).

An effective solution to overcome the problems of the central authority is a decentralized p2p approach for the ride-sharing process (Sánchez et al., 2016). Therefore, some research has exploited the blockchain technology to mitigate the issues of existing client-server architecture of RSS systems (Baza et al., 2019; Kudva et al., 2020; Baza et al., 2020; Vazquez & Landa-silva, 2021; Renu & Banik, 2021). Blockchain is a ledger with the characteristics of immutability, distribution, and traceability. One of the blockchain advantages is to allow mistrusting entities to transact with each other without relying on a central third-party. Instead of making the deal through one trusted centralized authority, blockchain has added many features to ride-sharing systems such as permitting users to directly connect with drivers who is willing to transport them and thus facilitating a cooperative management between them (Yuan & Wang, 2016; Baza et al., 2019; Kudva et al., 2020). This eliminates middlemen which perform any gatekeeping role. Transactional details are maintained in a distributed, transparent data structure which can be accessed by all the nodes and is managed by a network of computers called miners that run a p2p protocol. However, despite these advantages of blockchain-based RSSs, these systems suffer from some deficiencies such as the inability to store large-sized data in blockchain (Ye & Park, 2021). Thus, the blockchain lacks to scalability of data. Additionally, as the data storage in the blockchain grows, more processing is required by miners to validate these data, which adds more time and costs to end users. In addition, the reliance on only the blockchain causes that the chain becomes too large for many users to easily verify and store it with the increase of ride-sharing data.

Motivated by the above deficiencies of blockchain-based RSS and by (Baza et al., 2019) and (Baza et al., 2020), this paper proposes a ride-sharing system that integrates the blockchain with the Interplanetary File System (IPFS) to overcome the aforementioned deficiencies of using the blockchain only. So, the proposed ride-sharing system inherits the features of IPFS to reduce the size of ride-sharing data stored in the blockchain. The IPFS can provide a distributed and permanent storage for ride-sharing data without any storage limitations (Benet, 2014). The main contribution of this paper is to demonstrate how IPFS can be integrated with the blockchain to overcome its storage limitation and enable faster ride-sharing data processing. In the proposed system, all ride-sharing data are moved off-chain to the IPFS. Whereas the ride-sharing data in the blockchain is replaced with hashes which identify data or a file in the IPFS. These data refer to driver's offers and rider's request which includes trip-related information, including pickup time, location, and destination. A proposed methodology along with a smart contract development are also presented which provide a complete scenario of how to integrate IPFS and blockchain in RSS. By moving the ride-sharing data to the IPFS and storing their hash in the transaction instead, the burden of storing these data in the blockchain will be reduced. As a result of data storage reduction in the blockchain, the computation is reduced, which speed up the processing time as well lowering costs on end users. IPFS identifies files by their hash, and thus the inclusion of the ride-sharing data hash in the blockchain transaction allow for efficient and guaranteed retrieval of these data from IPFS correctly and without changes. The contributions of this paper can be summed up as follows:

- A decentralized ride-sharing system is proposed and a prototype of it is implemented as a decentralized application (DApp) on the top of Ethereum, a real-world public blockchain platform, to allow users to access the system in a decentralized manner.
- Following that, this paper proposed the integration of IPFS with the blockchain in the proposed system. For that, a proposed methodology is presented which provide a complete scenario of how to integrate IPFS and blockchain in RSS.
- In the proposed system, we build and deploy a smart contract, and we investigate the expenses incurred by employing such design for both drivers and riders.
- This smart contract is evaluated by deploying it to a local network Ganache.
- Lastly, extensive evaluation of the proposed system has been performed. Our experimental results demonstrate the scalability, verifiability, and superior performance of the proposed ride-sharing system.

The rest of this paper is organized as follows. The necessary background for this paper is covered in Section 2. Section 3 discusses the previous related research works. The proposed ride-sharing system architecture is detailed in Section 4. We present the proposed methodology of our proposed system in Section 5. While Section 6 presents tools, libraries, and languages exploited in experiments and in developing the proposed ride-sharing system. Finally, experimental results and their evaluations are presented in Section 7 followed by concluding remarks in Section 8.

2. Background

In this section, we provide the necessary background on intelligent transportation systems, ride-sharing, blockchain technology, smart contracts, IPFS, and DApps that are exploited in building the proposed system.

2.1. Intelligent Transportation Systems and Ride-Sharing

Intelligent transportation systems (ITSs) are the future of transportation (Mollah et al., 2021; Çaldağ & Gökalp, 2020). They have emerged in the last two decades to improve transportation systems performance, to enhance travel security and mobility, as well as to reduce harmful effects of traffic such as road accidents and air pollution, etc., (Çaldağ & Gökalp, 2020; Zhang et al., 2011). They are considered to be part of internet of things (IoT) and are a step towards smart cities and sharing economy (Yuan & Wang, 2016; Mollah et al., 2021; Priya et al., 2021; Maskey et al., 2020). According to (Çaldağ & Gökalp, 2020) and (Maskey et al., 2020), ITS is defined as the implementation of information technologies and communications in the transportation systems. It integrates vehicles, people, and roads by utilizing advanced information and communication technologies. The "intelligence", in intelligent TS, refers the process of transforming the generated data from ITSs into meaningful information that is useful for individuals and the economy (Zichichi et al., 2020). Smart cities, also known as intelligent environments, exploit ITS to achieve their goal. They are defined as intelligent environment that embeds information and communication technologies to create interactive systems. ITS helps smart cities to provide comprehensive optimization of the urban mobility. Additionally, it eases traffic flow in these cities by reducing travel time, bringing greater safety to drivers, and comfort and entertainment to passengers. applications and services provided by the ITS address and solve transportation problems of smart cities (Mollah et al., 2021). The focus of this paper is on ride-sharing applications due to their significant in assisting ITS to achieve its goal.

Ride-sharing represents a decentralized decision-making model because users are often self-interested and only motivated to team up with each other based on individual objectives (Chau et al., 2020). Improving ride-sharing has a great impact in improving ITS which helps to mitigate and overcome its long-standing problems. These problems include traffic congestion, road accidents, delay, high operation costs, low efficiency, and security risks of data storage in traditional centralized systems (Mollah et al., 2021; Baza et al., 2019; Çaldağ & Gökalp, 2020; Chau et al., 2020). Ride-sharing is also known as carpooling (Chau et al., 2020; Khanji & Assaf, 2019; Wang & Zhang, 2021; Vazquez & Landa-silva, 2021). Additionally, although the terms “ride-hailing services "(RHSs) and "ride-sharing services" (RSSs) are sometimes used interchangeable terminologies, they are difference between them (Aïvodji et al., 2018; Shivers et al., 2019; Shahbazi & Byun, 2022). The "ride-hailing" term refers to companies such as Uber and Careem. It enables riders to request a specific ride from their current location to a specified destination. While the term "ride-sharing" describes situations where a rider accompanies a driver for a portion of a trip. This trip is pre-planned by the driver and it will being held with or without riders. In ride-sharing, most drivers plans a ride for themselves in the first place and then offer to share the ride with others. While in ride-hailing, drivers make on-demand rides based on riders' requests because they have relatively strong origin constraints and no route or destination constraints.

2.2. Blockchain, Smart Contracts, and Ethereum

The origin concept of blockchain comes from the Bitcoin, a very popular cryptocurrency (Nakamoto, 2008). Blockchain is a buzzword that refers to a powerful technology with many definitions. These definitions collectively agree that the blockchain is a ledger with a decentralized architecture which store records of data with the following characteristics: distributed, shared, chained, chronological, secured through encryption and hashing, traceable, and immutable or append only (Nakamoto, 2008; Yuan & Wang, 2018; WOOD, 2021). These characteristics enables blockchain to exchange values in a p2p way without relying on a third party.

Practically, Blockchain represents a data structure which holds records of digital transactions (Yuan & Wang, 2016; Mollah et al., 2021; Yuan & Wang, 2018). Multiple different nodes, i.e., computing machines, store identical copies of the blockchain. They are connected in a p2p network. Transactions are the fundamental units of blockchain and a group of them are stored in a block. A chain of blocks is formed by continuously appending them in sequence. The decentralization importance is emphasized in the blockchain by enabling most of the participating nodes to collectively take the decision through a process known as consensus mechanism.

Although, the blockchain mainly appears for distributed cryptocurrency that enables the transfer of electronic cash without banks intervention. However, it was not limited to cryptocurrencies since it later evolved beyond that to support the deployment of more general-purpose distributed applications. This concept has been introduced by Vitalik Buterin and is called smart-contracts or decentralized autonomous organizations (WOOD, 2021). Although, the smart-contracts were first introduced in the middle of1990 by Nick Szabo, they did not find the environment in which they can grow (Szabo, 1996). A smart contract can be defined as a special type of computer program which is autonomous, stored and running on blockchain network. This program acts as a contract whose terms can be pre-programmed with the ability of self-executing and self-enforcing without the need for trusted authorities (WOOD, 2021). Smart contracts enable the solidity code to run as DApps on a blockchain system. Solidity is the most common programming language in creating smart contracts for Ethereum. It is derived from C++, JavaScript, and Python.

Ethereum is a blockchain platform similar to Bitcoin. Its existence comes as a response to the question of how to use the blockchain technology beyond the money and cryptocurrencies. As a result, it aims to allow developers or anyone to create arbitrary smart contracts and decentralized applications (DApps) that are scalable, standardized, feature-complete, easy to develop and interoperable. The Ethereum was born in 2013 by Vitalik Buterin who is a programmer and cryptocurrency researcher (Vitalik, 2014). It represents a public, permissionless, and global blockchain which built with Turing-complete programming language, Solidity, to overcomes several limitations of Bitcoin's scripting language. Every computation on Ethereum costs fees, and as a result, it defines the term Gas for measuring fees which are paid in Ether. Ether is the built-in cryptocurrency of Ethereum. Apart from payment, it is also used as a pricing instrument for running DApps in the system. All Ethereum operations and instructions are executed by Ethereum Virtual Machine (EVM) which runs in every node of the network.

In Ethereum, nodes[1] refer to a running piece of client software while client is an implementation of Ethereum that verifies all transactions in each block. Based on the data storage rate, the Ethereum network has three different types of nodes: full, light, and archive. Full node stores all blockchain data. Light node stores only the header chain and requests everything else, while archive node stores everything kept in the full node plus building an archive of historical states. Some of the potential domains of Ethereum are Transportation, Healthcare, Insurance, File Storage, Market Predictions, etc. Decentralized applications (DApps) for transportation are one of the significant use-cases of Ethereum that will be explored in this paper. A DApp is an application hosted on a p2p blockchain network. Key differences between traditional centralized web applications and DApps are shown in Table 1 below.

2.3. InterPlanetary File System (IPFS)

IPFS is a p2p distributed file system that has the goal of connecting all computing devices with the same system of files (Benet, 2014; Guidi et al., 2021). In some ways, it looks like the Web in providing a high throughput content-addressed block storage model and content-addressed hyperlinks. The IPFS depends on a Distributed Hash Table (DHT), namely Kademlia, to keep track of who can provides what data (Daniel & Tschorsch, 2022). Upon storing data, it splits files into chunks which identified by their own hash and then recording which chunks are belong to which file. These chunks then will be distributed to various nodes on the network. When a user requests a chunk, this request traverse the DHT to nodes where the hash exists there. All chunks, when needed, are combined to prepare the main object, and show it to the user. IPFS uses hashes for identification and retrieval of files. As a result, this provides the highly useful property of being able to verify a file by checking its hash against the hash of the file you were looking for. The equality of the two hashes acts as proof that the right file was received.

IPFS depends on what is called content identifier, or CID, because it represents a content-addressing system rather than location-addressing one (Benet, 2014). CID[2] is a self-describing content-addressed identifier which forms an address based on the content itself. Its size depends on the cryptographic hash instead of the content itself. By default, IPFS hashes its content using the cryptographic secure hash algorithm2-256 (sha2-256) which produces a CID that is 256 bits or 32 bytes. It supports multiple hashing algorithms through what is called multihashing. This multihashing represents a self-describing hash which itself contains metadata describing the cryptographic algorithm which generate it and its length. Multihashing format consists of three fields which are as

Table 1
Traditional web applications vs. DApps.

	Traditional web applications	Decentralized applications (DApps)
Interaction	Follows client-server architecture	Done directly via smart contracts
Authority	Depend on central authority for setting roles and permissions	No central authority
Privacy	No privacy	Less privacy issues
Transparency	Lack transparency	Transparent
Others	Security issues	More secure, autonomous, and immutable

[1] https://ethereum.org/en/developers/docs/nodes-and-clients/

[2] https://proto.school/anatomy-of-a-cid

follows: (1) one byte length which denotes the type of hash, e.g., sha2-256 would be 18-0 × 12 in hexadecimal; (2) another one byte and denotes the hash size where the length of sha2-256 is 256 bits or 32 bytes; and (3) the actual hash value or hash digest.

The IPFS used base encoding to represent CIDs as strings rather than plain binary as a series of 1s and 0s. IPFS used *base58btc* encoding to creates CIDs, when it was first created. The first version of the CID, version 0 (CIDv0), is formed from multihashing format and base58btc. CIDv0 is always has a prefix *Qm*. For instance, *QmP5Ubgn8SMpwxKqy9TgjHn2HVSaab8ia2R28we6AyBMb2 is a* CIDv0 example which is visualized online using CID Inspector[3] as shown in Fig. 1. It lists both the multibase and multicodec as "implicit", since it did not have those prefixes and always assumed to be *base58btc* and *dag-bp*. Multicodec refers to a pre-set number which uniquely identify a format or a protocol. More details on the IPFS and its related technologies can be found in (Benet, 2014).

3. Related Work

This section discusses the previous research of the ride-sharing, especially blockchain-based research. Blockchain-based ride-sharing services (RSS) are gaining traction because of allowing direction connection between people and drivers who is wanting to transport them (Yuan & Wang, 2016). Despite this, they are still in their early stages and starving to more research contributions (Çaldağ & Gökalp, 2020; Mahmoud et al., 2022).

Most of the previous research has sought only to introduce the blockchain technology in the ride-sharing field in order to move it from centralization to decentralization (Mahmoud et al., 2022). These research works are (Yuan & Wang, 2016; Kudva et al., 2020; (DACSEE, 2018); (Arcade, 2015); Vazquez & Landa-silva, 2021; Khanji & Assaf, 2019; Wang & Zhang, 2021; Shivers et al., 2019; Shivers et al., 2021; Pal & Ruj, 2019; Kanza & Safra, 2018). La'Zooz (Yuan & Wang, 2016), DACSEE (DACSEE, 2018), and Arcade (Arcade, 2015) are application scenarios which presented blockchain-based ride-sharing platforms. The work (Kudva et al., 2020) exploited the consortium blockchain along with smart contracts to overcomes the raised concerns of the current centralized ride-hailing services. The work (Vazquez & Landa-silva, 2021) mainly focused on the utilization of blockchain smart contracts in implementing a ride-sharing systems. While the study (Khanji & Assaf, 2019) investigated the benefits of utilizing blockchain features of decentralization and distribution and built a ride-sharing application namely GreenRide. The work (Wang & Zhang, 2021) proposed a framework for securing ride-sharing via blockchain inherit features. The studies (Shivers et al., 2019) and (Shivers et al., 2021) proposed a framework to develop a decentralized architecture for ride-hailing based on the blockchain and chaincode i.e., smart contracts in Hyperledger Fabric blockchain platform. The work (Pal & Ruj, 2019) focused on utilizing blockchain inherits features, i.e., transparency, decentralization, and distribution, to guarantee fairness in car-sharing. And finally, the work (Kanza & Safra, 2018) Illustrated how using blockchain, cryptocurrency, and smart contracts in ride-hailing services can preserve location privacy, pseudonym of users and also could be trust.

Beyond blockchain inherit features utilization in ride-sharing, the works (Baza et al., 2019; Baza et al., 2020; Renu & Banik, 2021; Li et al., 2019) didn't only introduce the blockchain technology in ride-sharing but also provided further improvements outside of blockchain utilization. Authors of (Baza et al., 2019; Baza et al., 2020) proposed a decentralized ride-sharing system dubbed B-Ride. B-Ride exploited public blockchain and smart contracts to allow direct interaction between system users, drivers and riders, without relying on trusted third-party. Users can use this system while their privacy is preserved through the cloaking technique. B-Ride leveraged trust among riders and drivers by the proposed time-locked deposit protocol and zero-knowledge. A pay-as-you-drive methodology was proposed to ensure fair payment which is based on the elapsed distance. For (Renu & Banik, 2021), authors presented an improved version of existing blockchain-based framework which replaces centralized framework for a ride-sharing service. They then implemented a prototype of the framework as a decentralized application (DApp) based on smart contracts on Ethereum blockchain. additionally, they utilized min-matching algorithm for matching riders with suitable drivers to save total travel distance. The work (Li et al., 2019) proposed a ride-sharing scheme using vehicular fog computing with blockchain. Matching riders with drivers are performed locally through installed Road-Side Units (RSUs) along roads. To enable data auditability, this RSUs maintain a private blockchain for storing ride-sharing data in distributed ledger. However, the dependency on RSU to store massive records of ride-sharing data represents a limited capabilities device which may be impractical especially in urban areas where there is a high demand to ride-sharing services.

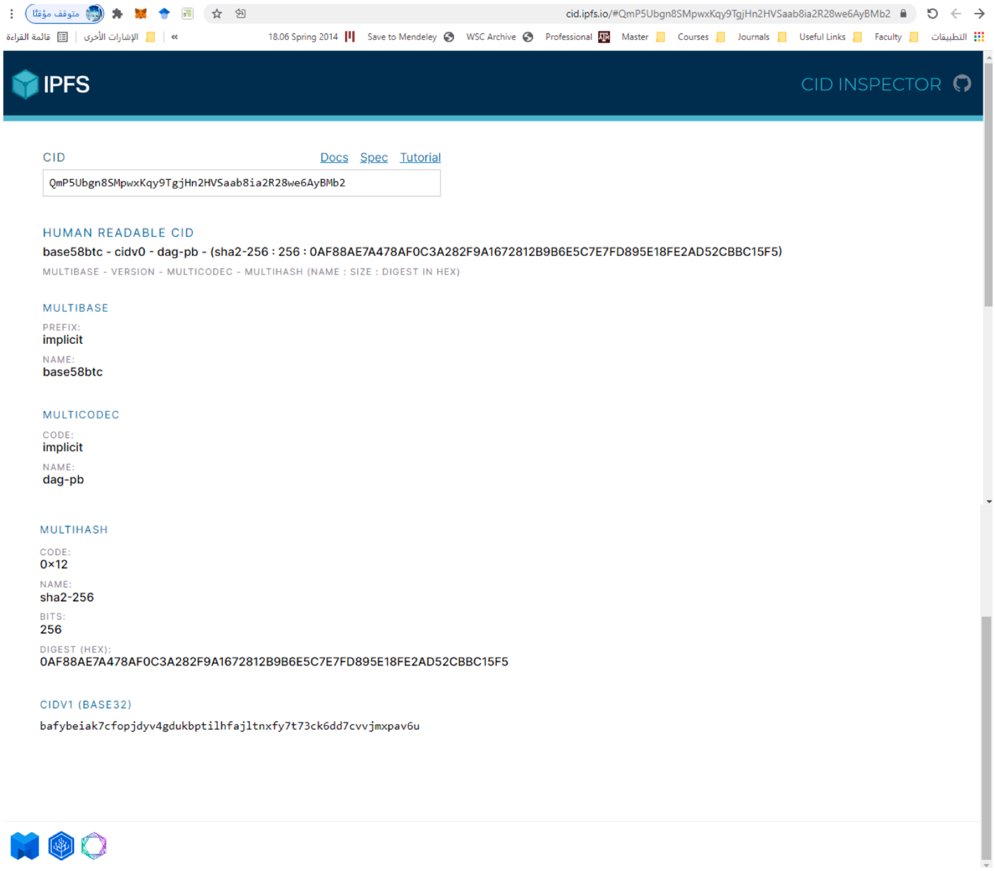
Apart from blockchain decentralization, a decentralized ride-sharing scheme was proposed in (Sánchez et al., 2016). In this work, the service provider was replaced by a p2p ride-sharing management network. Drivers and riders are the peers of this network. However, the scheme does not preserve the riders' privacy since not only the driver and the passenger(s) whose trips match who learn each other's real identity. additionally, the scheme lacks the transparency that the blockchain provides.

We can summarize blockchain-based ride-sharing works as follows based on the previous paragraphs and our survey (Mahmoud et al., 2022). The majority of earlier research focused mostly on introducing blockchain technology to the ride-sharing domain in order to move it from centralization to decentralization. Some research papers addressed only the utilization of blockchain in ride-sharing, but others also extending their proposal beyond this utilization to include privacy and confidentiality enhancement. More details and comparisons on almost all blockchain-based ride-sharing works can be found in our work (Mahmoud et al., 2022).

Although IPFS has not been discussed and introduced in RSSs before, to the best of our knowledge, it has been utilized for other applications. For instance, Norvill et al (Norvill et al., 2018) presented a method to reduce the size of the stored chain data in Ethereum blockchain by moving contract creation data off-chain and replacing it with a smaller hash in blockchain which allows the retrieval of the code. The work in (Alizadeh et al., 2020) proposed combining public blockchain and distributed hash table (DHT) to immutably store data in a decentralized way to mitigate the high storage cost of using blockchain only.

In the data sharing domain, the works (Ye & Park, 2021), (Dammak et al., 2022), and (Ullah et al., 2022) utilized the IPFS. For (Ye & Park, 2021), authors addressed the issues of security and capacity of storing and sharing vehicles data. They proposed a system that uses blockchain and IPFS that securely stores vehicle data. Last but not least, the study (Hossan et al., 2021) aimed at securing the ride of ride-sharing services from some difficulties during its operation. For instance, the driver may misconduct with riders, have trouble hiring, verbalize abusively, or be a victim of harassment. The ride security was achieved through recording a video for the start of takeoff until the finish. Finally, this video saved to the IPFS. Additionally, authors had provided a hyperledger-based private blockchain solution for recording all ride-sharing data namely Ride requests, ride acceptance, ride completion, and ride payment in order to stop the unethical comfort. In contrast to our ride-sharing domain, all of the previously stated studies that utilized IPFS target other domains. Additionally, the objectives of this paper include overcoming the storage limitation of blockchain-based ride-sharing systems, lowering processing costs to lower expenses for end users, and ultimately providing a scalable, verifiable and available solution. In ride-sharing services and nearly all other fields, these objectives have not yet been addressed.

[3] https://cid.ipfs.io/#QmP5Ubgn8SMpwxKqy9TgjHn2HVSaab8ia2R28we6AyBMb2



CID

QmP5Ubgn8SMpwxKqy9TgjHn2HVSaab8ia2R28we6AyBMb2

HUMAN READABLE CID

base58btc - cidv0 - dag-pb - (sha2-256 : 256 : 0AF88AE7A478AF0C3A282F9A1672812B9B6E5C7E7FD895E18FE2AD52CBBC15F5)

MULTIBASE - VERSION - MULTICODEC - MULTIHASH (NAME : SIZE : DIGEST IN HEX)

MULTIBASE

PREFIX:
implicit

NAME:
base58btc

MULTICODEC

CODE:
implicit

NAME:
dag-pb

MULTIHASH

CODE:
0×12

NAME:
sha2-256

BITS:
256

DIGEST (HEX):
0AF88AE7A478AF0C3A282F9A1672812B9B6E5C7E7FD895E18FE2AD52CBBC15F5

CIDV1 (BASE32)

bafybeiak7cfopjdyv4gdukbptilhfajltnxfy7t73ck6dd7cvvjmxpav6u

Fig. 1. IPFS CID 0 details in CID Inspector.

4. Proposed System Architecture

The proposed ride-sharing system architecture contains the following entities, as depicted in Fig. 2:

- *Blockchain.* it is the heart of the proposed system which manages all ride-sharing transactions. The applied blockchain type is permissionless, i.e., public, that allows to anyone to act as a driver or passenger. In addition to transactions handling, blockchain supports p2p payment which exploited in currency exchange, i.e., trip's fare between system users. Because Ethereum is a the popular blockchain platform in smart contracts for business applications, the proposed system depends on Ethereum as its blockchain infrastructure.
- *Smart contracts.* The business logic of the proposed ride-sharing system is carried out by smart contract entity which executed by blockchain network. The proposed smart contract is responsible for forcing the system logics, like driver-to-rider payment and vice versa.
- *Drivers and riders.* The rider entity submits a request to the proposed system to obtain a ride-sharing service. The role of this entity is to request a ride, get a ride's offer, pay ride's fare, and provide other information such as rating a ride. While the driver entity shares his/her planned trip or trips with riders. The role of the driver entity is to

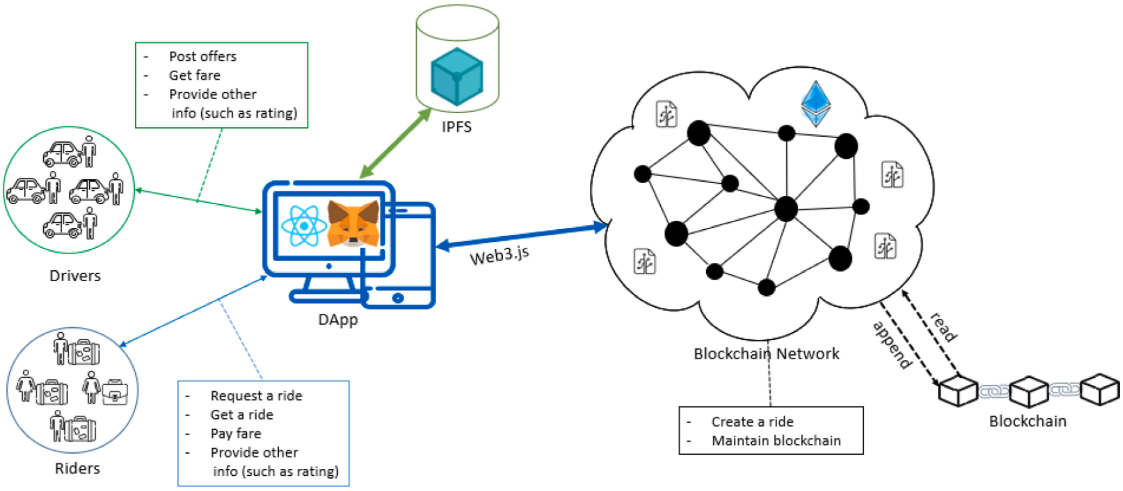


Fig. 2. Proposed ride-sharing system architecture.

provide planned trips as offers to the proposed system, get offer's fare, and also providing other information such as rating.

- *IPFS*. Ride-sharing data of riders' requests or drivers' offers are stored in IPFS. The generated hash is then stored in the blockchain. IPFS stores the data in multiple nodes if the data is higher than a predetermined threshold, for example if data size exceeds 256 KB. This threshold is specified in the IPFS setting. In the context of this paper, the IPFS is utilized as the off-chain database to store ride-sharing data in with the hash of these data being stored in the blockchain.
- *DApp*. It represents the front-end entity that provides the user interface for drivers and riders to access and interact with the proposed system. The DApp has a web interface developed by the React.js framework. It is accessed via the blockchain browser, in our case Chrome plus Metamask extension, that runs on any device type such as phone, laptop, etc.

The proposed system relies on the blockchain for execution correctness and availability, but it suffers from an uncomplete privacy based on the standard blockchain threat model presented in (Baza et al., 2020). Once deployed, the code of a smart contract is guaranteed to work as specified and tampering free. It is visible and readable by anyone in the system as well. Similarly, any data submitted and stored to the contract can be read by all the nodes of the system or any external users. The abbreviations and notations used in this paper are listed in Table 2.

5. Proposed System Methodology

The proposed system methodology is demonstrated in this section. This Methodology illustrates in detail how the integration of the IPFS with the blockchain in the proposed system works as well as how users —namely, drivers and riders—interact with this system. Additionally, the logic and conditions of the proposed system are applied through the developed smart contract which is described in algorithm 1. Data publishing, matching, and selection, processing, and finalizing are the three phases of the proposed system methodology. The following subsections provides an explanation of these phases.

Table 2
Explanation of paper Abbreviations & notations.

Abbreviation/Notation	Refers to
CID	Content IDentifier
DApp	Decentralized Application
DHT	Decentralized Hash Table
EVM	Ethereum Virtual Machine
ETH	ETHER
IPFS	InterPlanetary File System
ITSs	Intelligent Transportation Systems
P2P	Peer-to-peer
RHS	Ride-Hailing Service
RSS	Ride-Sharing Service
$B_{loc}^{(d)}$	Offer's start location of a driver
$B_t^{(d)}$	Offer's start time of a driver
D_o	Driver's offer
$D_t^{(r)}$	Drop-off time for a rider
$D_{loc}^{(r)}$	Drop-off location for a rider
$E_{loc}^{(d)}$	Offer's end location of a driver
$E_t^{(d)}$	Offer's end time of a driver
R_r	Rider's request
$S_{loc}^{(r)}$	Pick-up location for rider
$S_t^{(r)}$	Pick-up time for rider
$x^{(r)}$	Spatial slack distance that a rider willing to walk to driver pick up location
$y^{(r)}$	Temporal slack delay time that a rider willing to wait.

Algorithm 1
Pseudocode for proposed system smart contract

```
Contract RSDAppSC_IPFS
String offersHash // store driver's hash from IPFS
String requestHash // store rider's request hash from IPFS
Function setOffersHash(_offersHash)
offersHash ← _offersHash
Function getOffersHash ()
return offersHash
Function setRequestHash(_requestHash)
requestHash ← _requestHash
Function getRequestHash ()
return requestHash
Function payTripFare (riderAddress, driverAddress, tripFare)
If riderAddress = zero address return;
If driverAddress = zero address return;
riderBalance ← riderAddress.balance
riderBalance ← riderBalance − tripFare
driverBalance ← driverAddress.balance
driverBalance ← driverBalance + tripFare
```

5.1. Data Publishing

The ride-sharing data of drivers and riders as well as how these data are represented and stored in the proposed system are the focus of the data publishing phase. It is shown in Fig. 3's sequence diagram. This phase works as follows. Any rider can submit a request to the proposed system DApp at any time. This request, R_r, can be represented as follows:

$$R_r = \left\{S_{loc}^{(r)};\ D_{loc}^{(r)};\ S_t^{(r)};\ D_t^{(r)};\ x^{(r)};\ y^{(r)}\right\} \tag{1}$$

where $S_{loc}^{(r)}$is the pickup location, $D_{loc}^{(r)}$ is the destination location, $S_t^{(r)}$is the pickup time, and $D_t^{(r)}$is the drop-off time. While $x^{(r)}$ and $y^{(r)}$are two slacks that denote, respectively, distance and time. Where, $x^{(r)}$ represents the distance, in miles, that a rider can walk to meet a driver and $y^{(r)}$ represents the time, in minutes, that a rider can wait a driver to pick him up. $S_{loc}^{(r)}$ and $D_{loc}^{(r)}$ refer to the coordinates of these locations which represented by the latitude and longitude. A rider request R_r example is:

R_r= {30.70634348629828, 30.50644181488037; 30.706786266444926, 30.505154354553223; 1:12; 1:30; 2; 3}.

In this request R_r, $S_{loc}^{(r)}$ is {lat: 30.70634348629828, lng: 30.50644181488037}, $D_{loc}^{(r)}$ is {lat: 30.706786266444926, lng: 30.505154354553223}, $S_t^{(r)}$ is 1:12 PM, $D_t^{(r)}$ is 1:30 PM, $x^{(r)}$ is 2 miles, and $y^{(r)}$ is 3 minutes.

The proposed system DApp entity, then, receives this rider's request, R_r, and forwards it to the IPFS entity. The IPFS stores this request as an object along with its hash that was generated from entire request data. This hash is represented as a content identifier (CID) of the format *QmYVMbTfn3T7oEMcFQBTw6xNVti2VeLEVkaC1q8YxmPiaf*. The blockchain entity receives this CID from the DApp entity, stores it using algorithm 1's *setRequestHash()* function. Finally, the rider waits for recommended offers to be made in response to its request R_r .

In the other side of the proposed system, a driver provides her/his locations and times to the system's DApp before, after, or during the submission of the rider's request, R_r. These locations and times are shown in Table 3.

Based on these driver's locations and times, the DApp prepares driver offers. To illustrate how DApp constructs possible driver offers, the following example is considered. In this example, it is assumed that a driver provided the DApp with five locations and their corresponding times. DApp constructs the possible driver's offers according to the following Table 4:

The number of potential offers from drivers is calculated using Table 4 and the following equation:

$$N_{offers} = n(n-1)/2 \tag{2}$$

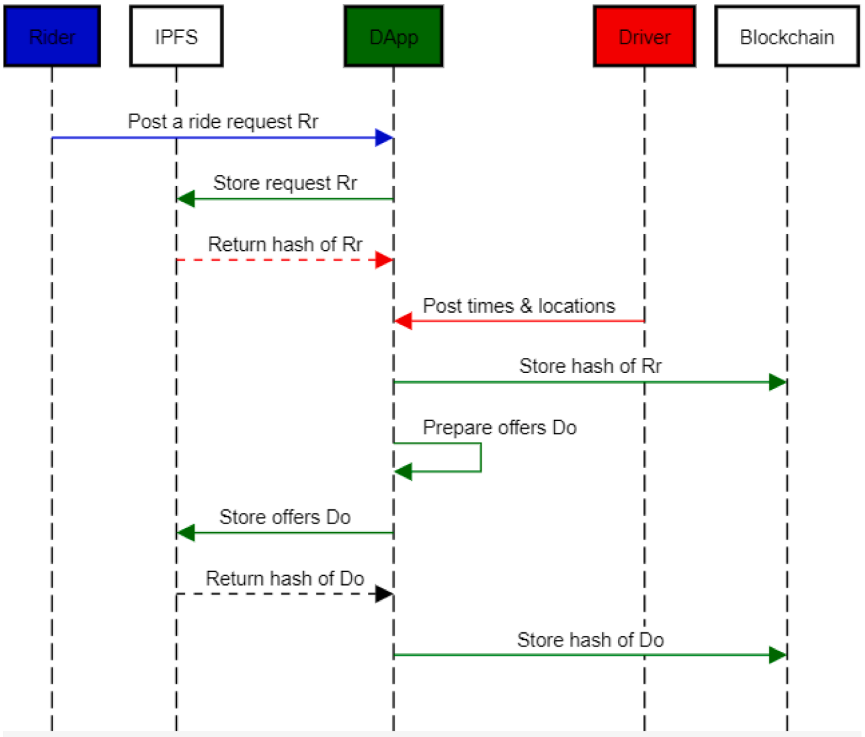


Fig. 3. Proposed system methodology phase 1 (data publishing) sequence diagram.

Table 3
Driver's locations and times.

Location-Time	Location coordinates (L)	Time (T)
(L_0, T_0)	30.705310324719676, 30.50446770904541	1:00
(L_1, T_1)	30.70634348629828, 30.50644181488037	1:10
(…, …)	…, …	…:…
(L_n, T_n)	Lat_n, lng_n	HH:MM

Table 4
The method of driver's offers construction.

Start\end	(L_0, T_0)	(L_1, T_1)	(L_2, T_2)	(L_3, T_3.)	(L_4, T_4)
(L_0, T_0)	-	Offer 1	Offer 2	Offers 3	Offer 4
(L_1, T_1)	-	-	Offer 5	Offers 6	Offer 7
(L_2, T_2)	-	-	-	Offers 8	Offer 9
(L_3, T_3.)	-	-	-	-	Offer 10
(L_4, T_4)	-	-	-	-	-

Where N_{offers} is the number of possible driver's offers and n is the number of location provided by the driver. An offer, D_o, is represented as follows.

$$D_o = \left\{B_{loc}^{(d)}; E_{loc}^{(d)}; B_t^{(d)}; E_t^{(d)}; offer\_price\right\} \tag{3}$$

Where $B_{loc}^{(d)}$ is the start location at time $B_t^{(d)}$ and $E_{loc}^{(d)}$ is the end location at the end time $E_t^{(d)}$. $B_{loc}^{(d)}$ and $E_{loc}^{(d)}$ corresponding to the coordinates of these locations, which are the latitude and longitude. While *offer_price* is the estimated offer fare based on the distance between the locations $B_{loc}^{(d)}$ and $E_{loc}^{(d)}$ of the offer D_o multiply by pre-specified price per mile from the driver. After that, based on the actual locations of the offer D_o and the request R_r at the end of the trip, this offer price may be adjusted. As an example of a driver offer D_o is:

D_o = {30.705310324719676, 30.50446770904541; 30.70634348629828, 30.50644181488037; 1:00; 1:10; 0.0000006501 *USD*}

In this offer D_o, $B_{loc}^{(d)}$ is {lat: 30.705310324719676, lng: 30.50446770904541}, $E_{loc}^{(d)}$ is {lat: 30.70634348629828, lng: 30.50644181488037}, $B_t^{(d)}$ is 1:00 PM, $E_t^{(d)}$ is 1:10 PM, and offer_price is 0.0000006501 *USD*. This price of offer D_o is calculated as follows:

1 The driver specifies that the miles per *Wei,* the smallest unit of Ethereum Ether cryptocurrency, *is* 1000000 *Wei* and the measured distance between $B_{loc}^{(d)}$ and $E_{loc}^{(d)}$ is 221.1976 miles. So, the offer's price can be calculated as follows:

Offer_price = 221.1976 * 1000000 = 221,197,600 *Wei*

1 Next, this offer's price in *Wei* is converted to Ether, Ethereum standard cryptocurrency. This conversion allows us to estimate this offer's price in USD. This is how conversion is calculated:

Offer_price = 221,197,600 * 10^{-18} (*Wei* per *ETH*) = 0.0000000002 *ETH*

1 Now, the offer's price can be estimated in *USD,* given the Ether price $2939 as of 18-1-2022 (Ethereum gas station), as follows:

Offer_price (in *USD*) = 0.00,000,000,02 * 2939 = 0.0000006501 *USD.*

These offers are sent to the IPFS after being constructed by the DApp as possible offers of each driver. The IPFS stores these received offers along with their hash. Then, this hash is forwarded back to the DApp which passes them to the blockchain to store it in its transactions. The *setOffersHash()* function of algorithm 1 is used to store the offers' hash in the blockchain. An instance of a driver's offers hash is *Qmaty9Eb5RSkyos3tNryVNdhY9f9vq1RG2nMAHHqnxUDhS.*

Every file or data associated with a hash is pinned by archive Ethereum nodes in the IPFS in a transaction's data field. Any node can retrieve the ride-sharing data associated with a hash, without the current necessity of storing it permanently as part of a block. The role of archive nodes remains the same under our proposal and the data we move off-chain can be retrieved in the same way. As long as the pinned local file(s) exist it can be requested by other nodes. Full and light Ethereum nodes request information they do not store locally from archive nodes when they need it (Norvill et al., 2018). A rider is now ready to receives the suitable offers for its request. While the driver waits for the suitable riders' request for its offers. The next phase, matching, of the proposed system methodology, which is covered in the following subsection, prepared these suitable offers and requests.

5.2. Matching

According to the rider's needs or preferences, the matching phase's objective is to prepare the suitable offers for each rider's request. This phase is illustrated in the sequence diagram in Fig. 4. It works as follows:

- First, a rider asks the proposed system DApp to provide him with appropriate drivers' offers which fulfil its request.
- In response to rider's request, the DApp retrieves from the blockchain the hashes of rider's request R_r and drivers' offers D_o.
- Then, the DApp utilizes these hashes to retrieve the actual rider's request R_r and drivers' offers D_o from the IPFS. Data availability, correctness, verifiability, and integrity are all guaranteed by retrieving rider's request and drivers' offers from the IPFS based on their hashes from the blockchain.
- Finally, the DApp applies matching to evaluate offers D_o and select the suitable ones for the rider's request R_r. Following are the steps used in the evaluation process:

1. Spatially matching each offer from each driver's start and end locations with the pickup and drop-off locations of rider's request to determine if

$$\left(B_{loc}^{(d)} - S_{loc}^{(r)} \le x^{(r)}\right) \& \left(E_{loc}^{(d)} - D_{loc}^{(r)} \le x^{(r)}\right) \tag{4}$$

Where $x^{(r)}$ is the maximum distance that a rider can walk to meet his or her corresponding driver's start or pickup location, or to reach his final destination.

2. Temporarily matching each offer from each driver's start and end times with the pickup and drop-off times of rider's request to see if

$$\left(B_t^{(d)} - S_t^{(r)} \le y^{(r)}\right) \& \left(E_t^{(d)} - D_t^{(r)} \le y^{(r)}\right) \tag{5}$$

Where $y^{(r)}$ is the maximum time that the rider can wait until his corresponding driver arrives to him or to reach his final destination.

5.3. Selection, Processing and Finalizing

The selection, processing, and finalizing phase demonstrates in detail how a rider and a possible corresponding driver communicate. It is illustrated in Fig. 5. We presume that both the riders and drivers are honest when they interact. The DApp prepared the suitable recommended offers in response to the rider's request in the previous phase, phase 2 matching, and presented them to him. The rider chooses an offer based on his preferences which may var from rider to rider. For instance, some riders might prefer a low-fare offer with a significant distance or waiting time between them and the car. After that, the corresponding driver is notified by the rider's chosen offer. The subsequent steps of this phase operate as described below.

- The selected offer is updated by setting its state to 'initial'. Where, the IPFS receives this new state of the selected offer from the rider then applies the update and produces a new hash. Then, this hash is forwarded to the blockchain to be updated in the corresponding state via its smart contract.
- If the driver is still available, the driver notifies the rider of accepting the selected offer and on that the rider waits the driver to come.
- When the driver arrives at the pickup location, the rider confirms car arrival.
- Then, the DApp updates the corresponding offer state to 'driverConfirmed' in the IPFS followed by the blockchain.
- When the rider reaches its destination, the driver confirms rider arrival. Accordingly, the offer state is updated to 'riderArrival' in the IPFS followed by updating the corresponding state with the new hash in the blockchain.
- Finally, the trip fare is automatically transferred to the driver through the proposed smart contract, i.e., *payTripFare()*, of algorithm 1. The rider confirms his arrival to its destination and the offer state is updated to 'completion' in the IPFS followed by the blockchain.

6. Experimental Setup

The DApp of the proposed ride-sharing system is tested by running different experiments over a local blockchain network namely Ganache. Table 5 provides a brief description of tools, libraries, and frameworks exploited during experiments along with their versions and short description of each one. The ride-sharing smart contract is implemented in Solidity. This contract is created, written, and compiled on Remix IDE and then deployed in Ganache through Truffle. The IPFS desktop and the command line are two variations of the IPFS. The IPFS desktop has a

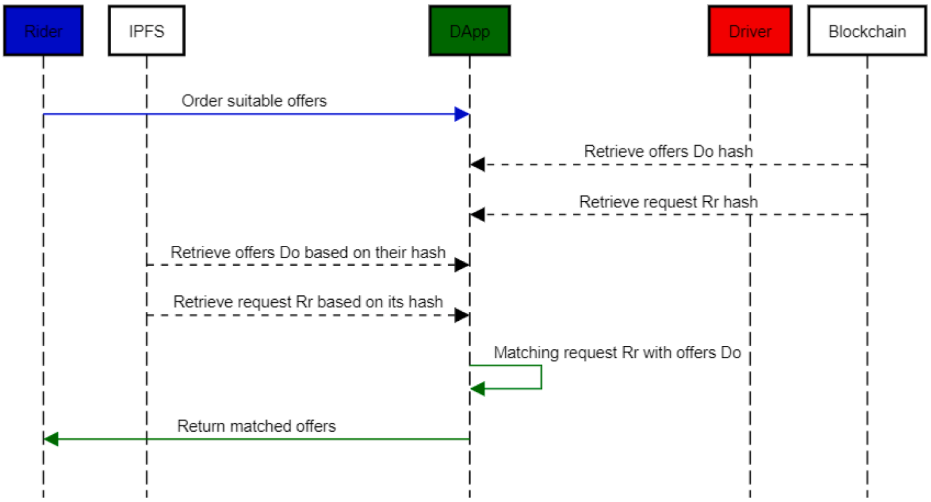


Fig. 4. Proposed system methodology phase 2 (matching) sequence diagram.

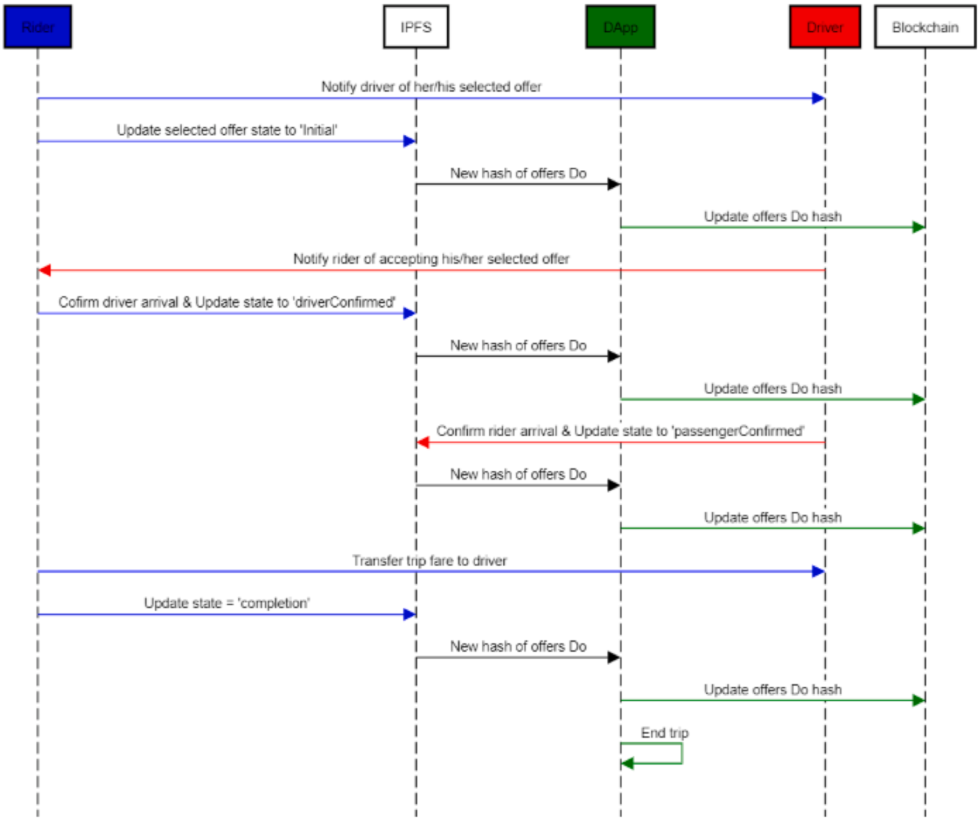

Rider
IPFS
DApp
Driver
Blockchain
Notify driver of her/his selected offer
Update selected offer state to 'Initial'
New hash of offers Do
Update offers Do hash
Notify rider of accepting his/her selected offer
Cofirm driver arrival & Update state to 'driverConfirmed'
New hash of offers Do
Update offers Do hash
Confirm rider arrival & Update state to 'passengerConfirmed'
New hash of offers Do
Update offers Do hash
Transfer trip fare to driver
Update state = 'completion'
New hash of offers Do
Update offers Do hash
End trip

Fig. 5. Proposed system methodology phase 3 (Selection, Processing and Finalizing) sequence diagram.

Table 5
Tools, Libraries, and frameworks exploited in the proposed ride-sharing system.

Tool, library, or Framework	Version	Brief description
Solidity	0.8.1	Object-oriented language for implementing the smart contract in Ethereum
Ganache	2.5.4	Local instance of Ethereum blockchain for emulating blockchain network
Truffle framework	5.3.7	Development environment for client-side applications and for writing, testing, and deploying smart contracts in EVM. Allows the creation of DApps on Ethereum.
Remix IDE	0.21.2	A web IDE that allows to create and deploy, i.e., for example to Ganache, plus executing and exploring the working of smart contracts on the Ethereum blockchain.
Metamask	10.8.1	Is an extension which converts web browsers to blockchain browsers. Allows to manage personal accounts i.e., send and receive transactions plus fetching data from blockchain.
Web3js	1.5.0	Open-source JavaScript library, i.e., API, for interacting with local Ethereum nodes
DApp's web front end	16.14.0	ReactJS which exploited web3js API for accessing blockchain nodes
NodeJS	6.14.12	Backend server for the DApp
IPFS	2.12.4	Inter-planetary File System which provides a distributed and immutable storage for files, data, etc.

visual effect and an easy interface for adding, pinning, and sharing files. IPFS Companion is a browser extension that makes the command line slightly easier to work with. The proposed system utilizes the IPFS desktop. The proposed DApp interface, which is managed and run via the NodeJS server, was developed using the ReactJS front-end library. Through the Web3JS library, the DApp interface and the backend of Solidity are connected. To manage Ganache blockchain accounts using the Chrome browser, the Metamask extension is utilized. Experiments are carried on a machine with the following specifications: Windows 10 Pro 64-bit operating system, Processor 2.10 GHz Intel Core i7, and 4 GB of Memory.

7. Results and Evaluation

The evaluation process of the proposed ride-sharing system is presented in this section. The comparison of the proposed system and ride-sharing former counterparts assesses the functionalities of the proposed ride-sharing system which is presented in Subsection 7.1. The proposed system performance, in terms of computational and storage costs, and its evaluation is next discussed in Subsection 7.2. Finally, Subsection 7.3 analyzes the privacy and security of the proposed system.

7.1. Functionality

Table 6 provides a comparison between the proposed system and other predecessors ride-sharing systems. It summarizes the architecture along with the required functionalities. Table 6's columns represent the metrics included in this summary and the following are their description in order:

- *Architecture.* It refers to the architecture of being centralized or decentralized in storage and communication for the ride-sharing systems. blockchain implicitly provides decentralized architecture.
- *Blockchain type.* It refers to the type of blockchain that is being utilized, either public, consortium, or private. Public blockchain allows anyone to access the network with the ability to enter and depart it at any time. In consortium blockchain type, the network is under the control of more than one organization. Only the nodes of the consortium organizations are allowed to access the blockchain ledger.

Table 6
Comparison between our proposal and other ride-sharing systems.

	Architecture	Type	Privacy	Transparency	Availability	Scalability	Verifiability
Existing RSSs	Centralized	–	×	×	×	×	×
Co-utile (Sánchez et al., 2016)	Decentralized	–	P	×	P	×	×
(DACSEE platform), (Arcade, 2015)	Blockchain	–	×	P	P	×	×
La'Zooz (Yuan & Wang, 2016)	Blockchain	–	P	P	P	×	×
(Li et al., 2019)	Blockchain	Private	G	P	P	×	×
(Shivers et al., 2019) & (Shivers et al., 2021)	Blockchain	Consortium	P	P	P	×	×
(Vazquez & Landa-silva, 2021)	Blockchain	Public	P	P	P	×	×
(Wang & Zhang, 2021)	Blockchain	Consortium	G	P	P	×	×
(Kanza & Safra, 2018)	Blockchain	-	G	P	P	×	×
(Pal & Ruj, 2019)	Blockchain	Private	P	P	P	×	×
(Khanji & Assaf, 2019)	Blockchain	Private	P	P	P	×	×
(Renu & Banik, 2021)	Blockchain	Public	P	P	P	×	×
PAPERS (Kudva et al., 2020)	Blockchain	Consortium	P	P	P	×	×
B-Ride (Baza et al., 2019) & (Baza et al., 2020)	Blockchain	Public	G	P	P	×	×
Our proposal	Blockchain	Public	P	P	G	G	G

P: denotes partial functionality realization; **G**: denotes fully guaranteed functionality realization;
×: denotes not realized functionality;–: denotes not presented or measured;

While the governance of the private blockchain network and the consensus are under the control of a single private organization.

- *Privacy*. It indicates the privacy provided by the paper's proposal. In ride-sharing, the privacy is composed of the following parts (Mahmoud et al., 2022). (1) Anonymity: During a ride, a user's location and identity should be protected from others. (2) Unlink-ability: user requests or responses should not be linked together. (3) Traceability: It should be impossible for any node to determine a user's real identity. (4) Transaction privacy: Sender, receiver, or transferred amount details should be protected from users who are irrelevant to the transaction. Blockchain implicitly provides partial privacy through the anonymity, with each user being identifiable by a set of public and private keys.
- *Transparency*. It refers to the ability of users to view and trace their transactions as well as other related processes.
- *Availability*. It refers to the constant and the on-demand availability of ride-sharing data.
- *Scalability*. It refers to the ride-sharing system ability to scale as its data, processing, or nodes grow.
- *Verifiability*. It refers to the ability to verify the ride-sharing data beyond blockchain verifiability.

According to table 6, we conclude that almost all existing centralized ride-sharing systems lack the majority of functionalities because of their centralization. DACSEE (DACSEE, 2018) and Arcade (Arcade, 2015) are two recently developed ride-sharing platforms based on blockchain. However, the majority of metrics are not taken into account by these platforms. In (Sánchez et al., 2016), a decentralized ride-sharing system is proposed using p2p communications. However, the scheme lacks the transparency that our proposed system achieves. additionally, privacy and availability are partially realized. Major previous blockchain-based works (Shivers et al., 2019), (Shivers et al., 2021), (Vazquez & Landa-silva, 2021), (Pal & Ruj, 2019), (Khanji & Assaf, 2019), (Renu & Banik, 2021) and (Kudva et al., 2020), relied only on blockchain inherit features. They thus provide partial functionality of privacy, transparency, and availability. There are fewer works that addressed fully functional privacy, namely (Li et al., 2019), (Wang & Zhang, 2021), (Kanza & Safra, 2018), (Baza et al., 2019), and (Baza et al., 2020). The proposed system outperforms all other previous works in scalability, availability and verifiability. Moreover, the scalability, availability and verifiability are fully guaranteed thanks to the integration of the IPFS with the blockchain. Although our proposal only partially addresses privacy.

7.2. Performance Analysis and Discussion

This subsection's goal is to evaluate the performance of the proposed ride-sharing system. Additionally, it measures the computational cost of executing the proposed smart contracts within the Ethereum network, namely Ganache. As a result, we can assess the economic viability of running the proposed ride-sharing system.

In general, the term *on-chain* is called on any operation performed inside the blockchain environment. On the other hand, any operation runs outside the blockchain is referred to as *off-chain*. The process of preparing offers, measuring trip distance and the matching process of rider's request with driver's offers are performed off-chain. Whereas the smart contract of the proposed ride-sharing system and its all-related transactions are executed on-chain. The concept of gas is introduced in Ethereum to quantify the computational effort amount required to execute each operation or transaction on its network. The cost is payable in Ether, the native Ethereum currency. In either the smart contract or in the Ethereum Virtual Machine (EVM), the gas cost per operation is fixed. For instance, the addition operation of two variables requires 3 gas, multiplication operation costs 5 gas, whereas computing a SHA3 hash needs 30 gas plus 6 gas for every 256 bits of input (WOOD, 2021). For evaluating the cost of on-chain operations, we rely on the following metrics.

- *Transaction cost*. It is the cost, i.e., in gas, of sending data to the blockchain. Typically, transaction cost is composed of four items; (i) the base cost of a transaction which is 21000 gas (ii) the cost of a contract deployment which is 31000 gas, (iii) the cost of every zero byte of data or code in a transaction which is 4 gas, and (iv) and the cost of every non-zero byte of data or code in a transaction which is 16 gas (Baza et al., 2019; WOOD, 2021).
- *Execution cost*. It indicates the gas cost of actually executing the code included in a transaction by the EVM (Baza et al., 2019).
- *Storage cost*. It is the cost of storing data in the blockchain.

Previous metrics allow us to assess the practicality of running our proposed ride-sharing system DApp on public blockchain via translating them to compute direct monetary cost on both drivers and riders.

Fig. 6 shows the estimated gas cost, i.e., transaction cost, of calling contract functions along with its comparison with other methods. The proposed system provides about 84506 gas for rider's request, 84572 gas for 10 offers of drivers, and 26644 gas for paying trip fare by a rider. Our results are compared to the previous works B-Ride (Baza et al., 2019) and (Vazquez & Landa-silva, 2021). In comparison to the presented previous works, the proposed system provides lower computation overhead and thus lower costs on end users and higher processing speed.

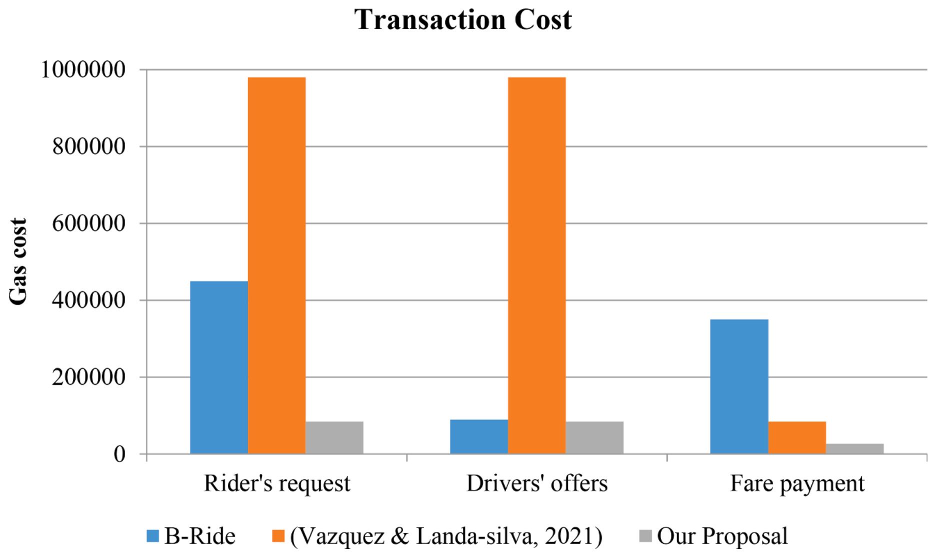


Fig. 6. Gas cost estimation, i.e., transaction cost, for calling smart contract functions.

In Fig. 7, the execution cost of invoking the same three contract functions is presented and compared with the identical works of Fig. 6. As shown in this Fig. 7, the proposed method provides the lower gas cost estimation compared with the presented previous works.

In Fig. 8, the estimated transaction and execution gas costs of driver's smart contract are presented and compared with the previous works (Renu & Banik, 2021) and (Kudva et al., 2020). The driver's smart contract gas cost is the total of all its functions costs. The proposed system gives about 818176 gas for the transaction cost, whereas the execution cost is about 609771 gas. Based on Fig. 8, the proposed system provides lowest results in comparison with other presented works. We can conclude form Figs. 6, 7, and 8 that the proposed ride-sharing system provides efficient computation overhead on the blockchain.

Given gas price of 60.7617 GWei and Ether price $ 2939 as of 18-1-2022 (Ethereum gas station), the estimated cost for a driver that has 10 trips is $ 15.13. The driver's costs raise to $ 31.82 by raising the gas price to 128 GWei,. For a rider, its request cost is estimated in the same way as the driver's offers cost estimation plus the cost of the traveled distance is added at the end of the trip which is very low as explained in subsection 5.1 (5.1. Data Publishing). These results clearly demonstrate that, in comparison to current ride-sharing service providers, the cost is very affordable for the end users. For the storage overhead in Bytes on the blockchain. The data storage on the blockchain for driver's offers, 10 offers, or rider's request is about only 132 Bytes. In comparison with (Baza et al., 2019) which consumed 12 Kbytes in storing 7 offers, still, our associated storage cost is low and practically acceptable.

The impact of the IPFS integration with the blockchain is evaluated. the proposed ride-sharing system's off-chain performance is the focus of this evaluation. The IPFS ride-sharing data storage and retrieval times are shown in Figs. 9 and Fig. 10, respectively. The experiment has been run three times and the time is calculated for each run for time measurement for drivers' trips, i.e., offers. The average time is then calculated for the three experiments' time. For instance, for calculating the time of storing five trips in the IPFS, as shown in Fig. 9, the experiment is executed three times. The calculated time, in seconds, for the three executions are 0.4548, 0.4497, and 0.4914, respectively. Then, the average time of the three executions is 0.465 seconds. While 20 trips are stored in 0.48 seconds.

In Figs. 9, the relationship between the number of trips and the storage time is non-linear. So, the regression line is measured to fits to the data of scattered plot. It represents the average rate of change, which corresponds to the average increase or decrease in the storage time. The

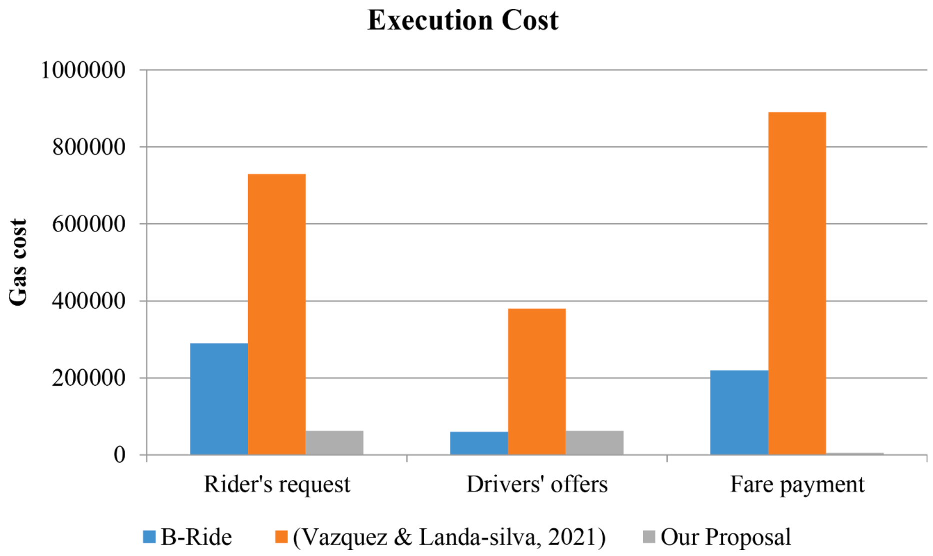


Fig. 7. Gas cost estimation, i.e., execution cost, for calling smart contract functions.

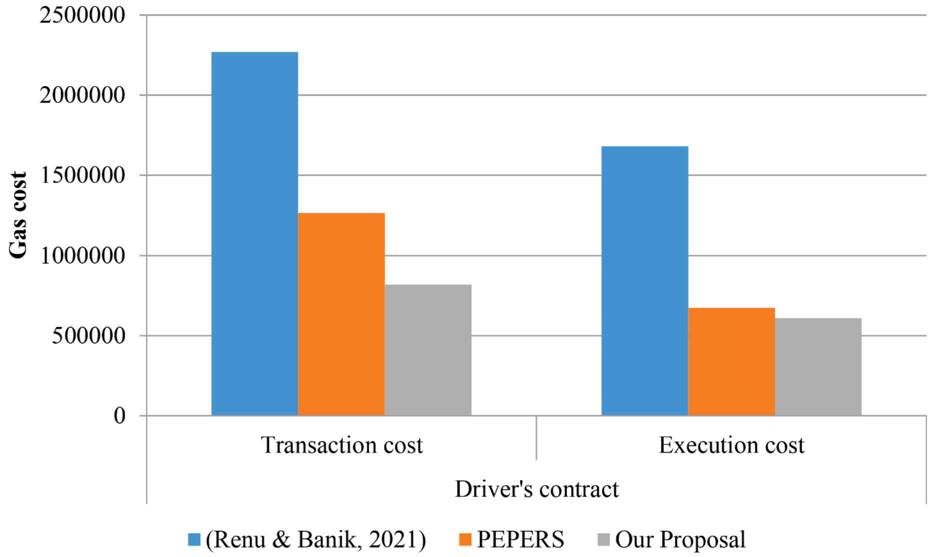


Fig. 8. Gas cost estimation for driver's smart contract.

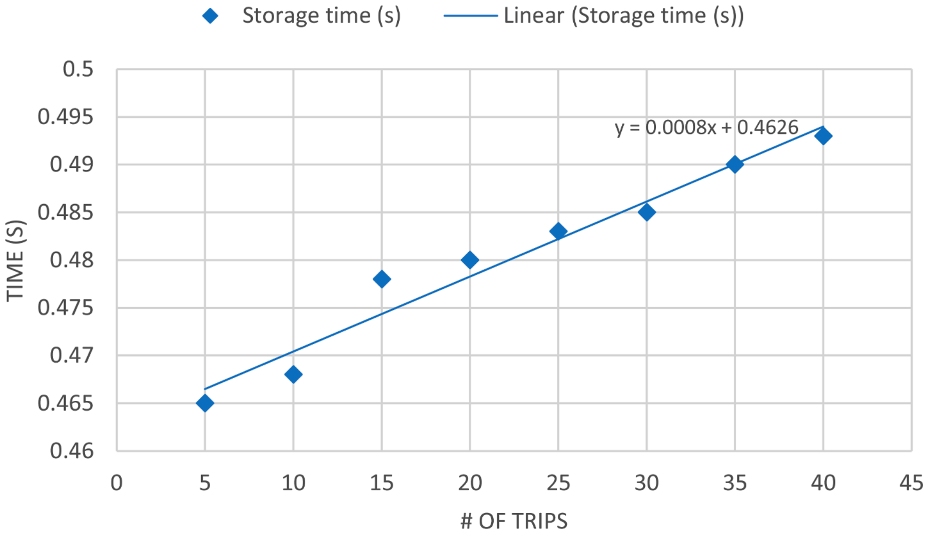


Fig. 9. Storage time of IPFS per number of trips.

linear regression line is represented by the equation y = mx + b, where m refers to the line slope. In Fig. 9, It has the equation y = 0.0008x+ 0.4626, where its slope is 0.0008. So, due to the increase in the number of trips, the average increasing in the storage time is 0.0008. With the increase of the number of trips, the time needed for storing these trips in the IPFS is slightly increased. From the former explanation, we conclude that the proposed ride-sharing system is efficient and scalable. The two indicators that support this conclusion are the non-linear relationship between the number of trips and the storage time, and the average change of 0. 0008.

Fig. 10 depicts the retrieval time of drivers' trips from the IPFS. Retrieving 10 and 30 trips take 0.0091 and 0.0095 seconds, respectively. The scattered plot in Fig. 10 demonstrates the non-linearity relationship between the number of trips and the retrieval time. Therefore, the regression line is calculated. It has the equation y = 5E-05x+ 0.008. It has 5E-05 slope. Therefore, due to the increase in the number of trips, the average increasing in the retrieval time is 5E-05. From the former explanation, we conclude that the proposed ride-sharing system is efficient. This conclusion is based on the two indicators: (1) the non-linear relationship between the number of trips and the retrieval time, and (2) the average change of 5E-05. According to the Figs. 9 and 10, we can conclude that the integration of the IPFS with the blockchain will not degrades the system performance where the storage and retrieval times of the IPFS have a slight effect on the performance.

7.3. *Analysis of Security and Privacy*

In this subsection, we discuss security and privacy concerns for proposed ride-sharing system on public blockchain and IPFS, as well as how it addresses with each of them. The proposed ride-sharing system have the following unique features:

Verifiability. Beyond blockchain auditability, the IPFS provides highly useful property of being able to verify a file or data by checking its hash against the hash of the file or data you were looking for. Additionally, the inclusion of the ride-sharing data hash in the blockchain transaction provides efficient and guaranteed retrieval of corrected and unchanged data from IPFS with their hash. The equality of the two, stored hash in IPFS and the retrieved from blockchain, acts as proof that the right file or data was received. Data verifiability, correctness, and integrity are all guaranteed by retrieving rider's request

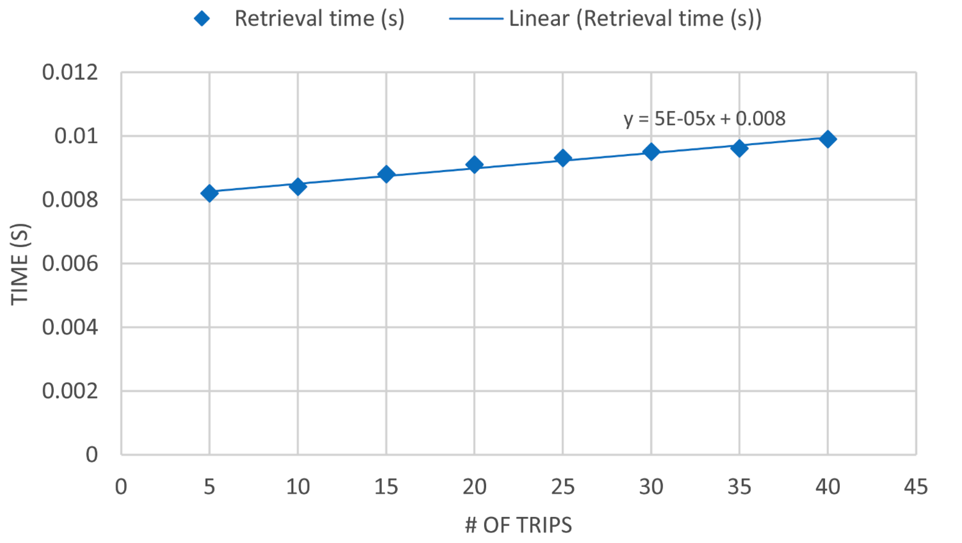


Fig. 10. Retrieval time from IPFS per number of trips.

and drivers' offers from the IPFS based on their hashes from the blockchain.

Scalability. The proposed blockchain-based ride-sharing system has scalable data storage thanks to the integration of IPFS with the blockchain. The burden and limitation of storing these data in the blockchain will be reduced by moving the ride-sharing data to the IPFS and storing their hash in the blockchain transaction instead. In addition, the IPFS's decentralized storage nodes can accommodate the growth in the ride-sharing data. Besides that, the proposed system can scale well without affecting its performance, especially blockchain, as the ride-sharing data size increase.

Also, the following features is unique compared with other centralized ride-sharing systems:

Decentralized ride-sharing service. No single authority can monopolize the system for its own benefits, such as raising prices, thanks to the blockchain technology. Furthermore, the selection process of drivers and riders are transparent as it works as a distributed auction process. Thanks to immutability and tamper-proof for data and smart contracts which ensure data integrity and no interfere with codes without the consent of all the nodes in the blockchain network.

Achieving transparency. Because each user has access to the blockchain, whether they are drivers or riders, they can assure and verify that their ride request or offer has been received and processed by the blockchain network. The phases of matching, and selection, processing, and finalizing are performed on public. Each driver can thereby be ensured that its offers have been sent, received, and processed by the blockchain network without bias to other drivers. Riders can trace her/his request and ensure that the suitable driver is recommended to its request. Therefore, the proposed ride-sharing system ensures transparency which is not offered through other centralized approaches.

8. Conclusion and future work

In this paper, we proposed a ride-sharing system that integrates IPFS with the blockchain. This system addresses the deficiencies of blockchain-based ride-sharing systems of storage limitation, lack of scalability and data storage growing in blockchain, which raises the service costs to end users. In the proposed system, we proposed a method for reducing the size of ride-sharing data stored in the blockchain by moving these data off-chain to the IPFS and replacing it with smaller hashes. Experiments demonstrate that the proposed system able to reduce the burden associated with storing ride-sharing data in the blockchain. As a result, the computation and processing time are reduced and thus costs on end users are minimized. The proposed system is also scalable, efficient, and verifiable. Additionally, experimental findings demonstrate that the proposed method achieved minimum impact on system performance and ensures nodes retain the ability to fully verify transactions.

For future work, we consider enhancing the privacy provided by the blockchain and securing the system against common threats. We also consider improving trust and the matching process between riders and drivers. Additionally, we plan to use remote pinning servers like Infura, Pinata, etc., rather than depending only on local nodes pinning.

CRediT authorship contribution statement

Nesma Mahmoud: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization, Investigation. **Asmaa Aly:** Supervision, Writing – review & editing. **Hatem Abdelkader:** Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aïvodji, U. M., Huguenin, K., Huguet, M.-. J., & Killijian, M.-. O. (2018). SRide : A privacy-preserving ridesharing system. In *Proceedinds 11th ACM Conference Security Privacy Wirelless Mobile Networks* (pp. 40–50). https://doi.org/10.1145/3212480.3212483

Alizadeh, M., Andersson, K., & Scheľen, O. (2020). Efficient decentralized data storage based on public blockchain and IPFS. IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2, 1–8. https://doi.org/10.1109/CSDE50874.2020.9411599.

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, PP(c)*, 1–16. https://doi.org/10.1109/TNSE.2019.2959230

Baza, M., Mahmoud, M., Srivastava, G., Alasmary, W., & Younis, M. (2020). A light blockchain-powered privacy-preserving organization scheme for ride sharing services. In *Proceedings of the IEEE 91th Vehicular Technology Conference (VTC-Spring), Antwerp, Belgium* (pp. 1–6). https://doi.org/10.1109/VTC2020-Spring48590.2020.9129197

Arcade city. [Online]. (2015) Available: Https://arcade.city/.

Benet, J. (2014). IPFS - content addressed, versioned, P2P file system (DRAFT 3). *ArXiv Preprint ArXiv:1407.3561, Draft 3*.

Çaldağ, M. T., & Gökalp, E. (2020). Exploring critical success factors for blockchain-based intelligent transportation systems. *Emerging Science Journal, 4*(Special Issue), 27–44. https://doi.org/10.28991/esj-2020-SP1-03

Chau, S.C., .Shen, S., & Zhou, Y. (2020). Decentralized ride-sharing and vehicle-pooling based on fair cost-sharing mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. https://doi.org/10.1109/TITS.2020.3030051.

Chau, S. C., Shen, S., & Zhou, Y. (2020b). Decentralized ride-sharing and vehicle-pooling based on fair cost-sharing mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. https://doi.org/10.1109/TITS.2020.3030051

Dammak, B., Turki, M., Cheikhrouhou, S., Baklouti, M., Mars, R., & Dhahbi, A. (2022). LoRaChainCare : An IoT architecture integrating blockchain and LoRa network for personal health care data monitoring. *Sensors, 22*(4), 1–24. https://doi.org/10.3390/s22041497

DACSEE platform. [Online]. (2018) Available: Https://dacsee.com/.

Daniel, E., & Tschorsch, F. (2022). IPFS and friends : A qualitative comparison of next generation peer-to-peer data networks. *ArXiv :2102.12737v3*, 1–22.

Ethereum gas station, ([*Online*]). Available: Https://ethgasstation.info/.

Guidi, B., Michienzi, A., & Ricci, L. (2021). Data persistence in decentralized social applications : The IPFS approach. In *IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)* (pp. 1–4). https://doi.org/10.1109/CCNC49032.2021.9369473

Hossan, M. S., Khatun, M. L., Rahman, S., Reno, S., & Ahmed, M. (2021). Securing ride-sharing service using IPFS and hyperledger based on private blockchain. In *2021 24th International Conference on Computer and Information Technology (ICCIT)* (pp. 1–6). https://doi.org/10.1109/ICCIT54785.2021.9689814

Kanza, Y., & Safra, E. (2018). Cryptotransport : Blockchain-powered ride hailing while preserving privacy, pseudonymity and trust. In *GIS proceeding ACM international symptoms advanced geographic information systems* (pp. 540–543). https://doi.org/10.1145/3274895.3274986

Khanji, S., & Assaf, S. (2019). Boosting ridesharing efficiency through blockchain : GreenRide application case study. In *2019 10th International Conference on Information and Communication Systems (ICICS)* (pp. 224–229). https://doi.org/10.1109/IACS.2019.8809108

Kudva, S., Norderhaug, R., Badsha, S., Sengupta, S., & Kayes, A. S. M. (2020). PEBERS: Practical Ethereum blockchain based efficient ride hailing service. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020* (pp. 422–428). https://doi.org/10.1109/ICIoT48696.2020.9089473

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

Mahmoud, N., Aly, A., & Abdelkader, H. (2022). Ride-sharing services : From centralization to decentralization. *International Journal of Computers and Information (IJCI), Article in*, 1–18. https://doi.org/10.21608/IJCI.2022.130527.1027

Maskey, S. R., Badsha, S., Sengupta, S., & Khalil, I. (2020). BITS : Blockchain based intelligent transportation system with outlier detection for smart city. In *2020 IEEE international conference on pervasive computer Communication work. PerCom work* (pp. 1–6). https://doi.org/10.1109/PerComWorkshops48775.2020.9156237

Mollah, M. B., Zhao, J., Niyato, D., Guan, Y. L., Yuen, C., Sun, S., et al. (2021). Blockchain for the internet of vehicles towards intelligent transportation systems: A survey. *IEEE Internet of Things Journal, 8*(6), 4157–4185. https://doi.org/10.1109/JIOT.2020.3028368

Nakamoto, S. (2008). *Bitcoin : A peer-to-peer electronic cash system*. 1–9. "Self-published paper" [Online]. Available: Https://bitcoin.org/bitcoin.pdf.

Norvill, R., Borja, B., Pontiveros, F., & Cullen, A. (2018). IPFS for reduction of chain size in Ethereum. In *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)* (pp. 1121–1128). https://doi.org/10.1109/Cybermatics

Pal, P., & Ruj, S. (2019). BlockV : A blockchain enabled peer-peer ride. In *2019 IEEE international conference on blockchain (Blockchain)* (pp. 463–468). https://doi.org/10.1109/Blockchain.2019.00070

Priya, D.W.D., Srihari, D.T., .& Kamlimuthu, Y. (2021). Intelligent Transport Systems (ITS). In *recent challenges in science, engineering and technology* (Issue January). https://doi.org/10.1002/9781118557495.ch6.

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, 11*(2), 167–173. https://doi.org/10.18280/ijsse.110205

Ride sharing market (2021). Available: Https://www.globenewswire.com/news-release/2021/12/01/2343775/0/en/Ride-Sharing-Market-Global-Statistics-2021-2028-Ride-Sharing-Industry-Size-Share-Growth-Factors-Forecast.html.

Sánchez, D., Martínez, S., & Domingo-Ferrer, J. (2016). Co-utile P2P ridesharing via decentralization and reputation management. *Transportation Research Part C: Emerging Technologies, 73*, 147–166. https://doi.org/10.1016/j.trc.2016.10.017

Shahbazi, Z., & Byun, Y. (2022). Blockchain and machine learning for intelligent multiple factor-based ride- hailing services blockchain and machine learning for intelligent multiple factor-based ride-hailing services. *Computers, Materials & Continua, 70*(3), 4429–4446. https://doi.org/10.32604/cmc.2022.019755

Shivers, R., Rahman, M. A., Faruk, M. J. H., Shahriar, H., Cuzzocrea, A., & Clincy, V. (2021). Ride-hailing for autonomous vehicles : Hyperledger fabric-based secure and decentralize blockchain platform. In *2021 IEEE international conference on big data (Big data)* (pp. 5450–5459). https://doi.org/10.1109/BigData52589.2021.9671379

Shivers, R., Rahman, M.A., .& Shahriar, H. (2019). Toward a secure and decentralized blockchain-based ride-hailing platform for autonomous vehicles. *ArXiv: 1910.00715v2*, 1–12. https://doi.org/10.48550/arXiv.1910.00715.

Szabo, N. (1996). Smart contracts : Building blocks for digital markets. Https://Www.Fon.Hum.Uva.Nl/Rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/Szabo.Best.Vwh.Net/Smart_contracts_2.Html, 1–11. https://www.researchgate.net/publication/340376424_Smart_Contracts_Building_Blocks_for_Digital_Transformation.

Uber China data breach (2018). Available: Https://www.nytimes.com/2018/09/26/technology/uder-data-breach.html.

Ullah, Z. I. A., Raza, B., SHAH, H., KHAN, S., & Waheed, A. (2022). Towards blockchain-based secure storage and trusted data sharing scheme for IoT environment. *IEEE Access : Practical Innovations, Open Solutions, 10*, 36978–36994. https://doi.org/10.1109/ACCESS.2022.3164081

Vazquez, E., & Landa-silva, D. (2021). Towards blockchain-based ride-sharing systems. In *Proceedings Ofthe 10th International Conference on Operations Research and Enterprise Systems (ICORES 2021), Icores* (pp. 446–452). https://doi.org/10.5220/0010323204460452

Vitalik, B. (2014). Ethereum white paper: A next generation smart contract & decentralized application platform. *Ethereum*, 1–36. *January* https://github.com/ethereum/wiki/wiki/White-Paper.

Wang, D., & Zhang, X. (2021). Secure ride-sharing services based on a consortium blockchain. *IEEE Internet of Things Journal, 8*(4), 2976–2991. https://doi.org/10.1109/JIOT.2020.3023920

WOOD, G. (2021). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 1–41.

Ye, H., & Park, S. (2021). Reliable vehicle data storage using blockchain and IPFS. *Electronics, 10*(1130), 1–15. https://doi.org/10.3390/electronics10101130

Yuan, Y., & Wang, F. (2016). Towards blockchain-based intelligent transportation systems. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 2016* (pp. 2663–2668). https://doi.org/10.1109/ITSC.2016.7795984

Yuan, Y., & Wang, F. (2018). Blockchain and cryptocurrencies : Model, techniques, and applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48*(9), 1421–1428. https://doi.org/10.1109/TSMC.2018.2854904

Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems, 12*(4), 1624–1639. https://doi.org/10.1109/TITS.2011.2158001

Zichichi, M., Ferretti, S., & D'Angelo, G. (2020). A framework based on distributed ledger technologies for data management and services in intelligent transportation systems. *IEEE Access : Practical Innovations, Open Solutions, 8*, 100384–100402. https://doi.org/10.1109/ACCESS.2020.2998012