2021 24th International Conference on Computer and Information Technology (ICCIT), 18-20 December, 2021.

Securing Ride-Sharing Service Using IPFS and Hyperledger Based on Private Blockchain

1st Md. Shajjed Hossan[1], 2nd Mst. Lelana Khatun[1], 3rd Shadia Rahman[1], 4th Saha Reno[1], 5th Mamun Ahmed[1]
Department of Computer Science and Engineering
Bangladesh Army International University of Science and Technology[1]
Cumilla, Bangladesh
sshajjedhussain@gmail.com, lelana2021@gmail.com, sadia.mim2110@gmail.com, reno.saha39@gmail.com, mamun@baiust.edu.bd

***Abstract*—When using the ride-sharing service, users may run across difficulties. For example, the driver may misconduct with them, have trouble hiring, verbalize abusively, or be a victim of harassment. So, the protection of the rider is extremely important in RSSs. We suggest a method to ensure protection that is based on private blockchain, in light of the current RSS's security concerns and centralism. Blockchain is a platform in which the stored transactions become immutable. Hyperledger is a blockchain that is better than public blockchain in almost every aspect. The benefits of the hyperledger is that the information which is stored in the hyperledger blockchain and can't be accessed outside the multiple organization. In our paper, we have secured the RSS by utilizing hyperledger predicated on a private blockchain system. If the users or riders faced any quandaries and they endeavored their issues via our hyperledger private blockchain, the organization could not modify or expunge the subsist report. In classical RSS, the ascendancy could modify or efface the report and they are able to expunge the riders ride information. To stop the unethical comfort, we have provided this hyperledger based private blockchain solution.**

Index Terms— Hyperledger, Blockchain, Smart Contract, IPFS, Consumer Rights, Ride-Sharing Security.

I. INTRODUCTION

Ride-Sharing Services is a System where passengers can book rides with the help of websites and mobile apps, with drivers of conveyances to hire them. If the rider confirms the ride then the driver picks them up in their personal conveyances or company's conveyances. Conventionally company conveyances have a company sticker or logo as an identifier and personal conveyances do not retain them. Driver's then drive them to their destination and take charges. Here the quandary is in our country harassment is a colossal issue. In a ride-sharing system, people face harassment and various types of problems. Later if any driver engenders any dilemma rider can file a licit charge on him but the quandary is if ascendancy wants, they can efface the repine. Therefore, we utilize blockchain here so that if someone engenders any quandary, they can file a licit charge on authority. The best part of a blockchain is that the company cannot abstract the repine. As we cannot update or delete blockchain's information, which denotes no one can expunge the information from this site which designates our data is secure here. In the papers [1] & [2] the authors recommended using a public blockchain to provide secure Ride Sharing Services. The researcher has implemented their proposed framework and deployed it using the Ethereum Blockchain. As they use a public blockchain, anyone can lick their information. Here nodes are not free thus, all transactions need to be paid. So, it is not user-friendly. The researcher of [3], proposed a secured ride sharing delivery system through blockchain, where the Blockchain's immutability and distributed architecture effectively prevent data tampering It is a consensus mechanism based on Byzantine Fault Tolerance (PBFT), suggested to improve the Blockchain, although all transactions in Bitcoin are public. So that anyone can visually perceive all the details and transactions of any address. The authors presented the passenger secure consortium blockchain based Ride-Sharing Services which guarantees the privacy of data interactions in a secured and confidential manner, using an attribute-based delegate re-encryption technique, in contrast to paper [4]. But as it is a multi-accessible blockchain, many people can access or see the details. In this paper, we propose a Securing Ride-Sharing Service using IPFS and Hyperledger based on private blockchain. Hyperledger is a private and access-controlled blockchain so that it is private for a person or organization. The reason behind the popularity of hyperledger is that the data stored inside the hyperledger cannot be accessed by everyone. It is limited to the particular organization or people and the access can be restricted using Hyperledger's access control functionality. One of the measure advantages of hyperledger is no transaction fee is required for any form of transaction can be carried out. Here is another point to note down that if the rider is the main culprit the driver can also get utilized by this system additionally. In [5] we find the security issue of ride-sharing services with a pragmatic approach. For exploring, evaluating, and building data structures, the researcher used the statistical technique of the quantitative data analysis method. The outcomes of this study are only from the passenger's point of view. But Driver' and other stakeholder's perspectives were not discussed in their accommodation. This denotes the accommodation is not utilizer-amicable but in our accommodation, we designate it as very auxiliary for innocents.

In RSSs, it is very paramount to ascertain the rider's safety. In light of the current RSS's security concerns and centralism, we propose a way to ascertain the security which is predicated on private blockchain.

978-1-6654-9435-9/21/$31.00 ©2021 IEEE

Ride requests, ride acceptance, ride completion, and ride payment all are stored as transactions in a ride share service. The entire ride will be captured on film. Which will be stored in IPFS. Since a hash is engendered for each file in IPFS so, if someone alters or deletes any data, the hash will be transmuted, and the fraud will be detected.

II. PROPOSED APPROACH

In our proposed Ride Sharing System there will be 3 kinds of transactions.

1. Create Asset ID.
2. Update Asset ID.
3. Report Asset.

When a passenger requests for a ride and the driver accepts it, the Ride Sharing System will be commenced, a report will be generated as a blockchain asset, corresponding to that particular ride. Each of the ride sharing reports will get identified by a unique Asset ID. Here the entire ride will be recorded on a webcam until user arrives at their destination. When the ride has consummated the details about when they arrived at their destination and the camera footage will be saved to IPFS. After being stored, IPFS will engender a hash for the video. Then the ride information and the hash will be updated in the user Asset ID. In this whole ride, if any user encounters any kind of abuse, he or she may file a complaint against the driver and it will additionally store in Asset id at report asset. In our system, we used our IPFS with hyperledger based blockchain protocol, therefor the admin cannot effaces any video or information.

III. RELATED WORKS

Many researchers have been conducting this particular sector. However, most of them are not satisfied due to some limitations. The author of [6] did a qualitative investigation of Uber, its mobility in Dhaka city and analyzed it just towards its driver and rider in terms of Marion Young's ethos of justice. The researcher used data from interviews with drivers and riders, as well as content analysis of riders' Facebook posts, to link technology interventions to policy through algorithms, objective analysis, and argument making. In This paper the researchers analyzed the mobility of Uber in Dhaka city and the injustice experienced by drivers and riders. But no technological terms are discussed here. Qadir and the other authors of [7] proposed a privacy-ensuring ride matching system for finding suitable partners for ride-sharing in RSSs. The researcher has developed their proposed framework using a spatial region-based selection mechanism. They have raised a selection technique, which is less privacy -preserving partner selection scheme that takes into account the trust levels of drivers and riders in RSSs. The authors of [8] suggested a Highest Aggregate Score Vehicular Recommendation (HASVR) framework, which suggests to the seeking passenger the vehicle with the highest aggregate score. The researcher has developed their proposed framework using HASVR and confirmed the usefulness of HASVR compared to subsisting schemes in decreasing the full mileage used to deliver all passengers, decreasing the passenger's climber, rising driver's profit and rising the percentage of gratified ride requests. In this paper, they have proposed a long scheduling technique. So, by introducing an optimization cadaster skill, a passenger gets the advantage of changing rides to reach as much closer to destination as possible. Hasan , Datta and Rahman in [9] proposed a new management system called vHike, that is convenient for dynamic ridesharing using smartphones and will eliminate the threat and social discomfort one feels by using ride sharing service. The researchers have developed their proposal that will work using the well-known technique of social network Web 2.0. In this paper they proposed a management system that will not use navigation of location data of both rider and user, it may create a problem assuming the time required to reach the user. No use of the block chain may create problems of security for the user. The author of [10] proposed a Blockchain-based solution of safe and reliable model of autonomous vehicle services and demonstrated the solution on a synthetic case study. The researcher has implemented their planned work using Hyperledger Fabric, a Blockchain architecture developed by the Linux Foundation and hosted on its servers. It is an enterprise blockchain which means this is designed for business application development but it does not have many skilled programmers. It has shown a lack of use cases. It has a complex architecture. It is not a Network fault-tolerant. The author of [11] outlines an ITS(intelligent transportation systems)-oriented, seven-layer theoretic model for blockchain, and on this cornerstone addresses the central research issues in Blockchain-based ITS (B2ITS). The purpose of this paper is to give a preliminary investigation into the emerging blockchain technology and its possible uses in transportation research. They're just getting started with blockchain technology, so B2ITS could take a long time to come to fruition. Yu and the other authors of [12] proposed passenger safety in Ride Sharing Services. The researcher has developed their proposed framework using dash cams in the rides and put it on a live transmission media and alarm system and also keeping indoor lights on during after-dark hours. In this paper, they have proposed an independent watchdog network system, which means at a time one volunteer can notice one or a few rides. So, If the riding system is too large then the watchdog system needs too many volunteers and it also contains a large cost. And on the other hand, the riders can depend on another person. So, it's also harmful. The authors of [13] introduced a privacy-ensuring online group ridesharing matching intended for ORH services (PGRide). The researcher has developed their proposed framework using an encrypted total space computation obtainment, combining homomorphic encryption with ciphertexts packing. In this paper, they have proposed homomorphic encryption, which still cascades concisely in the real world. Despite significant improvements over the years, it remains exceedingly slow and unresponsive, making it unsuitable for most corporate applications. One of the most major drawbacks of homomorphic encryption is that it requires either program updates or dedicated and specialized client-server applications to function properly. Organizations also can't use its methods to execute ad-hoc/discovery-based queries. In[14] they present BlockV, a blockchain qualified accomplishment to confirm the fairness of the ride which will be obtainable to all participants in the companion to companion network. The researcher demonstrated that this approach on the Ethereum platform is highly scalable, with a low transaction cost. The Ethereum blockchain is still undergoing lot of changes. So, It is not stable yet. As they did not make it private therefore anyone can see their work. Ethereum is not user friendly. The security issues inside the connected vehicle smart sensors, which can be hacked by professional intruders, is discussed in

this [15] study by suggesting a blockchain based framework. The researcher has developed their proposed framework using the Blockchain. In this paper, they did not make the blockchain private therefore anyone can see their work. In this [16] paper the author propose PEBERS (Practical Ethereum Blockchain-based Efficient Ride Hailing Service) is a system that shows how a consortia blockchain based decentralized system can be built to track every ride-sharing data. The researcher has developed their proposed framework using the Ethereum Blockchain. In this paper, they proposed a public blockchain named Ethereum. As it is not private anyone can licked their work. Here nodes are not free thus, all transactions need to be paid.

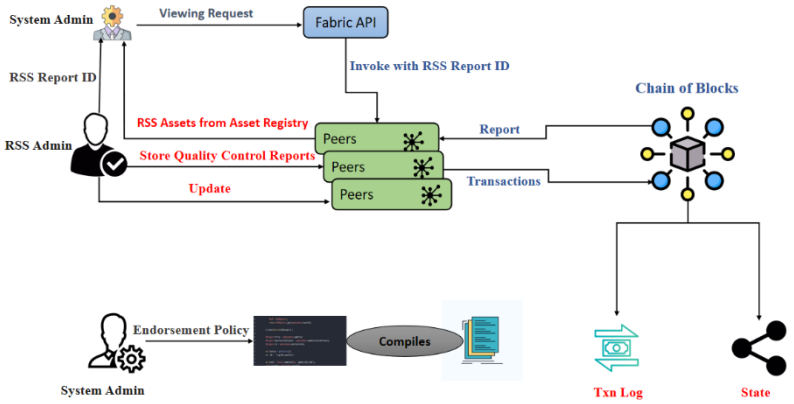

Figure 1: Information Flow Among the Peers and Interaction with the Ledger.

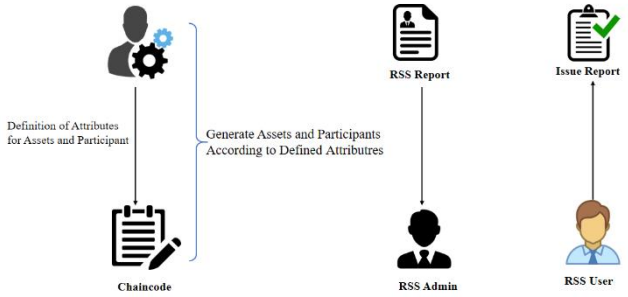

Figure 2: Utilization of Smart Contract for Defining Assets and Participants

IV. PROPOSED METHODOLOGY

Both the assets and creating, developing, or omitting these assets using the transactions are two essential aspects of our proposed framework, since our system's private blockchain is based on Hyperledger Fabric. In the next subsection, we will define how to create an asset, how to defines assets and transactions, how to execute a transaction, and how to use IPFS.

A. Defining The Assets

Our system stores information about a specific RSS report as an unique asset, which includes the RRS report ID, RRS report name, manufacture date, and other relevant data. All of this information is known to as asset definition. All of these assets are recorded and stored in our blockchain asset directory. Our system assets are associated by their RSS report ID, which is used to index them in the asset registry. To ensure the immutability of these attributes' histories, only transactions can be used to create and update information about them.

B. Execution of Transactions

Transactions are used to carry out every possible action in our proposed blockchain-based protocol, exactly like any other blockchain framework. If a transaction is saved in several of the chain's blocks, it becomes permanent, it can't be altered without a huge attack like the 51 percent attack, which is nearly hard to carry out. Which is basically impossible to execute. Among other things, our system has three main transactions: (i) Create RSS Report, (ii) Create Complain Report and (iii) Update RSS Report. The asset formation transaction description includes the same parameters as the asset definition. The transaction for deleting an asset, on the other hand, only requires the detection of the asset ID. A transaction ID is generated when a transaction for generating or removing an asset is completed, and this transaction ID may be used to quickly track the transaction from our private blockchain. Figure 2 shows how to create and delete asset transactions using the utility feature.

C. Identifying the Types of Participants

Accordingly the Hyperledger Fabric, any blockchains, including ours, requires users. In our suggested design, two categories of users are anticipated to run the process: (i) Administrator and (ii) Client. The roles of all of the contributors are discussed in the following segment.

1. Administrator: The system administrator can generate assets and obtain the history file (explained in the next segment) as well as the asset registry. Despite the fact that our

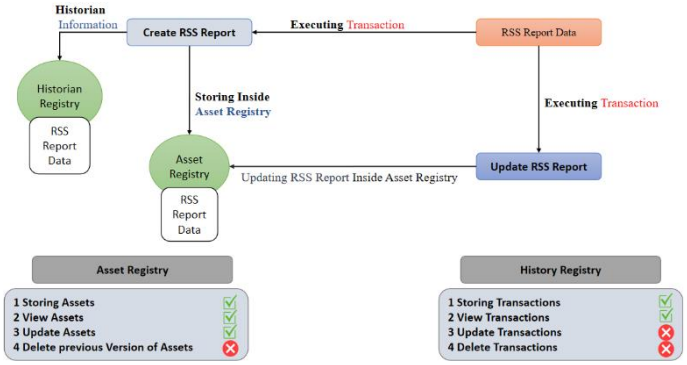

Figure 3: Interaction with Asset and Historian for Successful Processing of Transactions.

Transaction Executing Utility for RSS_Data Asset Creation and Updates

Execution of Transaction for Creating Report:

Input:AssetId. of the RSS_Data Asset, Name, Price, Gender,From, To, DestinationDistance, ACorNonAC, RideStartingTime, RideEndTime, License, Physical Harassment, Sexual Harassment, Abusive Language, IPFS.

```
RSS_DataRegistry ← Instance of getAssetRegistry(org.RSS.assetTxn.RSSreport)
factory ← Instance of getFactory()
NS ← org.RSS.assetTxn.RSSreport

RSSreport← Instance of factory.newResource(NS, 'RSS_Data', RSS_Data.IDNo)
det ← New Details Concept Creation Using factory.newConcept (NS, "Details")

det[3] ← [RSS_Data.Name, RSS_Data.price, RSS_Data.DestinationDistance,
RSS_Data.From, RSS_Data.To, RSS_Data.Gender, RSS_Data.Sexual Harassment,
RSS_Data.Physical Harassment]

RSSreport.RSS_Data← det

RSS_Data.From ← RSSreport.NewFrom
RSS_Data.To ← RSSreport.NewTo
RSS_Data.DestinationDistance ← RSSreport.NewDestinationDistance

event ← Instance of factory.newEvent(NS, " RSSreportCreated")
event.AssetId ← AssetId
emit(event)

return RSS_DataRegistry.add(RSS_Data)
```

Execution of Transaction for Update RSS_Data:

```
Input: ID no. of the RSS_Data Asset
RSS_DataRegistry ← Instance of
getAssetRegistry(org.RSS.assetTxn.RSSreport)

NS ← org.RSS.assetTxn.RSSreport

event ← Instance of factory.newEvent(NS, " update_RSS_issue_Ride")
event.AssetId ← AssetId
emit(event)

return RSS_DataRegistry.update(RSS_Data.AssetId)
```

Figure 4: Execution Methodology for Create-Update Transactions.

system's end users' default authorization range is restricted to simply finding and retrieving specific asset details and promptly removing the obtained asset, the system admin can ascend or descend the dominion. In addition, for better system administration, the system administrator has the authority to create more participant form now on. The covenant and obtainability of our Hyperledger-based blockchain are also managed by the administrator.

2. Client: A client can obtain asset details by confirming their RSS ID after becoming an active member in our system. After the asset details is retrieved from the asset register and the user ID is placed within the transaction, the transaction to remove the specific RSS information is conducted from the user end. End users do not have access to the asset information authorization. The system administrator has the authority to cancel active users' memberships if any rules are breached.

D. Reserve Assets Inside Asset Directories

The system's asset registry contains all of the asset information. If a system administrator or a user wants to get information about a particular asset, they must enter the asset ID into this register. Using our system's querying mechanism, we may search this registry for relevant RSS information. To maintain the immutability of blockchain, The complaint is not completely abstracted when a user abstracts it; instead, it is stored in the state and transaction.

E. Transactions in the Historian's Record

All transactions on our private blockchain are logged in blocks, which may be accessed using the historian. The history record usually contains the transaction ID, the user ID who completed the transaction, and the transaction type. Our technology allows for several sorts of transactions, which are then categorized in the historical record. Hyperledger Fabric allows queries in the same way as a standard database does. We will use SQL for our blockchain, As condition in the SQL we will use transaction related information like- transaction type, transaction user, transaction time etc. for fetching data from SQL history. Figure 4 shows the process for accessing the asset register, creating a historical record, and verifying authenticity.

F. Creating Permissions Regulations

Access to system resources such as business models, transaction records, asset records, and transactions can be restricted using the Hyperledger Fabric Access Control function. With the exception of a few, program managers have access to a wide range of program resources. The admin does not have the power to delete RSS data, but end users do. Data from the asset directory and the historical document can also be retrieved by end users. The end users are not provided with the access to the transaction responsible for changing the hash of a specific piece of evidence received from IPFS inside the RSS Report, so that no one can accuse a rider for something he or she hasn't done on the blockchain. Figure 5 depicts how we deployed Hyperledger Fabric's Access Control functionality in our application.

Table I: Information on Random Rides for System Evaluation Purpose

Source: Pathao, Bangladesh.

Name	Gender	From	Destination	Distance	Price	Starting Time	Ending Time	Vehicle License
Natasha	Female	Sylhet	Cumilla	175km	300BDT	9.00 AM	3.15PM	198k98
Tishi	Female	Kandirpar, Cumilla	Cantonment, Cumilla	25km	30BDT	1.00 PM	2.45PM	75cd22
Shohag	Male	Jindabazar, Sylhet	Noyashorok, Sylhet	50km	100BDT	7.00 AM	10:05AM	1425c2
Mim	Female	Bisnakandi,Sylhet	Lalakhal, Sylhet	125km	200BDT	4:25 PM	7:35PM	12364c
Arif	Male	Dhanmondi,Dhaka	Mirpur 12, Dhaka	30km	70BDT	12:00PM	4:10PM	162rk2

Validating RSS by Accessing Transaction and Asset Registries, Participant creation, Assigning Identities and Providing The Access to Execute Transactions

Query AccessHistorianRecord:
description ← "Returns Historian Records"
statement ← SELECT org.hyperledger.composer.system.HistorianRecord where (transactionType == _$input)
Query AccessAssetRegistry:
description ← "Returns Asset Registry Elements"
statement ← SELECT org.hyperledger.composer.system.AssetRegistry where (assetID == _$input)

accessSpecificAsset:
bnUtil ← Business Network Definition of Ride Sharing System
assReg ← bnUtil.connection.query('AccessAssetRegistry')
qry ← bnUtil.connection.buildQuery(assReg)
return bnUtil.connection.query(qry,{input: assReg#ID)

accessUpdateIssueAssetTransaction:
bnUtil ← Business Network Definition of Ride Sharing System
histRees ← bnUtil.connection.query('AccessHistorianRecord')
qry ← bnUtil.connection.buildQuery(histRees)
return bnUtil.connection.query(qry,{input: 'update_RSS_issue_Ride')

Assigning Identities to End-Use Participant:
participantInstance ← composer.participant.add(class: EndUser, participantKey:1234, fname: Saha, lname: Reno, contactNum: 01521234214, email: saha.reno@baiust.edu.bd)

composer.identity.issue(participantInstance, admin@Ride-Sharing-System)

Defining Access Control for System Admin and End Users:
Rule AdminSystemResource:
description ← "Grant business network administrators full access to system resources"
participant ← "org.hyperledger.composer.system.NetworkAdmin"
operation ← ALL
resource ← "org.hyperledger.composer.system.**"
action ← ALLOW

Rule EndUserPermissionHistorianRecordREAD_CREATE:
description ← "Can VIEW and CREATE HistorianRecord inside the Historian"
participant ← "org.RSS.assetTxn.RSSreport.EndUser"
operation ← READ, CREATE
resource ← "org.hyperledger.composer.system.HistorianRecord"
action ← ALLOW

Rule EndUserPermissionHAssetRegistryREAD:
description ← "Can only VIEW Assets inside the AssetRegistry"
participant ← "org.RSS.assetTxn.RSSreport.EndUser"
operation ← READ
resource ← "org.hyperledger.composer.system.AssetRegistry"
action ←ALLOW

Figure 5: Asset directory, Historian Record and Access Control Querying Methods

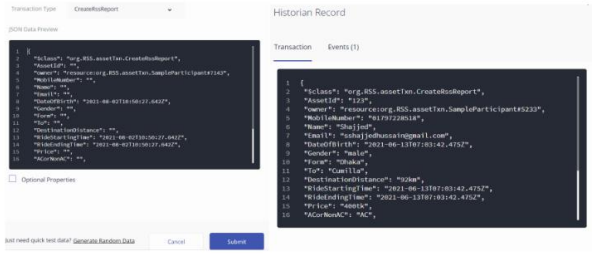

Figure 6: Accessing from the Historian Record and executing the CreateAsset transaction.

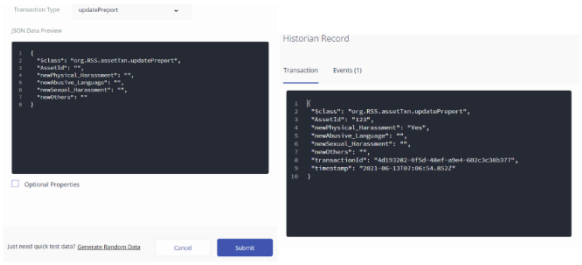

Figure 7: Accessing from the Historian Record and executing the UpdateIssueReport transaction.

V. RESULT ANALYSIS

To test our system, we obtained and used boarding data from Pathao, one of the most popular transit companies for rental companies. Figure 1 shows an example of ride details. This first-pass data for Table 1 was generated by the import transaction and recorded on the blockchain using admin sanctions.

To test our system, we obtained and used boarding data from Pathao, one of the most popular transit companies for rental companies. Figure 1 shows an example of ride details. This first-pass data for Table 1 was generated by the import transaction and recorded on the blockchain using admin sanctions. To test our system, we obtained and used boarding data from Pathao, one of the most popular transit companies for rental companies. Figure 1 shows an example of ride details. This first-pass data for Table 1 was generated by the import transaction and recorded on the blockchain using admin sanctions. Fig.5 shows that the AssetCreation activity is terminated and the asset ID is used for active access.

After receiving transaction details on the first trip, our blockchain system peer-to-peer transaction, and this transaction data is provided to the user, who confirms the second transaction.

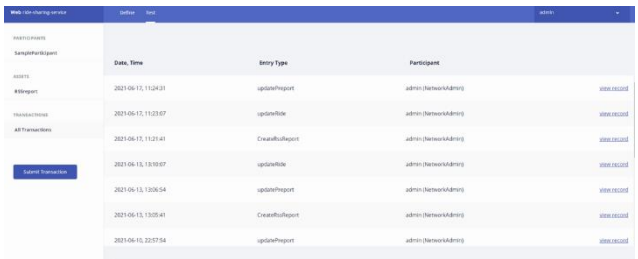

Figure 8: Indexed Historian Record

Table II: Time Required to Perform Transactions in Various Blockchain Technologies.

Number of Transactions	Time for Hyperledger	Time for Ethereum	Time for Bitcoin
100 Tran	0.35seconds	11.25 seconds	13.3 seconds
80 Tran	0.3seconds	9.45 seconds	11.43 seconds
60 Tran	0.25seconds	7.26 seconds	9.5 seconds
40 Tran	0.15seconds	5.35 seconds	7.5 seconds
20 Tran	0.065seconds	2.33 seconds	4.66 seconds

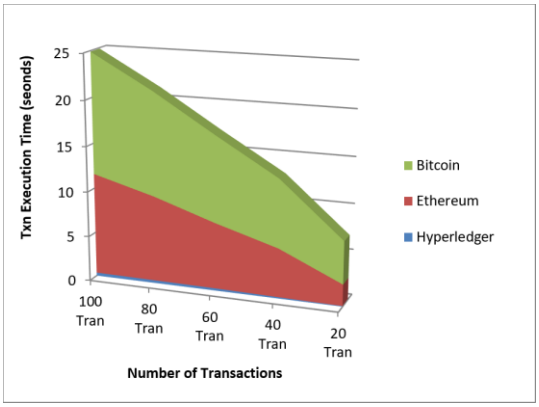

Figure 9: Correlation of the Time Complexity of Three Blockchain Systems.

As shown in Figure 7, our system has a historian record list that may be used to retrieve specific assets and transaction info. We created some random ride data and analyzed the time it took to enter it into the private blockchains (Bitcoin and Ethereum) to test the efficiency of the Hyperledger predicated ride shearing mechanism.Hyperledger enables quicker information storage and ride detection, as seen in Table 2 and the graph in Fig. 9.

VI. CONCLUSION

The Hyperledger Blockchain's permissioned and private structure makes ridesharing verification significantly more accurate and dependable than already existing public and sanction-less blockchain-based systems. Ethereum needs crypto currency to complete any transaction and transaction data is public, thus making it incompatible with the construction of a ride-sharing system. Hyperledger Fabric is a distributed ledger technology. The Hyperledger-based sanctioned network uses the decentralized network's historian record to ensure that a recorded transaction cannot be edited by anyone. Members are also subjected to restrictions depending on their responsibilities in the system, which is impossible to achieve without sanctions in a Blockchain system.

REFERENCES

[1] M. Baza, M. Mahmoud, G. Srivastava, W. Alasmary, and M. Younis, "A Light Blockchain-Powered Privacy-Preserving Organization Scheme for Ride Sharing Services," in 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1–6. doi: 10.1109/VTC2020-Spring48590.2020.9129197.

[2] M. Baza, N. Lasla, M. Mahmoud, G. Srivastava, and M. Abdallah, "B-Ride: Ride Sharing with Privacy-preservation, Trust and Fair Payment atop Public Blockchain," IEEE Transactions on Network Science and Engineering, pp. 1–1, 2019, doi: 10.1109/TNSE.2019.2959230.

[3] X. Zhang, J. Liu, Y. Li, Q. Cui, X. Tao, and R. P. Liu, "Blockchain Based Secure Package Delivery via Ridesharing," in 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), 2019, pp. 1–6. doi: 10.1109/WCSP.2019.8927952.

[4] D. Wang and X. Zhang, "Secure Ride-Sharing Services Based on a Consortium Blockchain," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2976–2991, 2021, doi: 10.1109/JIOT.2020.3023920.

[5] S. T. Cynthia, M. Majumder, A. Tabassum, N. N. Khanom, R. Amin Tuhin, and A. K. Das, "Security Concerns of Ridesharing Services in Bangladesh," in 2019 2nd International Conference on Applied Information Technology and Innovation (ICAITI), 2019, pp. 44–50. doi: 10.1109/ICAITI48442.2019.8982128.

[6] N. Kumar, N. Jafarinaimi, and M. Bin Morshed, "Uber in Bangladesh: The Tangled Web of Mobility and Justice," Proc. ACM Hum.-Comput. Interact., vol. 2, no. CSCW, Nov. 2018, doi: 10.1145/3274367.

[7] Y. He, J. Ni, X. Wang, B. Niu, F. Li, and X. Shen, "Privacy-Preserving Partner Selection for Ride-Sharing Services," IEEE Transactions on Vehicular Technology, vol. 67, no. 7, pp. 5994–6005, 2018, doi: 10.1109/TVT.2018.2809039.

[8] H. Qadir, O. Khalid, M. U. S. Khan, A. U. R. Khan, and R. Nawaz, "An Optimal Ride Sharing Recommendation Framework for Carpooling Services," IEEE Access, vol. 6, pp. 62296–62313, 2018, doi: 10.1109/ACCESS.2018.2876595.

[9] C. Stach, "Saving time, money and the environment - vHike a dynamic ride-sharing service for mobile devices," in 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011, pp. 352–355. doi: 10.1109/PERCOMW.2011.5766904.

[10] M. G. M. Mehedi Hasan, A. Datta, and M. A. Rahman, "Poster Abstract: Chained of Things: A Secure and Dependable Design of Autonomous Vehicle Services," in 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), 2018, pp. 298–299. doi: 10.1109/IoTDI.2018.00048.

[11] Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, pp. 2663–2668. doi: 10.1109/ITSC.2016.7795984.

[12] B. Chaudhry, A.-U.-H. Yasar, S. El-Amine, and E. Shakshuki, "Passenger Safety in Ride-Sharing Services," Procedia Computer Science, vol. 130, pp. 1044–1050, Jan. 2018, doi: 10.1016/j.procs.2018.04.146.

[13] 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, pp. 1–1, 2020, doi: 10.1109/JIOT.2020.3030274.

[14] P. Pal and S. Ruj, "BlockV: A Blockchain Enabled Peer-Peer Ride Sharing Service," in 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 463–468. doi: 10.1109/Blockchain.2019.00070.

[15] G. Rathee, A. Sharma, R. Iqbal, M. Aloqaily, N. Jaglan, and R. Kumar, "A Blockchain Framework for Securing Connected and Autonomous Vehicles," Sensors, vol. 19, no. 14, 2019, doi: 10.3390/s19143165.

[16] S. Kudva, R. Norderhaug, S. Badsha, S. Sengupta, and A. S. M. Kayes, "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. doi: 10.1109/ICIoT48696.2020.9089473.