Decentralized Ride Hailing platform based on Blockchain

1st Nilima Patil
Electronics and Telecommunication dept
JSPM's Rajarshi Shahu College Of Engineering, Pune-411033
Maharashtra, India
patilnilima774@gmail.com

2nd Krishna Patil
Electronics and Telecommunication dept
JSPM's Rajarshi Shahu College Of Engineering, Pune-411033
Maharashtra, India
krishnapatil6446@gmail.com

3rd Sahil Kuthe
Electronics and Telecommunication dept
JSPM's Rajarshi Shahu College Of Engineering, Pune-411033
Maharashtra, India
sahilkkuthe2004@gmail.com

Dr. S. A. Bhisikar
Electronics and Telecommunication dept
JSPM's Rajarshi Shahu College Of Engineering, Pune-411033
Maharashtra, India
sabhisikar_entc@jspmrscoe.edu.in

***Abstract*—**

Ride-hailing applications such as Uber and Ola have significantly improved the way people travel but have introduced challenges like unpredictable surge pricing, high intermediary fees, and data privacy concerns. This paper proposes a decentralized ride-hailing platform utilizing blockchain technology to eliminate intermediary fees and enhance transparency and reliability. By leveraging the Solana blockchain and the InterPlanetary File System (IPFS), the proposed solution addresses the inherent problems of traditional ride-hailing systems. The decentralized platform aims to provide a fairer and more transparent service model, reducing costs and increasing user trust. This paper outlines the framework for developing such a platform, detailing the methodologies and technologies involved, and evaluating the potential advantages, limitations, and future applications of the proposed system.

Keywords: Blockchain, Solana, Consensus algorithm, IPFS

I. Introduction

Ride hailing services like Ola, Uber are a great way to book rides from one place to another without searching for a taxi for hours, But these platforms are centralized systems, which can lead to issues such as high service fees, lack of transparency, and data privacy concerns[1-2]. Moreover the drivers who are connected to these apps are charged with heavy commissions which result in less income of drivers and more profits to the company. Blockchain technology offers a potential solution to this problem[3]. `

Ride-hailing services using centralization methods experience various challenges like privacy violation, lack of security, and distributed denial of services (DDoS) attack, etc. So, blockchain-based Ride hailing services mitigate such problems through decentralization[4]. 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[5-6]. Additionally, the blockchain lacks scalability of data because of the inability to store large-sized data and accommodate the growth of ride-sharing data.

As mentioned by Nesma Mahmoud, Asmaa Aly, Hatem Abdelkader [7]. The ride sharing data can be moved outside of the blockchain using Inter Planetary File System (IPFS) and replaced with a small hash. The blockchain will manage the application state and users, it will automate the processes through smart contracts. While the IPFS stores data for blockchain in an immutable and integral way.

Table 1-Centralized System VS Decentralized System

Feature	Centralized System (Uber/Ola)	Decentralized System (Proposed)
Fee Structure	High intermediary fees	Lower intermediary fees
Transparency	Limited	High
Data Privacy	Vulnerable	Enhanced
Security	Prone to attacks	Improved through blockchain
Dispute Resolution	Manual and opaque	Automate through smart contract

II. Literature Survey

A blockchain is a distributed ledger with growing lists of records (blocks) that are securely linked together via cryptographic hashes.Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. Since each block contains information about the previous block, they effectively form a chain (compare linked list data structure), with each additional block linking to the ones before it. Consequently, blockchain transactions are irreversible in that, once they are recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks [5].
Blockchains are typically managed by a peer-to-peer (P2P) computer network for use as a public distributed ledger, where nodes collectively adhere to a consensus algorithm protocol to add and validate new transaction blocks.

Solana blockchain is a high-performance, open-source blockchain platform designed to support decentralized applications (dApps) and crypto-currencies.
It uses a proof-of-stake mechanism to provide smart contract functionality[4].

Consensus in blockchain refers to the process by which a network of nodes (computers) agrees on the validity of transactions and the state of the blockchain. It ensures that all nodes in the network have a consistent and up-to-date copy of the distributed ledger.
Solana uses a unique consensus algorithm called Proof of History (PoH) combined with a delegated Proof of Stake (dPoS) mechanism.

Proof of History (PoH):
Proof of History is a sequence of computation that can provide a way to
cryptographically verify passage of time between two events. It uses a cryptographically secure function written so that output cannot be predicted
from the input, and must be completely executed to generate the output [4].

Delegated Proof of Stake(dPoS):
dPoS enhances the PoH by adding a layer of security and decentralization to the consensus process.
In dPoS, token holders vote to elect a group of validators (or leaders) who are responsible for validating transactions and adding new blocks to the blockchain. Validators are incentivized to act honestly by receiving rewards for their work, and they are penalized or replaced if they act maliciously[4].

In [31], M. Bez et al. raise some concerns about Ethereum's scalability. Every transaction has to be processed by every node on the network. This redundant processing means that as the blockchains' state grows, fewer nodes have the required hardware to do so, which means the network's decentralization suffers.
Solana takes care of the scalability problem by combining PoH and dPoS to handle thousands of transactions per second (TPS), making it one of the fastest blockchains in operation.

The idea of smart contracts was first mentioned by Nick Szabo [28,29]. He describes how contracts written on paper, typically enforced by governments, are the very foundation of the free market economy. He proposes a new digital form of contracts called smart contracts. He describes the four principles when designing smart contracts as observability (everyone should be able to see and prove the execution of a smart contract); verifiability (it should be possible to prove that a smart contract has been executed or failed); privity (knowledge and control over the smart contract should only be described to as many parties as necessary, whereas each additional party with control is a new attack vector); and enforceability (reducing the need for enforcement to a minimum.).

Nick Szabo [28] also describes how cryptographic protocols are the foundation of archiving these principles and explains how they can solve them in detail. He also mentions digital currencies and smart properties. Smart contracts can enable a transfer of ownership of physical property via digital representations; for example, a car could only get started with a digital key which can be transferred via a smart contract in exchange for digital currency.

A smart contract is a script stored on a blockchain [4], and is created by sending a transaction containing the code in some form to an address. Each contract has its unique address on the network. A user can send a transaction to that given address to invoke the smart contract, which is then subsequently executed by every network node receiving the transaction, with the data included in the transaction serving as arguments. Ethereum was one of the first blockchains to support implementing Turing Complete smart contracts and stores the byte-code of deployed smart contracts and their state in Merkle Trees.

Transactions in Ethereum require gas, which is paid in the native asset of Ethereum, Ether [15]. The higher the complexity of the code executing in the smart contract, the higher the required amount of gas, as described in great detail by Mokaluse et al. [16] . Any excess gas after the code execution is sent back to the user. If the user does not send enough gas with the transaction for the code execution, the smart contract's state is rolled back, but all gas is consumed. This mechanism attempts to prevent denial of service attacks on Ethereum since the execution of smart contracts stops once there is no gas left, and gas is an expensive resource. Smart contracts can also invoke other smart contracts, a process which is implemented and activated through messages. Messages are similar to transactions, but can never come from an external source. Such a message only contains the unspent gas of the original transaction sent to the smart contract. The code of smart contracts is typically written in the domain-specific language Solidity and compiled to a bytecode language called the Ethereum Virtual Machine code (EVM). The bytecode of a contract is then executed as infinite loop, until either the execution runs out of gas or it reaches a return statement.

The problem of high gas fees on Ethereum has prompted significant research and development into alternative blockchain architectures that can offer more efficient and cost-effective transaction processing. Among these alternatives, Solana stands out due to its innovative approach to consensus and network design, which effectively mitigates the gas fee problem encountered on Ethereum.

Solana's ability to address high transaction costs is rooted in its high throughput capabilities. The blockchain can process up to 65,000 transactions per second (TPS), a stark contrast to Ethereum's approximate 30 TPS. This high throughput is achieved through parallel transaction processing enabled by Solana's Sealevel technology. Sealevel allows multiple smart contracts to run concurrently, significantly increasing the network's capacity and preventing the congestion that typically drives up gas fees on Ethereum.

Central to Solana's efficiency is its unique Proof of History (PoH) consensus mechanism. PoH acts as a cryptographic clock that timestamps transactions, providing a verifiable order without the need for constant node communication. This drastically reduces the computational overhead required for consensus, leading to faster and more efficient transaction processing. Complementing PoH[4]. is the Delegated Proof of Stake (dPoS) mechanism, which involves a smaller, elected group of validators. This delegation reduces the overall resource requirements for validation, contributing to lower operational costs and thus lower transaction fees.

Solana's network architecture further supports low transaction fees through efficient block propagation and transaction bundling. The Turbine protocol facilitates the rapid propagation of transaction data by breaking blocks into smaller pieces, which are then distributed through the network quickly and efficiently. This minimizes the time and resources needed for block propagation, enhancing overall network performance and reducing costs. Additionally, Solana's ability to bundle multiple transactions into a single block optimizes block space usage, decreasing the cost per transaction.

The predictability and affordability of Solana's transaction fees are crucial advantages over Ethereum. While Ethereum's gas fees fluctuate widely based on network demand, Solana maintains low and predictable fees due to its high throughput and efficient processing mechanisms. Solana also implements a fee market that allows users to prioritize their transactions with higher fees if desired, though even these prioritized fees remain substantially lower than those typically seen on Ethereum.

Support for developers is another important aspect of Solana's strategy to reduce transaction costs. The platform provides various incentives and grants to encourage the development of decentralized applications (dApps) on its network. This support helps foster a vibrant ecosystem without the burden of high transaction fees, drawing projects that might otherwise have been deterred by Ethereum's cost issues.

Finally, Solana's consensus mechanism is less energy-intensive than Ethereum's Proof of Work (PoW). Lower energy consumption translates to reduced operational costs for validators, which in turn helps maintain low transaction fees. This energy efficiency is an increasingly important factor as the blockchain industry moves towards more sustainable practices.

III. Methodology

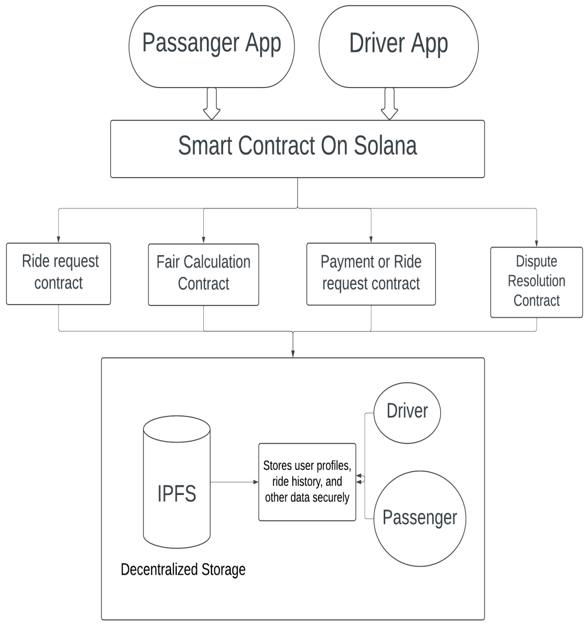

Fig (1) Architecture of System

The figure (1) depicts a decentralized ride-hailing platform architecture using Solana blockchain and IPFS. It consists of a Passenger App and Driver App, which allow users to request and accept rides. At the core, smart contracts on Solana handle essential functions like ride requests, fare calculations, payments, and dispute resolutions.

IPFS provides decentralized storage for user profiles, ride history, and other data, enhancing security and reducing the risk of data loss. This architecture aims to improve transparency, reliability, and efficiency compared to traditional centralized ride-hailing systems.

Here's how you can structure the **Methodology** section of your review paper to explain the flow of your decentralized ride-hailing project, including the technologies used.

The proposed decentralized ride-hailing platform utilizes blockchain technology to eliminate the need for intermediaries, enabling direct interactions between riders and drivers. The system is designed to be transparent, secure, and cost-effective by leveraging smart contracts and the Solana blockchain. The methodology for developing and implementing the project follows a systematic approach, outlined below:

1.System Design and Architecture

The platform is designed as a decentralized application (DApp), where users (riders and drivers) interact with the system directly, without a central authority. The architecture consists of the following components:

-**User Interface (UI)**: The platform will provide an intuitive, user-friendly interface for both riders and drivers, built using modern web technologies such as React.js for the frontend. The UI will facilitate the creation of profiles, ride requests, and ride bookings.

-**Blockchain Network (Solana):** Solana, a high-performance blockchain, will serve as the backbone of the system, ensuring low transaction fees and fast processing. It will handle the core functionalities, such as user authentication, ride requests, and transactions, through smart contracts.

-**Smart Contracts:** Smart contracts will automate various tasks, including ride fare calculation, payment processing, and ride completion. They will ensure that transactions are transparent and trustless, with no third-party involvement. These contracts will be deployed on the Solana blockchain.

- **Off-Chain Storage (IPFS)**: For storing non-critical data such as user profiles, ratings, and reviews, off-chain storage solutions like the InterPlanetary File System (IPFS) will be utilized. IPFS provides a decentralized file storage system that is more efficient and scalable.

2. User Registration and Authentication

- **Rider/Driver Registration:** Both riders and drivers will be required to register on the platform using their wallet addresses, which will serve as their unique identifier in the decentralized network.

- **Authentication via Wallet:** User authentication will be handled via wallet-based login (e.g., MetaMask or Phantom for Solana), ensuring secure access and control over the user's data without needing traditional username/password credentials.

3.Ride Request and Matching

- **Rider Initiates Request:** Riders will submit their ride requests through the platform's UI. The request will include the pick-up and drop-off locations, preferred ride type, and time.

- **Matching Algorithm:** A matching algorithm will evaluate available drivers based on proximity, availability, and ride preferences. Once a match is found, the system will automatically notify both the rider and driver.

4. Smart Contract Execution

- **Fare Calculation and Agreement:** Once a ride is matched, a smart contract is created to lock in the fare based on factors like distance, time, and ride type. The rider and driver both agree to the fare before the ride begins, which is then recorded in the blockchain.

- **Payment Processing:** Upon ride completion, the smart contract will automatically process the payment from the rider to the driver, ensuring that no central authority is involved. The transaction details will be immutably recorded on the blockchain, ensuring transparency.

5.Surge Pricing and Dynamic Adjustments

- **Surge Pricing:** In cases of high demand (e.g., during peak hours), surge pricing will be implemented. The surge multiplier will be calculated based on the demand in the area, and the updated price will be displayed to the rider before confirming the booking.

- **Smart Contract Adjustments:** Surge pricing will be dynamically adjusted by the smart contract based on the real-time data from the network, ensuring that both riders and drivers are fairly compensated.

7. Data Privacy and Security

- **Encryption and Privacy:** All sensitive data (e.g., payment information, ride history) will be encrypted before being stored. Only necessary transaction data will be recorded on the blockchain, ensuring that personal details remain private and secure.

-**Blockchain Security:** The decentralized nature of blockchain ensures that no single entity has control over the entire system. This reduces the risks of data breaches and unauthorized access.

8. Scalability and Performance

- **Solana Blockchain:** To ensure scalability, Solana's high throughput and low-cost transaction processing capabilities are leveraged. Solana can handle thousands of transactions per second (TPS), ensuring smooth operation even during peak usage.

-**Off-Chain Computation:** For less critical tasks, such as user ratings and reviews, off-chain computations will be used to reduce the load on the blockchain, enhancing the system's performance.

I. Technology Stack

A. html : This is the standard language used to create and design web pages and applications. In this application html will be heavily used.

B. NativeWind : NativeWind is a library that provides utility-first styling for React Native applications, inspired by Tailwind CSS. It allows developers to apply styles using a set of predefined utility classes, streamlining the design process. The Application will be built and designed on NativeWind.

C. Javascript : Javascript is a versatile, high-level programming language commonly used to create interactive and dynamic content on websites. It runs in web browsers and can also be used on the server side with environments like Node.js. JavaScript supports object-oriented, imperative, and functional programming styles, making it integral to modern web development, and application development.

D. TypeScript : TypeScript is a superset of JavaScript that adds static typing and advanced features to the language, improving code quality and maintainability. It compiles down to JavaScript, ensuring compatibility with existing JavaScript codebases and environments.

E. React JS : ReactJS is a popular JavaScript library for building user interfaces, particularly single-page applications, with a component-based architecture. It enables efficient, declarative, and reusable UI development through its virtual DOM and state management.

F. Solana : Solana is a high-performance blockchain platform designed for decentralized applications and crypto-currencies, known for its fast transaction speeds and low costs. It uses a unique consensus mechanism called Proof of History (PoH) combined with Proof of Stake (PoS) to achieve high throughput and scalability. Solana's architecture aims to support a wide range of applications, from DeFi to NFTs.

G. Rust : Rust is a systems programming language focused on performance, safety, and concurrency. It provides fine-grained control over system resources while preventing common programming errors like null pointer dereferences and buffer overflows. Rust's ownership model ensures memory safety without needing a garbage collector, making it suitable for both low-level and high-level development.

II. Ease of use

A. *Transparent Pricing and Anonymous Transactions*

Users can see transparent pricing information, which is computed using smart contracts to ensure fairness and consistency. The smart price auditing system[8] works in the background, automatically auditing ride prices and compensating users if any discrepancies are found, without requiring user intervention.

The use of Solana blockchain ensures that transactions are secure and user identities remain anonymous[9-10].

B. *Cross Platform Availability*

The app is available on multiple platforms, including iOS and Android, ensuring that it can be used on a wide range of devices. Build with the help of React-native and javascript as the programming language for developing the user app.

III. Overview

Blockchain is a sequence of blocks, which holds a complete list of transaction records like conventional public ledger and ensures secure, transparent, and immutable record-keeping.

A. *Categories of Blockchain*

Current blockchain systems are categorized roughly into three types: public blockchain, private blockchain and con-sortium blockchain. In public blockchain, all records are visible to the public and everyone could take part in the con-sensus process. Differently, only a group of pre-selected nodes would participate in the consensus process of a consortium blockchain. As for private blockchain, only those nodes that come from one specific organization would be allowed to join the consensus process. The consortium blockchain constructed by several organizations is partially decentralized since only a small portion of nodes would be selected to determine the consensus [5].

B. *Consensus Algorithm*

Consensus in blockchain is the mechanism by which a distributed network of nodes agrees on a single version of the truth for the ledger. It ensures that all participants in the network have a consistent view of the blockchain and that all transactions are valid and ordered correctly.

Consensus algorithm used in solana is Proof of History(PoH) and Proof of Stake(PoS). Proof of History is a sequence of computation that can provide a way to cryptographically verify passage of time between two events. It uses a cryptographically secure function written so that output cannot be predicted from the input, and must be completely executed to generate the output [4].

Proof of Stake is designed for quick confirmation of the current sequence produced by the Proof of History generator, for voting and selecting the next Proof of History generator, and for punishing any misbehaving validators. This algorithm depends on messages eventually arriving to all participating nodes within a certain timeout.

C. *Challenges and Recent Advancements*

Scalability- This requires high transaction throughput, efficient consensus mechanisms like Proof of Stake, and the use of off-chain solutions such as payment channels to enhance speed and reduce costs. Interoperability with other blockchains and traditional systems, along with decentralized infrastructure, helps balance the load and prevent single points of failure. Additionally, optimizing smart contracts reduces computational load and gas fees, enabling the platform to scale effectively while providing fast, reliable, and cost-efficient services.

Data Storage- IPFS

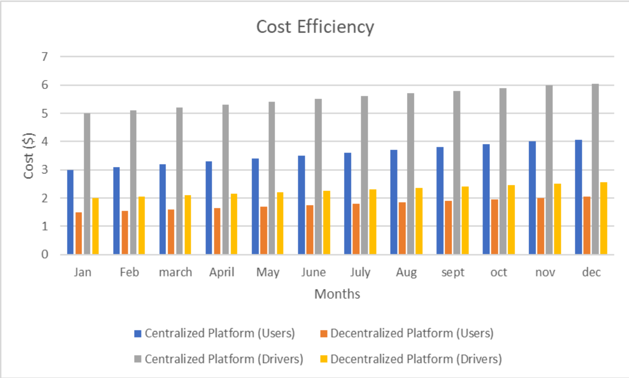

Table 2- Cost Efficiency

Months	Centralized Platform (Users)	Decentralized Platform (Users)	Centralized Platform (Drivers)	Decentralized Platform (Drivers)
Jan	$3	$1.50	$5	$2
Feb	$3.10	$1.55	$5.10	$2.05
march	$3.20	$1.60	$5.20	$2.10
April	$3.30	$1.65	$5.30	$2.15
May	$3.40	$1.70	$5.40	$2.20
June	$3.50	$1.75	$5.50	$2.25
July	$3.60	$1.80	$5.60	$2.30
Aug	$3.70	$1.85	$5.70	$2.35
Sept	$3.80	$1.90	$5.80	$2.40
Oct	$3.90	$1.95	$5.90	$2.45
Nov	$4.00	$2.00	$6.00	$2.50
Dec	$4.10	$2.05	$6.10	$2.55

Conclusion

Centralized ride-hailing platforms like Uber and Ola have improved transportation but come with high fees, unpredictable pricing, and data privacy issues. This paper proposes a decentralized ride-hailing platform using Solana blockchain and IPFS to address these challenges. By eliminating intermediaries, the platform reduces costs for users and drivers while enhancing transparency and security. Smart contracts automate key functions, and decentralized storage ensures data integrity. The proposed model offers a fairer, more efficient solution with significant cost savings, promising to transform the ride-hailing industry.

Future Scope

The proposed decentralized ride-hailing platform opens up several avenues for future research and development. Scalability enhancements can be explored by investigating layer-2 solutions or alternative blockchain technologies to handle increased transaction volumes efficiently. Additionally, user experience improvements can be achieved by refining the user interface and integrating AI-driven features for better ride demand prediction and dynamic pricing. These advancements can further enhance the platform's efficiency, reliability, and user satisfaction.

References

1. S. Kudva, R. Norderhaug, S. Badsha, S. Sengupta and A. S. M. Kayes, "PEBERS: Practical Ethereum Blockchain based Efficient Ride Hailing Service," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 422-428, doi: 10.1109/ICIoT48696.2020.9089473. keywords: {Contracts;Vehicles;Edge computing;Cryptography;Servers;Authentication},

2. O. Naik, N. Patel, S. A. Baba and H. Dalvi, "Decentralized Ride Hailing System using Blockchain and IPFS," 2022 IEEE Bombay Section Signature Conference (IBSSC), Mumbai, India, 2022, pp. 1-5, doi: 10.1109/IBSSC56953.2022.10037296. keywords: {Industries;IEEESections;Userinterfaces;MaintenanceEngineering;Blockchains;InterPlanetary FileSystem;Fuels;Blockchain;Decentralization;Ride-Hailing;Ethereum;IPFS},

3. Y. Lu, Y. Qi, S. Qi, Y. Li, H. Song and Y. Liu, "Say No to Price Discrimination: Decentralized and Automated Incentives for Price Auditing in Ride-Hailing Services," in *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 663-680, 1 Feb. 2022,

doi:10.1109/TMC.2020.3008315 keywords:{Pricing;Insurance;Mobilecomputing;Smartcontracts;Data protection;Distributedledger;Auditing;blockchain;smart contract;ride-hailing;gametheory;insurancescheme;decentralized system},

4. Anatoly Yakovenko Solana: A new architecture for a high performance blockchain v0.8.13

5. Zheng, Zibin & Xie, Shaoan & Dai, Hong-Ning & Chen, Xiangping & Wang, Huaimin. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 10.1109/BigDataCongress.2017.85.

6. R. Shivers, M. A. Rahman, M. J. H. Faruk, H. Shahriar, A. Cuzzocrea and V. Clincy, "Ride-Hailing for Autonomous Vehicles: Hyperledger Fabric-Based Secure and Decentralize Blockchain Platform," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 5450-5459, doi: 10.1109/BigData52589.2021.9671379. keywords: {Distributed ledger;Public key;Organizations;Big Data;Fabrics;Blockchains;Security;Blockchain;Hyperledger Fabric;Ride-Hailing;Ride-Sharing;Information Security},

7. Mahmoud, Nesma & Aly, Asmaa & Abd elkader, Hatem. (2022). Enhancing Blockchain-based Ride-Sharing Services using IPFS. Intelligent Systems with Applications. 16. 200135. 10.1016/j.iswa.2022.200135.

8. Y. Lu, Y. Qi, S. Qi, Y. Li, H. Song and Y. Liu, "Say No to Price Discrimination: Decentralized and Automated Incentives for Price Auditing in Ride-Hailing Services," in IEEE Transactions on Mobile Computing, vol. 21, no. 2, pp. 663-680, 1 Feb. 2022, doi: 10.1109/TMC.2020.3008315.

9. keywords: {Pricing;Insurance;Mobile computing;Smart contracts;Data protection;Distributed ledger;Auditing;blockchain;smart contract;ride-hailing;game theory;insurance scheme;decentralized system},

10. M. Li, Y. Chen, C. Lal, M. Conti, F. Martinelli and M. Alazab, "Nereus: Anonymous and Secure Ride-Hailing Service Based on Private Smart Contracts," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, pp. 2849-2866, 1 July-Aug. 2023, doi: 10.1109/TDSC.2022.3192367.

11. keywords: {Vehicles;Privacy;Blockchains;Smart contracts;Resistance;Mixers;Costs;Ride-hailing services;privacy;collusion attack;private smart contract;SGX},

12. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

13. Buterin, V. (2014). A Next-Generation Smart Contract and Decentralized Application Platform. https://ethereum.org/en/whitepaper/

14. Smith, J., & Brown, R. (2020). The potential of blockchain for transportation systems. *Journal of Transportation Research, 45*(2), 123-135.

15. Buterin, V. A next-generation smart contract and decentralized application platform. White Pap. 2014, 3, 13–17..

16. Mokalusi, O.; Kuriakose, R.; Vermaak, H. A Comparison of Transaction Fees for Various Data Types and Data Sizes of Blockchain Smart Contracts on a Selection of Blockchain Platforms. In ICT Systems and Sustainability: Proceedings of ICT4SD 2022; Springer: Berlin, Germany, 2022; pp. 709–718

17. Y. Lu et al., "Safety Warning! Decentralised and Automated Incentives for Disqualified Drivers Auditing in Ride-Hailing Services," in IEEE Transactions on Mobile Computing, vol. 22, no. 3, pp. 1748-1762, 1 March 2023, doi: 10.1109/TMC.2021.3108012.keywords: {Vehicles;Distributed ledger;Licenses;Security;Insurance;Safety;Mobile computing;Auditing;blockchain;smart contract;ride-hailing;game theory;insurance scheme;decentralized system},

18. H. Yu, H. Zhang, X. Yu, X. Du and M. Guizani, "PGRide: Privacy-Preserving Group Ridesharing Matching in Online Ride Hailing Services," in IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5722-5735, 1 April1, 2021, doi: 10.1109/JIOT.2020.3030274. keywords: {Vehicles;Privacy;Aggregates;Encryption;Internet of Things;Protocols;Encrypted distance;group ridesharing matching;online ride hailing (ORH);privacy preserving},

19. Measuring when Uber behaves as a substitute or supplement to transit: An examination of travel-time differences in Toronto https://www.academia.edu/41409952/Measuring_when_Uber_behaves_as_a_substitute_or_supplement_to_transit_An_examination_of_travel_time_differences_in_Toronto

20. Technologies of Control, Communication, and Calculation: Taxi Drivers' Labour in the Platform Economy https://www.academia.edu/41109885/Technologies_of_Control_Communication_and_Calculation_Taxi_Drivers_Labour_in_the_Platform_Economy

21. Mobility-on-demand: An empirical study of internet-based ride-hailing adoption factors, travel characteristics and mode substitution effects https://www.academia.edu/42638295/Mobility_on_demand_An_empirical_study_of_internet_based_ride_hailing_adoption_factors_travel_characteristics_and_mode_substitution_effects

22. A Closer Look at the Sharing Economy, Global Ridesharing, and Ridesharing in Bangladesh https://www.academia.edu/41259733/A_Closer_Look_at_the_Sharing_Economy_Global_Ridesharing_and_Ridesharing_in_Bangladesh

23. RIDE SHARING SYSTEM: A REVIEW AND METHODOLOGY https://ijarse.com/images/fullpdf/1481036001_1315.pdf

24. Pricing and Efficiency in a Decentralized Ride-Hailing Platform https://www.semanticscholar.org/paper/Pricing-and-Efficiency-in-a-Decentralized-Platform*-Gaineddenova/e22a468a919bcd4c74ae25af90506ecbaaf39f2f

25. Optimal Pricing for ride-sourcing platform https://www.sciencedirect.com/science/article/abs/pii/S0377221719303819

26. https://www.researchgate.net/publication/348655240_The_Optimization_Model_of_Ride-Sharing_Route_for_Ride_Hailing_Considering_Both_System_Optimization_and_User_Fairness

27. Decentralized Ride Sharing and Vehicle-Pooling Based on Fair Cost sharing Mechanism https://www.semanticscholar.org/paper/Decentralized-Ride-Sharing-and-Vehicle-Pooling-on-Chau-Shen/41ce0cefc5fcf375889100c9800a1d479f01beed

28. Szabo, N. Smart Contracts: Building Blocks for Digital Markets. Extropy J. Transhumanist Thought 1996, 18, 28

29. Szabo, N. The Idea of Smart Contracts. Nakamoto Inst. 1997. Available online: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html (accessed on 3 April 2023).

30. Impact of Ride-Hailing Services on Transportation Mode Choices: Evidence from Traffic and Transit Ridership https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3244207

31. Bez, M.; Fornari, G.; Vardanega, T. The scalability challenge of ethereum: An initial quantitative analysis. In Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), San Francisco East Bay, CA, USA, 4–9 April 2019; pp. 167–176