Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Decentralized Ride Hailing platform based on Blockchain

Swati Bhisikar[1], Nilima Patil[1], Krishna Patil[1], Sahil Kuthe[1]
[1]*JSPM Rajarshi Shahu College of Engineering Tathawade*

ARTICLE INFO	ABSTRACT

Received: 29 Dec 2024

Revised: 15 Feb 2025

Accepted: 24 Feb 2025

Introduction:
The evolution of decentralized technologies has opened up new possibilities in transforming traditional, centrally managed services. Ride-hailing platforms today often face challenges like high intermediary costs, limited transparency, and data privacy issues. This project explores an innovative approach to ride-hailing by leveraging blockchain technology, specifically the Solana network, to offer a secure, transparent, and cost-effective alternative. The proposed system integrates smart contracts, real-time GPS tracking, and user-friendly interfaces to facilitate direct peer-to-peer interaction.

Objectives:
The primary objectives of this project are to eliminate intermediaries in ride-hailing services, enhance transaction transparency and user privacy through blockchain, reduce service fees, and improve system reliability. Additionally, the project aims to demonstrate the feasibility of a decentralized model in real-world urban mobility applications.

Methods:
The system architecture is designed using **Next.js** for the frontend interface and **Web3.js** to connect the frontend to the Solana blockchain. Smart contracts written in **Rust** are deployed on Solana to handle ride requests, fare calculations, and payment settlements. **Google Maps API** is integrated to enable route planning and live location tracking. The user interface ensures ease of interaction, while the backend smart contract logic facilitates secure, immutable, and automated operations. The platform promotes seamless booking, verification, and payment between drivers and riders without a centralized authority.

Results:
The decentralized system successfully demonstrated secure and efficient peer-to-peer ride booking. It reduced transaction costs by removing intermediary service providers and offered increased transparency in operations. Usability testing indicated smooth user interaction and ease of navigation through the Next.js interface. Scalability assessments showed that the system could handle multiple ride transactions concurrently with minimal delay, making it suitable for urban deployment. The financial model proved to be more beneficial for both riders and drivers compared to conventional platforms.

Conclusions:
This project highlights the potential of blockchain to revolutionize ride-hailing services by decentralizing control and empowering users. By utilizing Solana's high-speed blockchain and smart contracts, the system ensures secure, real-time, and low-cost operations. It enhances user trust, improves transaction fairness, and contributes to the growing ecosystem of decentralized applications. Future improvements may include broader integration with mobile platforms and expansion to multi-modal transport options.

Keywords: Blockchain, Ride-Hailing, Solana, Rust, Web3.js, Smart Contracts, Next.js, Google Maps API

INTRODUCTION

Ride-hailing platforms like Uber and Ola have significantly improved urban transportation by offering easy and accessible services through centralized mobile applications. However, these centralized systems come with several

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

limitations. Drivers are charged high commission fees, users have little control over how prices are set, and both riders and drivers face privacy concerns due to centralized data storage and processing.

Recently, blockchain technology has drawn interest as a possible remedy for these issues. It makes decentralized, transparent, and trustless systems possible, allowing users to communicate with one another directly without the need for middlemen.

We suggest a decentralized ride-hailing application based on the Solana blockchain to overcome the drawbacks of conventional ride-hailing systems. By eliminating the need for middlemen, this method aims to increase control for both drivers and passengers. The platform automates important procedures, including fare computation, payments, and ride confirmations, using Solana's Rust-based smart contracts. Solana is more suited for real-time applications like ride-hailing since it has quicker processing speeds and cheaper transaction fees than Ethereum-based solutions.

The user interface is created with Next.js, which helps deliver quick loading times and a responsive experience. To connect the frontend with the blockchain, Web3.js is used for tasks like linking digital wallets, verifying transactions, and accessing blockchain data.

This paper provides a detailed explanation of the system's architecture, development process, smart contract logic, and test results. Our goal is to show how decentralized technologies can help create more equitable, efficient, and privacy-friendly transportation solutions.

OBJECTIVES

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mi in nulla posuere sollicitudin aliquam. Egestas diam in arcu cursus. Tincidunt arcu non sodales neque. Id neque aliquam vestibulum morbi. Donec enim diam vulputate ut pharetra sit amet aliquam id. Enim sed faucibus turpis in eu mi bibendum neque egestas. Sed enim ut sem viverra. Donec ultrices tincidunt arcu non. Varius sit amet mattis vulputate enim nulla aliquet porttitor. Ultrices dui sapien eget mi proin sed libero enim. Sem viverra aliquet eget sit. Malesuada nunc vel risus commodo viverra maecenas accumsan lacus vel.

Quis risus sed vulputate odio ut enim. Laoreet suspendisse interdum consectetur libero id faucibus nisl. Egestas maecenas pharetra convallis posuere morbi. Vitae suscipit tellus mauris a diam maecenas. Sit amet cursus sit amet. Dui nunc mattis enim ut tellus. Amet nulla facilisi morbi tempus iaculis. A iaculis at erat pellentesque adipiscing commodo elit at imperdiet. Pulvinar mattis nunc sed blandit libero volutpat sed. Tincidunt ornare massa eget egestas purus viverra accumsan in nisl. Fermentum odio eu feugiat pretium. Tellus mauris a diam maecenas. Tincidunt lobortis feugiat vivamus at. Tincidunt tortor aliquam nulla facilisi cras. Enim neque volutpat ac tincidunt vitae. Amet massa vitae tortor condimentum. Ut tortor pretium viverra suspendisse potenti nullam ac tortor. Convallis aenean et tortor at.

METHODS

METHODOLOGY

This section outlines the step-by-step approach used in the design and development of the proposed decentralized ride-hailing application. The system is built using the Solana blockchain, smart contracts written in Rust, and Web3 technologies to ensure secure, transparent, and efficient interactions between drivers and riders. The development process is divided into five main stages: system architecture design, smart contract development, frontend integration, geolocation services, and final testing and deployment.

System Architecture Overview

Each of the four primary components of the system cooperate to give the user an uninterrupted experience.

The frontend is developed using Next.js, offering fast rendering and responsiveness. It enables users to interact with the application through features like ride requests, wallet connections, and viewing ride history.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

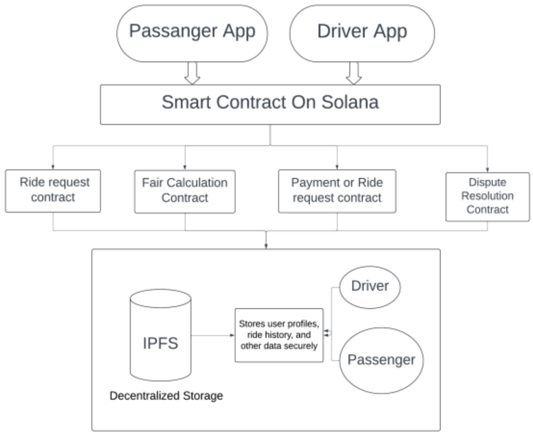
Research Article

Using the Anchor framework, a smart contract layer is implemented on the Solana blockchain and coded in Rust.It handles core operations, including registering users, managing ride bookings, locking fares, and processing payments automatically once a ride is completed.

Web3.js is used to link the frontend with the blockchain. It manages wallet logins, executes transactions, and listens for blockchain events, helping the frontend stay in sync with the backend processes.

The system also incorporates geolocation services using the Google Maps API. This provides real-time tracking, route mapping, and fare estimation based on travel distance. A visual overview of the architecture is shown in Figure 1 of the paper.

Smart Contract Development



The smart contracts are at the heart of the application and are written in Rust. They are deployed on the Solana Devnet and carry out several important tasks. These include starting trip requests, registering drivers and passengers using their public wallet addresses, and locking the fare fee according to the frontend's distance calculation. The payment is immediately sent from the rider to the driver by the smart contract when a ride is over. It also emits events to keep the front frontend updated. All contracts are carefully tested using the Solana CLI and Anchor's testing tools before being deployed.

Frontend Development

To provide a sleek and contemporary design, the application's frontend is constructed with Next.js as well as stylized with Tailwind CSS. It supports secure login using Phantom Wallet, allowing users to sign in without a password. The rider interface enables input of pickup and drop-off points, while the driver dashboard shows available ride requests, ride progress, and transaction history. A transaction tracker shows real-time updates of blockchain activity. Web3.js is used to connect the frontend with the smart contracts, allowing for secure transaction signing and data exchange.

Geolocation Integration

Google Maps JavaScript API is used to add geolocation functionality to the application. It helps drivers and passengers visualize pickup and drop-off locations by providing an interactive map. The frontend estimates the price using the shortest route and total journey distance determined by the API. This fare information is then sent to the smart contract to be locked before the ride begins.

Tools and Technologies Used

The entire system is built using a combination of modern tools and technologies. These include Next.js for frontend development, Rust and the Anchor framework for writing and deploying smart contracts, Solana Devnet for blockchain deployment, Web3.js for blockchain interactions, Google Maps API for geolocation services, and Tailwind CSS for styling the user interface.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Technology/Tool	Description
Next.js	Framework for building the frontend UI
Tailwind CSS	Styling and responsive layout
Web3.js	Blockchain connectivity and wallet integration
Phantom Wallet	Solana-based wallet for user authentication
Rust + Anchor	Smart contract development on Solana
Google Maps API	Geolocation, route planning, and distance estimation
Solana CLI & Devnet	Smart contract deployment and testing

Feature	Description
Decentralization	No centralized entity controls user data or pricing
Security	All payments and operations are executed via verifiable smart contracts
Transparency	Transaction history is viewable on-chain using Solana Explorer
Cost Efficiency	Solana's low gas fees (~0.000005 SOL) reduce platform charges [4], [12]
Privacy	User identity is protected through wallet-based authentication without KYC
Scalability	Solana's throughput (up to 65,000 TPS) ensures the system is scalable [4]

The idea put forth builds a safe, effective, and expandable platform by utilizing blockchain and contemporary web technology. Next.js is used as the main framework for building the frontend user interface, offering fast performance and server-side rendering capabilities. Tailwind CSS is integrated to ensure responsive design and consistent styling across devices. For blockchain connectivity and wallet integration, Web3.js is employed, allowing smooth interaction with the Solana network.

User authentication is handled through the Phantom Wallet, a popular Solana-based digital wallet. Smart contracts are built using Rust along with the Anchor framework, which offers a streamlined and effective approach to developing decentralized applications on the Solana blockchain. The use of the Google Maps API for features like distance estimate, route planning, and geolocation enhances the platform's usability. Solana CLI and Devnet are used for deploying and testing smart contracts in a development environment.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The system also benefits from key blockchain features. It supports decentralization by removing control from a single authority, giving users ownership over their data and transactions. Security is maintained through smart contracts, which automatically verify and execute operations. Transparency is maintained by making all transaction records accessible on the blockchain through tools like the Solana Explorer, allowing users to view and verify activity. The platform is cost-efficient due to Solana's low transaction fees, which significantly reduce the charges for users. Privacy is ensured by using wallet-based login methods that do not require Know Your Customer (KYC) procedures.

Lastly, Solana's architecture makes the system extremely scalable, enabling it to process as many as 65,000 transactions per second.

RESULTS

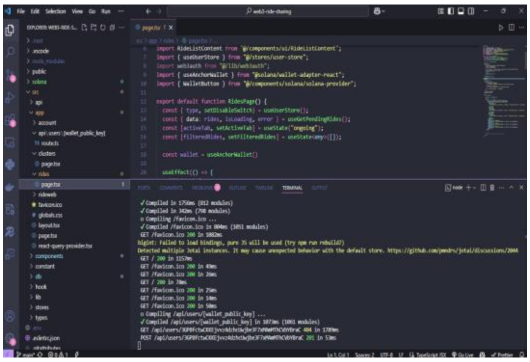


Fig 1. Project code

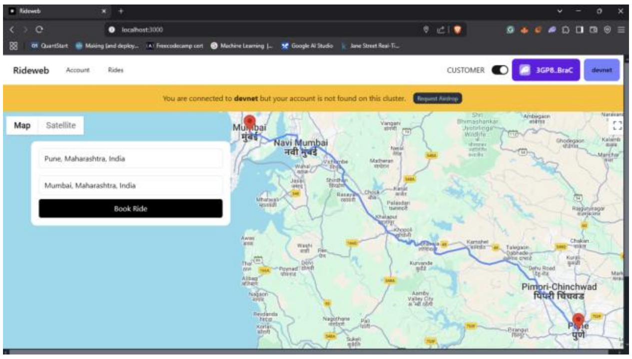


Fig. 2. Ride Booking by the user

The Fig 2 shows the map-based ride booking system, where the user selects pickup and destination locations. The system calculates the route between Pune and Mumbai and enables the user to proceed with the booking process.

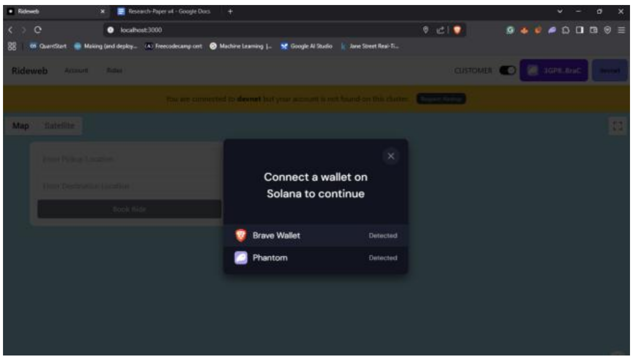


Fig. 3. Wallet Authentication options

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

The Fig. 3 displays the initial authentication step where the user is prompted to connect a cryptocurrency wallet (Brave or Phantom) to the Solana blockchain. This secure connection is essential for enabling decentralized identity and transaction verification on the platform

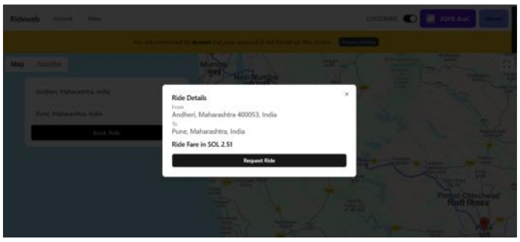


Fig. 4. Ride details

Upon selecting the ride details, the application estimates the fare in SOL (Solana's native cryptocurrency). This allows users to review fare details before confirming the ride request through the blockchain.

Fig. 5. Ride History

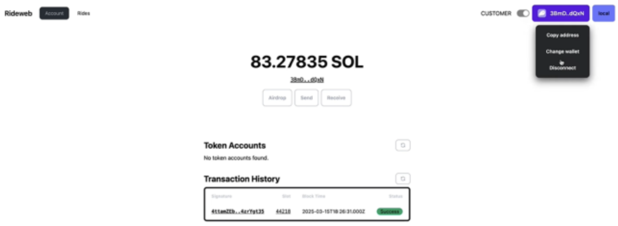


Fig. 6. Payment using SOL tokens

The account dashboard in Fig 7 displays the user's SOL balance and transaction history. It also provides features to request airdrops, send/receive tokens, and switch or disconnect wallets, ensuring transparent and secure asset management.

Fig 7. Driver Side

From the driver's dashboard (Fig), pending ride requests are listed with complete route details and fare. Drivers can accept the ride directly through the interface, triggering a decentralized transaction and service confirmation.

Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DISCUSSION

The envisioned decentralized ride-hailing platform lays a strong foundation for transforming urban mobility, yet there are numerous avenues for further development and exploration. One key direction for future work involves deploying the platform on the Solana Mainnet to evaluate its performance in real-world scenarios with active users. Incorporating decentralized identity (DID) solutions could allow the creation of verifiable and privacy-preserving user profiles, effectively removing the reliance on centralized KYC procedures. The efficiency and customer satisfaction of the platform might be significantly increased by utilizing AI-driven algorithms for driver and passenger pairing and route optimization. The introduction of a native utility token could foster an in-app economy that supports user incentives, governance, and loyalty programs. To ensure data integrity and transparency, decentralized storage solutions such as IPFS or Arweave can be employed for storing ride history, user feedback, and related metadata in a secure and tamper-proof manner. Expanding compatibility to include various blockchain networks may also boost interoperability and offer users more flexibility. Furthermore, building mobile applications with cross-platform technologies like React Native can help attract a wider user base and accelerate the widespread adoption of decentralized ride-hailing services.

REFRENCES

[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. 56–61.

[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. 1–6.

[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," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 663–680, Feb. 2022.

[4] A. Yakovenko, "Solana: A New Architecture for a High Performance Blockchain v0.8.13," *Whitepaper*, 2018. [Online]. Available: https://solana.com/solana-whitepaper.pdf

[5] Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," *2017 IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, 2017, pp. 557–564.

[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 Decentralized Blockchain Platform," *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA, 2021, pp. 3741–3750.

[7] N. Mahmoud, A. Aly and H. Abd Elkader, "Enhancing Blockchain-based Ride-Sharing Services using IPFS," *Intelligent Systems with Applications*, vol. 16, 2022, Art. no. 200135. [Online]. Available: https://doi.org/10.1016/j.iswa.2022.200135

[8] 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," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 2849–2866, Jul.–Aug. 2023.

[9] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[10] V. Buterin, "Ethereum White Paper," GitHub repository, 2013. [Online]. Available: https://ethereum.org/en/whitepaper/

[11] J. Smith and R. Brown, "The Potential of Blockchain for Transportation Systems," *Journal of Transportation Research*, vol. 45, no. 2, pp. 123–135, 2020.

[12] O. Mokalusi, R. Kuriakose and H. Vermaak, "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. 158–172.

[13] Y. Lu, et al., "Safety Warning! Decentralised and Automated Incentives for Disqualified Drivers Auditing in Ride-Hailing Services," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1748–1762, Mar. 2023.

[14] H. Yu, H. Zhang, X. Yu, X. Du and M. Guizani, "PGRide: Privacy-Preserving Group Ridesharing Matching in Online Ride Hailing Services," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5722–5735, Apr. 2021.

[15] N. Szabo, "Smart Contracts: Building Blocks for Digital Markets," *Extropy: The Journal of Transhumanist Thought*, no. 16, 1996, pp. 28–37.

[16] R. Gaineddenova, “Pricing and Efficiency in a Decentralized Ride-Hailing Platform,” *Working Paper*, 2021. [Online]. Available: https://doi.org/10.2139/ssrn.3740057

[17] M. Bez, G. Fornari and T. Vardanega, "The Scalability Challenge of Ethereum: An Initial Quantitative Analysis," *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, San Francisco East Bay, CA, USA, 2019, pp. 167–176.