Chained of Things: A Secure and Dependable Design of Autonomous Vehicle Services

MGM Mehedi Hasan*, Amarjit Datta*, Mohammad Ashiqur Rahman*, and Hossain Shahriar†
*Department of Computer Science, Tennessee Tech University, USA
†Department of Information Technology, Kennesaw State University, USA
Emails: {mmehediha42, adatta42}@students.tntech.edu, marahman@tntech.edu, hshahria@kennesaw.edu

***Abstract*—Car manufacturers are investing heavily in Autonomous Vehicles (AVs) because of their increasing popularity. AVs employ a combination of high-tech sensors and innovative algorithms to detect and respond to their surroundings, including radar, laser light, GPS, odometer, drive-by-wire control systems, and computer vision. When a user requests for a ride share or rent, the responding AV/IoT device does not have the sufficient capability to store, process, or verify each other. Currently, every AV/IoT device typically needs to rely on a trusted third party, which invokes another trust issue. If the trusted third party is rogue or loses credibility, the whole system will collapse. This paper introduces Chained of Things Framework and develops a blockchain-based secure ride-sharing service between passengers and autonomous vehicles. We implement an initial prototype and provide our initial results based on a proposed protocol.**

***Index Terms*—Ride-sharing, Autonomous Vehicle, Security, Blockchain.**

I. Introduction

Autonomous Vehicles (AVs) are the next generation vehicles, which are equipped with the advanced sensing and communication capabilities, navigation devices, computer vision technology to drive autonomously with limited or no human intervention. Since most of the car crashes are the result of human errors and distractions [1], a computer would be an ideal driver as it can use complicated algorithms to determine appropriate driving measures. Therefore, a significant cost saving is expected, especially costs with respect to the insurance and accident recovery [2]. When a computer takes over the driving responsibility, the driver can use the time to do some other fruitful things. Disabled individuals, who have to rely on assistance from others to get around, will be benefited from self-driving cars with new freedom and enhanced mobility. For larger cities that are plagued with inadequate public transportation and high traffic congestion, self-driving cars can provide a practical solution [3].

As more and more AVs will be used in practice, there will be need for ride-sharing services to alleviate these urban problems by increasing vehicle occupancy. Ride-sharing offers a platform of increased transportation options available to consumers and businesses, which can significantly increase consumer welfare. A user will communicate an AV and upon the fulfillment of certain criteria/requirements and price agreement, the consumer will get the ride. Ride-sharing platforms connect the AVs with the consumers through a network. A consumer uses a smart device (e.g., mobile phone) to request a ride with necessary information and requirements (e.g., pick-up point, time, drop-off point). The platform will let the ride providers (AVs) know about the ride request, which can be satisfied by an AV when it finds it feasible to provide the requested service.

While AVs open a new era for ride-sharing services, there are many trust issues associated with the benefits. The ride-sharing using AVs depends on the communication among the consumers and the AVs. The AVs and IoT devices involved in this system need to be authenticated and trusted to one another. A trusted third party-based mechanism can provide a solution but there is still the concern of trust, along with the issues like privacy and single point of failure. There is a greater need for a non-centralized and third party-less framework that can provide security-ensured, privacy-preserved, and trustworthy AV-based ride-share services.

Blockchain is a distributed database that maintains a continuously growing chain of data records. There is no central computer holding the entire chain rather each participating node have a copy of it. Blockchain has already been proven to be effective, which is currently being used by Bitcoin, a worldwide cryptocurrency and digital payment system, as an underlying technology [4], [5]. In this paper, we address the problem of secure ride-sharing services between passengers and AVs by proposing a Chained of Things (CoT) framework. We implement a prototype of the proposed CoT architecture and demonstrate case studies on feasibility.

The rest of this paper is organized as follows: In Section II, we provide background and related work. Section III discusses the proposed protocol design. In Section IV, we discuss implementation and provide example case studies. Finally, we conclude the paper in Section V.

II. Background and Related Work

A. Blockchain

Blockchain is a database that maintains a continuously growing set of data records. It is distributed in nature, meaning that there is no master computer holding the entire chain. Rather, the participating nodes have a copy of the chain. It is ever growing, so records are only added (rather deleted) to the chain. A Blockchain consists of two types of elements. (a) Transactions: These are the actions created by the participants in the system. (b) Blocks: Blocks record transactions and make sure they are in the correct sequence and tamper free.

Blocks record time stamps when the transactions are added. The big advantage of Blockchain is that it is public. Everyone participating in the Blockchain can see the blocks and the transactions stored in them. This does not mean everyone can see the actual content of the transaction. The private key protects the content of the transaction. A Blockchain is decentralized, so there is no single authority that can approve the transactions or set specific rules to have the transactions accepted. Blockchain is secure because the database is immutable [6]. The database can only be extended and previous records cannot be changed.

When someone wants to add a transaction to the chain, all the participants in the network will validate it [7]. They do this by applying an algorithm to the transaction to verify its validity. The criteria of a valid transaction are defined by the Blockchain system and can differ between systems. Then it is up to a majority of the participants to agree that the transaction is valid. A set of approved transactions is then bundled in a block, which are sent to all the nodes in the network. They in turn, validate the new block. Each successive block contains a hash, which is a unique fingerprint, of the previous block. In a public Blockchain, everyone can read or write data. Some public Blockchains limit the access to just reading or writing. Bitcoin [4], for example, uses an approach where anyone can write. In a private Blockchain, all the participants are known and trusted. This is useful when the Blockchain is used between companies that belong to the same legal mother entity.

The ledger is tamper-proof and cannot be manipulated by malicious actors because it does not exist in any single location. The man-in-the-middle attacks [8] cannot be staged because there is no single thread of communication that can be intercepted. Blockchain makes trustless, peer-to-peer messaging possible and has already proven its worth in the world of financial services through cryptocurrencies such as, Bitcoin and Ethereum [9] providing guaranteed peer-to-peer payment services without the need for third-party brokers.

B. Related Work

Kamali *et al.* work on vehicle platooning in which each agent captures the autonomous decisions carried out by each vehicle [10]. They use formal verification to ensure that these autonomous decision-making agents in vehicle platoons never violate safety requirements. Mladenovic *et al.* work on a self-organizing and cooperative control framework for distributed vehicle intelligence [11]. Hu *et al.* propose a lane changing maneuver to balance the trade-off between efficiency and safety in AVs [12]. Petrov and Nashashibi develop a feedback controller for autonomous overtaking without utilizing roadway marking and inter-vehicle communication [13]. Li *et al.* present a multi-level fusion-based road detection system for driverless vehicle navigation to ensure safety in various road conditions [14].

Alsabaan *et al.* utilizes traffic light signals and vehicle-to-vehicle (V2V) communications to help vehicles adapt their speeds and avoid unnecessary stop, acceleration, and excessive speed [15]. Gomes *et al.* design a driver-assistance system, which allows a vehicle to collect real-time camera images from other vehicles in the neighborhood over V2V communications [16]. Ma *et al.* proposes a taxi ride-sharing system called T-Share, where the dynamic taxi ride-sharing problem was studied [17]. Kosba *et al.* present a decentralized smart contract system that does not store financial transactions in the clear on the Blockchain, thus retaining transactional privacy from the publics view [18]. To formally define and reason about the security of our protocols, they formalize the Blockchain model of cryptography.

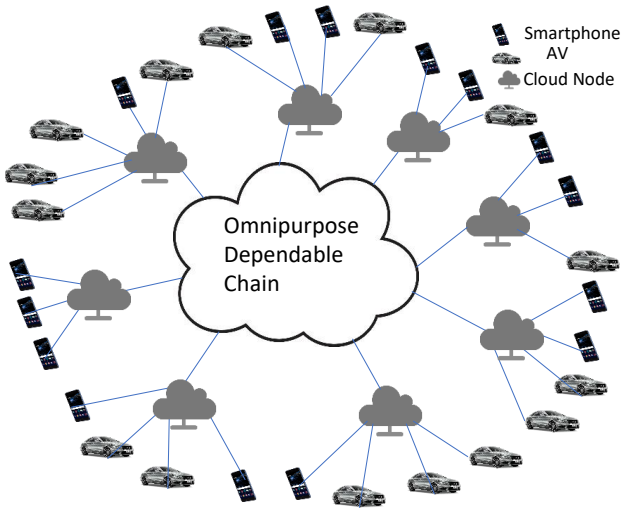

Fig. 1. Every Smart device is connected to a VM of a CSP.

Dorri *et al.* claim that their proposed BC-based smart home framework is secure by thoroughly analyzing its security with respect to the fundamental security goals of confidentiality, integrity, and availability [19].They present simulation results to highlight that the overheads (in terms of traffic, processing time and energy consumption) introduced by their approach are insignificant relative to its security and privacy gains. Christidis *et al.* describe how a Blockchain-IoT combination can facilitate the sharing of services and resources leading to the creation of a marketplace of services between devices [20]. Hardjono *et al.* proposed Cloud-Based Commissioning of Constrained Devices using Permissioned Blockchains [21].

None of the work discusses how the concept of the Blockchain can be exploited in providing AV services. In this work, we propose a Blockchain based platform that can provide an autonomic, secure, and dependable AV services.

III. Protocol Design Of the ODC

A. Proposed ODC Model

We consider a private Blockchain, where a Cloud Service Provider (CSP) [22] manages a Blockchain network. We name this Blockchain Omnipurpose Dependable Chain (ODC). The CSP only manages the Blockchain network, not its operations, deploying necessary virtual machines (VMs) based on the number of users, *i.e.*, ride-sharing service providers and clients. These VMs act as the Blockchain (BC) nodes in the ODC. That

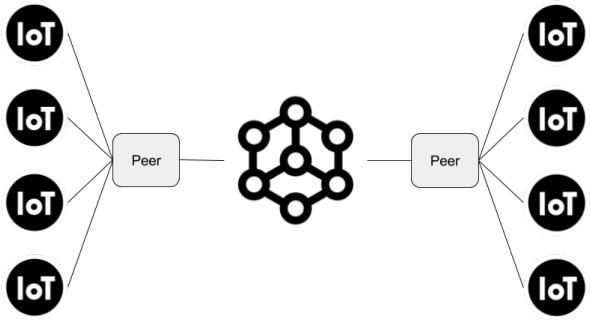

Fig. 2. AV ride-sharing Blockchain network

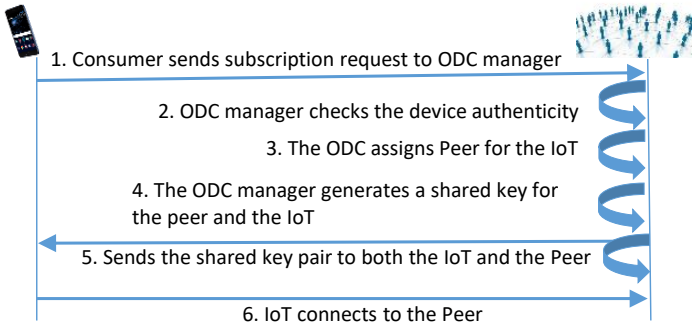

Fig. 3. Subscription request protocol to the ODC.

is, the BC nodes are VMs in a cloud that provides Blockchain as a Service (BaaS). These BC nodes are often named as peers in a Blockchain network. Some or all of the VMs work as the miner nodes [23]. Each IoT device is connected to an arbitrary BC node. We consider the limited capability of many IoT devices, and therefore we assume a participating entity cannot be a BC node itself, rather it is connected to (or subscribed to) a BC node. The BC node does necessary computations for the connected IoT nodes. All the IoT devices communicate among themselves through the representative BC nodes.

Fig. 1 shows a simplified diagram of how the IoT devices will be connected to a Blockchain network. Here, ODC is the Blockchain, which has got several nodes. IoT devices like smartphones, AVs, etc. get connected with any of the BC nodes, which are known as the corresponding node. The IoT devices that are connected to the same corresponding node are called sibling nodes. Different BC nodes might have a different number of sibling nodes. The total number of nodes required may vary depending on the number of IoTs.

Any IoT can connect to any corresponding BC node at any time. A BC node can accommodate a certain number of IoTs at a given time. There are barriers on what can be the maximum number of IoTs a BC node can serve. Running a BC node incurs cost so the ODC service provider will try to keep an optimum number of BC nodes at a given time. The ODC service provider (*i.e.*, the CSP) also needs to make sure that a certain number of miners are always available for mining. This will make sure that the average time to mine a new block does not cross the threshold.

B. Entities

Our proposed ODC framework has the following entities:

- *Blockchain Network:* BC nodes or peers are connected with the Blockchain network. In the AV ride-sharing application, there can be more than one peer. When a new transaction takes place, Blockchain peers share the transaction information with each other using the Blockchain network. In Fig. 2, we can observe a typical AV ride-share Blockchain network. In this case, each IoT device is connected with only one peer. There can be more than one peer. Based on the capability (computational resources) of the peers, different peers may have different number of connected IoT devices.
- *BC Node and Peer:* Every machine that has the minimum capability of maintaining a Blockchain ledger and is directly connected to the ODC is considered as BC node. In this case, a BC node can be a VM of the CSP that is facilitating the ODC. A Blockchain is typically managed by a peer-to-peer network. A User connects to the network through a peer node. We use the terms BC Node and peer interchangeably.
- *Miner Node:* Miner nodes are special purpose nodes that are responsible for adding blocks in the Blockchain after verifying transactions.
- *IoT Devices:* In this framework, every smart device (*e.g.*, smartphones, AVs, etc) is a IoT device. Each IoT device is connected with a peer node through a communication network. The communication network between the IoT devices and the peer is encrypted. An user can log into its IoT device, connect with the peer, and initiate request or response.

C. Subscription to Service

We consider a user as consumer when the user takes ride-sharing service. Similarly, when a user provides ride-sharing service, we consider it as provider. In this work, we use the terms provider and AV interchangeably. When a user wants to subscribe for the ride-sharing services, it creates its user profile. In its profile, the user mentions its identity, vehicle information, and the payment detail. Using standard authentication mechanism, an IoT device is connected with the Blockchain peer.

Fig. 3 shows how an IoT subscribes to a peer of the ODC network. In the ODC network, there will be peer who will work as an ODC manager. The ODC manager will be responsible for generating cryptographic keys. The ODC manager is randomly selected from the existing peers for a particular time period. In this way, every peer has the equal probability of getting selected as the ODC manager, which ensures fairness to the peers. When an ODC manager is selected, a secondary ODC manager is also selected. The secondary ODC manager will get

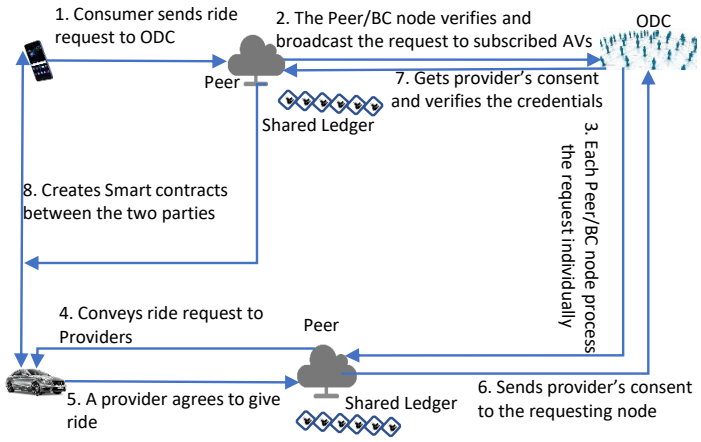

Fig. 4. Ride-sharing request-response protocol.

the chance to act as the ODC manager only when the primary ODC manager fails within the particular time frame.

The selection of secondary ODC manager ensures that there will not be any single point of failure. If an IoT device wants to subscribe to the ODC network, it will send the subscription request to the ODC manager. The ODC manager will check the authenticity of the IoT with the help of device information (*i.e.*, ESN (Electronic Serial Number), IMEI (International Mobile Equipment Identity), MEID (Mobile Equipment Identifier), etc). If the authentication is successful, the ODC manager will generate a shared key. The key will be shared between the IoT and a peer to which the IoT will be subscribed. Finally, the IoT will connect to the assigned peer using the shared key.

D. Ride-Share Request Response

When a consumer X needs ride, it makes ride-sharing request and shares the following information:

- Pick-up and drop-off points of the ride
- Earliest pick-up time and latest drop-off time.
- Route preference (optional).

This ride-sharing request is visible to all providers (AVs). If a provider Y wants to respond to this request, it can share its intent to X. The provider Y may choose to respond to a ride-sharing request based on following conditions:

- Route of the consumer: If the travel route of Y matches with the route of X, then Y may choose to provide ride-sharing service.
- Reputation of user X: As an user uses ride-sharing application, it builds its reputation through its transactions. A ride-sharing user may get positive or negative review from the other users based on its behavior. In this work, we consider a ranking mechanism whether a user is ranked in a scale of 0 to 5. When a user has better rank, it means that the user is more trustworthy. An AV can choose an user X for ride-sharing based on its score.

In AV ride-sharing service, multiple communications are performed between the service requester and service provider.

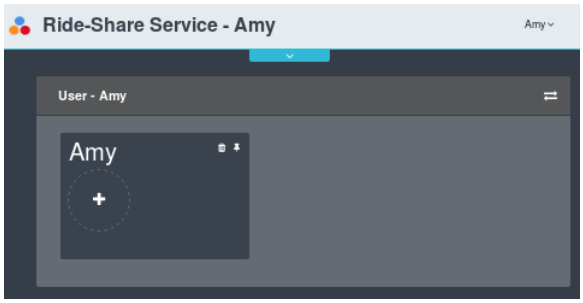

Fig. 5. Amy's interface.

Security is very important for the operation of the AV services. Here, we list the security constraints:

- When a user X requests for a ride-sharing service, the Blockchain peer node authenticates the user using its credential. Only authorized users are allowed to make a ride-sharing requests.
- To make a ride-sharing request for a particular travel route, an user must have sufficient fund.
- When a user X initiates a ride-sharing request, other AVs can respond to that request. An AV v_i can only respond to the ride-sharing request if the AV is authenticated by the Blockchain peer node.
- When an AV v_i accepts ride-sharing request of X, multiple communication happen between the users. User X acknowledges user v_i's response. When the AV v_i reaches to user x, the user x must hop in to the AV v_i. AV v_i must drop user X in the appropriate location. Finally, the transaction between user x and AV v_i must complete successfully once the ride-sharing is complete.

Fig. 4 shows the request-response protocol for the ride-sharing service. Here, a consumer sends a ride-sharing request to the ODC network via his/her smartphone. The corresponding peer node, to which the consumer is subscribed to, verifies the request and broadcast this to the ODC network. At this point, all the AVs of the ODC network can see the request. The willing AVs will make their interest known to the corresponding peer. The corresponding peer, in turn, will inform the requesting peers about the ride-sharing intention of its subscribed AV. The requesting peer will verify and inform the consumer about an AVs ride-sharing intention. When the consumer agrees with the AV, the ride-sharing request will complete the cycle.

E. Exceptional Scenario

There might be cases when one of the parties fails to fulfill the commitment. This can take place from either of the parties. we discuss below how our architecture can handle such situations.

1) Break of Commitment by the Consumer: A consumer Amy asks for a ride and a provider AV1 agrees to give ride but later Amy decides not to go. This can be considered as a violation of a promise. Because when both the parties agree for a service, this is considered a commitment. Any kind of break

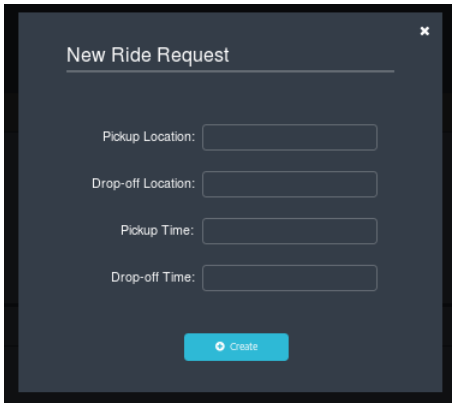

Fig. 6. Amy is making a ride-sharing request.

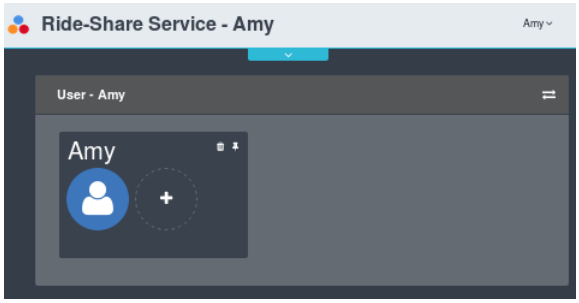

Fig. 7. Amy's ride-sharing request is in place.

in commitment should be dealt with some form of penalties. In this case, Amy should receive some penalties. This penalty can be financial punishment or reputation punishment (like rating) or it can include both.

2) Break of Commitment by the Provider: There might be a case, where a provider agrees to give the ride but later could not fulfill the promise. For example, Amy asks for a ride and AV1 agrees to give the Ride. However, later AV1 decides not to give ride. This scenario can also be considered as a break of commitment. In this case, AV1 should be at the receiving end of the punishment.

IV. Protocol Implementation and Case Studies

We implement our proposed AV ride-sharing Blockchain framework using Hyperledger Fabric [24][25]. Hyperledger Fabric is a Blockchain framework, which is implemented and hosted by the Linux Foundation. Hyperledger Fabric is a platform for distributed ledger solutions. It delivers high degrees of confidentiality, resiliency, flexibility, and scalability. Over the IBM-Blockchain Marbles project [26], we create our custom layer. On the layer, we implement AV ride-sharing rules, constraints, and business logics. We completely customize the IBM-Blockchain Marble demo by changing the user interface and adding custom rules. This implementation is a simple prototype of a secure and dependable AV ride-sharing exploiting the Blockchain. We run our implementation

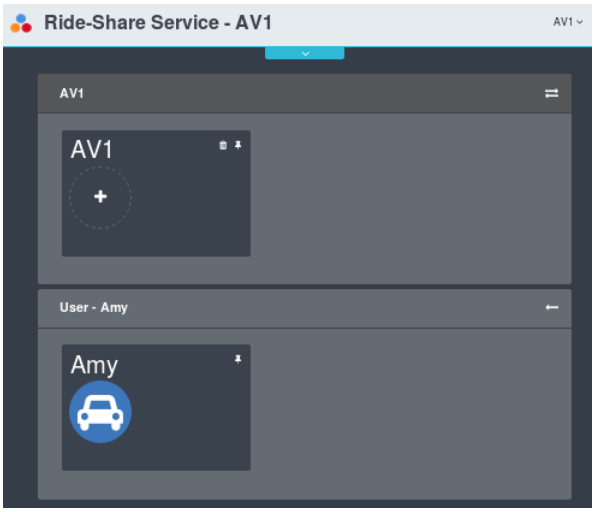

Fig. 8. AV1 is responding to Amy's request.

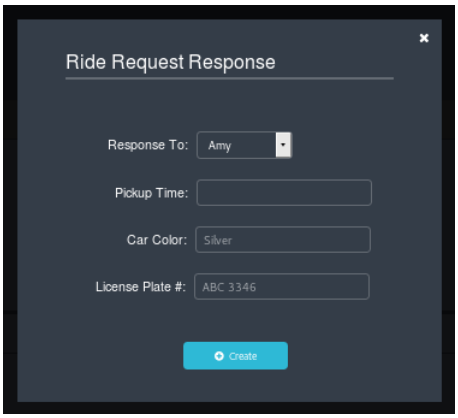

Fig. 9. AV1 is filling up the ride request response form.

in Debian operating system, over Intel Core i7 processor with 16 GB memory. In this section, we illustrate the execution of the proposed ride-sharing framework using several example case studies.

We now discuss an example scenario. At first, users log into the system using their individual login credentials. Each user has their own IoT device to log in. In our implementation, each user logs into the same peer node. If a user wants ride-sharing, it can initiate the ride-sharing request. In this example, let Amy is the consumer who wants ride-sharing service. Fig. 5 shows Amy's initial interface. Amy now creates a new request by filling up the ride information. In Fig. 6, we observe Amy's ride-sharing request. Here, Amy gives information about pick-up point, drop-off point, pick-up time, and drop-off time.

When Amy creates the new request, all AVs in the system can see it (Fig. 7). When a new request comes, an AV can accept the request (based on its route and timing) or can ignore the request. Let AV1 is accepts the request. AV1 acknowledges Amy's request by creating a response in the system. When AV1 replies, only Amy can see AV1's response. In Fig. 8, we can

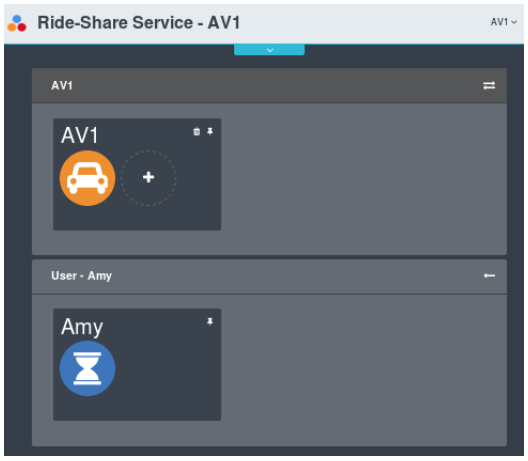

Fig. 10. AV1's response to Amy is complete.

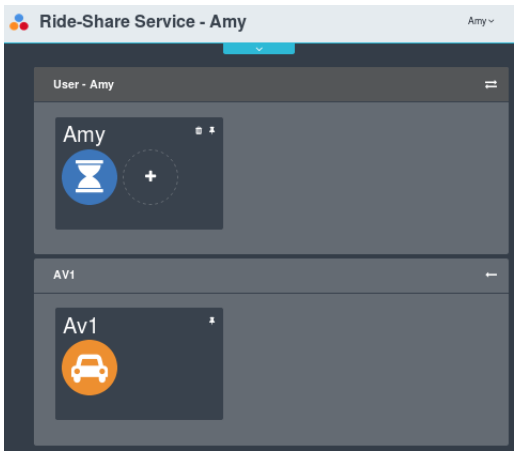

Fig. 11. Amy is waiting for AV1.

observe AV1's interface to Amy's request. When AV1 responds to the ride-sharing request of Amy, its is AV1's responsibility to fulfill the criteria given in Amy's ride-sharing request. In Fig. 9, we see the response from of AV1, where AV1 can only fill up the pick-up time in the response form. The other field are not editable as these are the constraints of Amy that must be met by the provider (AV1).

When Amy receives the response of AV1, she can see AV1's details along with the probable pickup time information. Other users cannot see this transaction. In Fig. 10, we see that AV1 completed the response to Amy's request. Now, Amy is waiting for AV1 as can be seen in Fig. 11. Once the request-response process is completed, AV1 will pick up Amy at the designated pick-up time.

V. Conclusion

With the rise of AVs, ride-sharing will gain momentum among users. However, AVs need to earn the trust of the users to promulgate the idea of ride-sharing through AVs. Exploiting the Blockchain as an underlying communication mechanism will ensure the trust and dependability. We have proposed a framework, protocol for share ride services and utilized the blockchain technology to provide a trusted or authenticated means of communication among the users. In the future work, we plan to implement our algorithms in a simulator to see how it operates in a whole AV network.

References

[1] J. C. Stutts, D. W. Reinfurt, L. Staplin, E. A. Rodgman, et al. The role of driver distraction in traffic crashes, 2001.

[2] A. Ozimek. The massive economic benefits of self-driving cars, 2014.

[3] L. Burns. Sustainable mobility: a vision of our transport future. *Nature*, 497(7448):181–182, 2013.

[4] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[5] Sarah Underwood. Blockchain beyond bitcoin. *Communications of the ACM*, 59(11):15–17, 2016.

[6] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto. Bigchaindb: a scalable blockchain database. *BigChainDB*, 2016.

[7] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE, 2013.

[8] Y. Desmedt. Man-in-the-middle attack. In *Encyclopedia of cryptography and security*, pages 759–759. Springer, 2011.

[9] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32, 2014.

[10] M. Kamali, L. Dennis, O. McAree, M. Fisher, and S. Veres. Formal verification of autonomous vehicle platooning. *Science of Computer Programming*, 2017.

[11] M. Mladenovic and M. Abbas. Self-organizing control framework for driverless vehicles. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2076–2081. IEEE, 2013.

[12] J. Hu, L. Kong, W. Shu, and M. Wu. Scheduling of connected autonomous vehicles on highway lanes. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 5556–5561. IEEE, 2012.

[13] P. Petrov and F. Nashashibi. Modeling and nonlinear adaptive control for autonomous vehicle overtaking. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1643–1656, 2014.

[14] Q. Li, L. Chen, M. Li, S. Shaw, and A. Nuchter. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. *Vehicular Technology*, 63(2):540–555, 2014.

[15] M. Alsabaan, K. Naik, and T. Khalifa. Optimization of fuel cost and emissions using v2v communications. *IEEE Transactions on intelligent transportation systems*, 14(3):1449–1461, 2013.

[16] P. Gomes, C. Olaverri-Monreal, and M. Ferreira. Making vehicles transparent through v2v video streaming. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):930–938, 2012.

[17] S. Ma, Y. Zheng, and O. Wolfson. T-share: A large-scale dynamic taxi ridesharing service. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 410–421. IEEE, 2013.

[18] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *Security and Privacy (SP)*, pages 839–858. IEEE, 2016.

[19] A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. In *PerCom Workshops*, pages 618–623. IEEE, 2017.

[20] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.

[21] T. Hardjono and N. Smith. Cloud-based commissioning of constrained devices using permissioned blockchains. In *International Workshop on IoT Privacy, Trust, and Security*, pages 29–36. ACM, 2016.

[22] Cloud service providers. https://www.sdxcentral.com/cloud/definitions/what-are-cloud-service-providers/.

[23] K. O'Dwyer and D. Malone. Bitcoin mining and its energy footprint. 2014.

[24] Hyperledger fabric. https://github.com/hyperledger/fabric/tree/master/docs.

[25] C. Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[26] IBM. Ibm-blockchain marbles. https://github.com/IBM-Blockchain/marbles.