Blockchain-Based Solution for Product Recall Management in the Automotive Supply Chain

PRATYUSH KUMAR PATRO 1, RAJA WASIM AHMAD 2, IBRAR YAQOOB 2, (Senior Member, IEEE), KHALED SALAH 2, AND RAJA JAYARAMAN 1

1Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
2Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates

Corresponding author: Ibrar Yaqoob (ibrar.yaqoob@ku.ac.ae)

This work was supported by Khalifa University through the Center for Digital Supply Chain and Operations Management under Award RCII-2019-002.

ABSTRACT
Product recall management in the automotive industry is a challenging problem that affects human lives and the safe operation of automobiles. Product recalls can assist in removing potentially unsafe products from the marketplace and minimizing a company’s responsibility for corporate negligence. Today’s systems and technologies leveraged for product recall management in the automotive supply chain fall short in providing transparency, traceability, reliability, audit, security, and trust features. In this paper, we propose a blockchain-based approach to overcome the aforementioned problems related to product recall management. We employ the public Ethereum blockchain and integrate it with the decentralized storage of the InterPlanetary File System (IPFS) to deal with the large-sized data problem. We present the system design and six algorithms explaining the working principles, information exchange flow, and stakeholders’ detail and their sequential interactions. We discuss the implementation details, generalization aspects, and cost and security analyses to evaluate the performance of the proposed approach. The proposed solution is cost-effective, secure, and enables automakers to have end-to-end visibility of information during product recalls. We make the smart contracts’ code publicly available on GitHub.

INDEX TERMS
Blockchain, Ethereum, automotive industry, product recall management, automotive supply chain, smart contract, traceability.

I. INTRODUCTION
The U.S. automotive industry is one of the contributors to the country’s economy and accounts for up to 4% of the overall country’s Gross domestic product (GDP) [1]. The growing demand for high-spec vehicles and automobile product requirements have made the automotive supply chain very complex [2]. Almost every automaker adopts various supply chain strategy to improve their business operations. The overall automotive supply chain comprises both the forward supply chain and the reverse supply chain. A forward automotive supply chain involves transferring automobile products introducing several stakeholders from an automobile parts supplier, automaker, dealer, and consumer [3]. In contrast, a reverse supply chain begins with the product and information flowing from the consumer to the automaker.

Product recall is one of the critical categories of the reverse supply chain [4]. Therefore, every automaker must prioritize handling its overall supply chain efficiency to stay competitive in the global market without compromising its quality and services. Failure to maintain the safety standards of the products can cause product recalls, which impact the company’s financial performance and brand reputation.

A product recall is a safety measure taken by a manufacturer to recover faulty or defective goods that have been used by consumers and take the appropriate steps to correct the problems [5], [6]. In the U.S., the scale of automotive recalls has grown from 30,000 to 90,000 vehicles per recall incident in 2016–2017 [7]. Such a defect poses a significant risk of severe injury or death and may cause an unexpected and catastrophic failure of a component with little or no warning to the user. In [8], the authors surveyed various defects found in multiple automotive functional systems, such as electronics, steering systems, fuel systems, etc., on
345 passenger vehicles in the U.S.. According to the study, vital automotive components and systems accounted for half of all recalls. This type of recall significantly affects the brand reputation and financial performance of the company. In 2010, the accelerator pedal recall conducted by Toyota was one of the most highly publicized recalls in the year 2010. It cost the company $7 billion due to the reparations, litigation fees, loss of sales, etc. [9]. Similarly, Ford Motor Company recalled over 14.9 million cars worldwide due to speed control deactivation leaking circuits between 1999-2009 [10].

The common causes of recall in the automotive industry involve both major problems like design defects, counterfeiting of spare parts, manufacturing defects, and minor defects like mislabelling automobile products [5], [6], [8]. In the case of several complaints reported about automobile defects, the automaker initiates a recall based on the standard guidelines. A product recall in an automobile can be defined as a recall focusing on specific automotive products like tyres, brake pads, airbags, etc. There are two types of automotive recall in the US automotive market: a voluntary recall and a reactive recall [9]. Automakers initiate a voluntary recall when a defect is identified before the involvement of NHTSA. In comparison, the reactive recall involves the NHTSA directing the automaker to initiate a recall when the NHTSA receives several complaints from various external entities like customers, dealers, etc. [9]. Customers and consumers, the National Highway Traffic Safety Administration (NHTSA), automakers/manufacturers, retailers/dealers, and automobile parts suppliers are all involved in the current product recall process in the US automotive industry [9], [11], [12]. In this paper, we predominantly focus on improving the existing process of reactive recall management using blockchain. Figure 1 illustrates an overview of the product recall management process in the automotive industry in the US.

When an automotive product recall is initiated, all stakeholders in the automotive supply chain must work together to ensure a smooth recall. As vehicle recalls are an alarming social and safety issue, the automaker needs to be proactive in approaching the stakeholders and customers for the fix as soon as possible. To deal with this, the automakers must maintain absolute information and data transparency among all the stakeholders [13]. The most evident challenges in a recall situation are disconnected data sources, undocumented processes, poor traceability, a lack of a structured process to coordinate among stakeholders, and inefficient procedures resulting in long turnaround times [11]. The main causes of a longer turnaround time in responding to an automotive product recall situation are asymmetric information and various cognitive structures among the external stakeholders such as component suppliers, customers, and dealers [14]. In addition, there is always a risk in the relationship between supplier and buyer in the aspect of social sustainability in the event of a product recall, which requires mutual trust and information transparency among the supply chain stakeholders [15]. To counter the challenges of preventing and managing automotive product recalls, the auto executives of various organizations use various data analytic and predictive analytic methods [16]. In [17], the authors propose a data mining approach to improve the product traceability and coordination of workflows among the stakeholders during the recall notification process for efficient recall management. However, these techniques pose the risk of trustworthiness with the data being generated from the resources. In the case of centralized servers, there is a chance of data manipulation by malicious users or system server failures resulting in data loss. Moreover, these techniques require significant trust in information sharing for successful execution. Modern technology advancements like blockchain technology with smart contracts can deliver a powerful and feasible solution, ensuring the traceability of automotive products and eradicating the need for a centralized authority [18].

The widespread adoption of blockchain technology in various business sectors such as manufacturing, aviation, finance,
energy, healthcare, and agriculture, among others, demonstrates the technology’s potential to address the long-standing challenges of the last few decades. The blockchain is genuinely an unalterable, decentralized, and distributed public ledger that permits the participants to safely maintain a record of transactions [20]. It is a shared distributed ledger comprised of augmented blocks containing details of all transaction data and traceable performance results [21]. Each block of the blockchain network can be hashed and connected to the next block, making the network more secure and tamper-proof [22]. Therefore, we chose to use blockchain technology to address the challenges in the existing automotive product recall management process. This paper aims to show how blockchain can efficiently improve the existing automobile product recall management process using Ethereum smart contracts. The main contributions of this paper are as follows:

- We propose a blockchain-based approach to enable product recall management in the automotive supply chain in a manner that is transparent, traceable, auditable, secure, and trustworthy.
- We develop four smart contracts and six algorithms. We present the system architecture by explaining the main interactions among the stakeholders using sequence diagrams. We make the smart contracts code publicly available on the GitHub repository.\(^1\)
- We evaluate the proposed blockchain-based recall management approach using cost and security analyses to show our solution’s affordability, reliability, and robustness. Also, we compare our solution with the existing blockchain-based solutions.
- Our proposed approach is generic enough to be applied to either public or private blockchain networks with minimal changes depending on the needs of the existing automotive industry.

The remainder of the paper is structured as follows. Section II presents the related work. Section III explains the proposed blockchain-based solution with all the system participants and shows the interactions of stakeholders using sequence diagrams. Section IV describes the implementation details and algorithms used in the smart contracts. Section V presents the testing and validation details of the proposed solution. Section VI presents the cost and security analyses, comparison analysis, and generalization aspects. We provide the concluding remarks in section VII.

II. RELATED WORK

This section presents some of the important existing non-blockchain and blockchain-based approaches proposed for traceability of products during recalls in the automotive, pharmaceutical, and food industries.

The supply chain process of every product requires interaction and trust among various stakeholders that are part of the supply chain. Tracking and tracing of products and processes are very important. In [22], the authors proposed a framework for a food product traceability system driven by big data technology that can improve product traceability and minimize recall costs. The main supply chain stakeholders considered in this system include the manufacturer, packaging, logistics, distribution, and the customer. The proposed approach leveraged big data technology, which can store and handle a large amount of data. The proposed architecture extracts and collects product behavior data using mobile applications. The product information collected includes the product id, timestamp, product condition, and product location in a data management system. This approach depends on using the QR-Code technique. However, this approach has not addressed the problems related to the counterfeiting of products, one of the major causes of product recalls.

In [23], the researchers proposed a model based on game theory concept to analyze traceability interactions and optimize product reliability in the supply chain using optimization techniques during product recalls. It demonstrates three types of supply chain tracking models to trace and sell the products. The findings also explain how cost influences the relationship between product traceability and reliability in product recall in the supply chain. However, this approach has not demonstrated any prototype solution that can efficiently address the existing product recall challenges. In [24], the authors explained the implementation of seven components or characteristics of an efficient traceability system to manage product recall for Halal products. The traceability system contains the following elements: certification, food safety policy, legislation, documentation sustainability, chain communication, competitive advantage, and cost reduction. These seven elements can be referred to when developing a good traceability system which can be very helpful during any product recall. This paper does not provide the system architecture and prototype to create an efficient product tracking system.

In [25], the researchers used a descriptive statistics technique to analyse and identify the best recall strategy for effective recall management in the meat industry. The study considers two measures: recall completion time and recalled product recovery, to develop an effective recall strategy. This technique employs a statistical hypothesis approach to analyse and develop an effective recall strategy. However, the generalization of this approach was not discussed in this study.

Although the blockchain applications in various businesses like supply chain, banking, logistics, and finance have been increasing exponentially, the literature on product recall management is minimal. Therefore, there exists an opportunity to improve the scope of blockchain technology in this aspect. In [26], the authors proposed a blockchain-based framework for pharmaceutical product recall management that can shorten the time to start a transparent recall of medicines. The proposed approach consists of an intelligent gateway and component-based smart contracts to process and regulate the pharmaceutical recall process.

\(^1\)https://github.com/PratyushKumarPatro/Automotive-Recall
One of the most common causes of a recall is counterfeit automotive parts. In [27], the authors proposed a blockchain enabled vehicle system named “blockchain-aware vehicle” that can prevent counterfeiting of automotive parts by tracking and tracing the part’s details through RFID or QR code. The main stakeholders considered by the proposed system include the supplier, the original equipment manufacturer (OEM), and the dealer. The system uses chaincodes to control authorization, confidentiality, and the accountability of every action by the respective stakeholders in a permissioned blockchain platform. However, the cost analysis of the proposed system is not presented to show the affordability of the solution.

In [28], the authors proposed a decentralized application named PartChain that enables tracking and tracing of parts across the supply chain to help the manufacturer identify defective parts during a product recall. The PartChain application creates, monitors, and provides a virtual representation of physical components in the form of digital twins. The main stakeholders involved in the proposed system include the OEM, suppliers, and a logistics service provider. The system employs the Hyperledger Fabric platform to deploy the system’s working prototype. The smart contracts automatically verify and execute the conditions and functions needed to create, place, deliver, and transfer ownership of the orders. This study can be further improved by analyzing the cost and security analysis.

Unlike the approaches mentioned above, in [29], the authors proposed a blockchain-based system to track the ownership of automotive spare parts throughout their life cycle.
This approach involves stakeholders from different departments like purchasing, inventory management, etc., and uses Ethereum based smart contracts to track and trace the business operations. It also demonstrates the cost and security analysis to show the affordability of the prototype solution. However, this approach does not explain the application of tracking and tracing of automotive products in the event of a recall.

In summary, none of the existing solutions are applicable to automotive product traceability in the recall process. Also, the existing solutions do not cover the broader scope of product recall management with a specific framework, approach, cost analysis, and security analysis in the automotive industry. In contrast, our method provides a blockchain-enabled solution that can help the stakeholders trace the products, activities, and processes to manage the recall situation in a secure and cost-efficient way.

III. PROPOSED BLOCKCHAIN-BASED SOLUTION

This section presents the system architecture of the proposed blockchain-enabled approach with Ethereum smart contracts to efficiently manage the product recall process in the automotive industry. Figure 2 shows the stakeholders, system elements, actions, and responsibilities of the stakeholders involved in the automotive product recall management process.

A. SYSTEM STAKEHOLDERS AND ELEMENTS

Herein, we discuss the key activities of each stakeholder engaged in the automotive product recall management process. It also details the system’s critical components that enable the sharing of essential data among system stakeholders and elements.

1) CUSTOMER
The customer acts as an informer or reporter of any defect observed in the automobile and raises a defect complaint to the NHTSA. It also receives the recall notices sent by the automaker. Upon completion of all the necessary services, it accepts the serviced vehicle and confirms receiving it.

2) NHTSA
The NHTSA acts as the regulatory authority and registers all the stakeholders in the blockchain network. It accepts the defect complaint raised by the customer and requests for an inspection from the inspection department. It is also responsible for auditing all the actions and issuing penalties to the automaker in case of non-compliance with standard regulations during the entire recall process.

3) INSPECTION DEPARTMENT
The inspection department accepts the inspection request and conducts an inspection. Based on the regulations and standards set by the regulatory authorities, it performs the inspection and uploads an inspection report to the decentralized storage system InterPlanetary File System (IPFS).

4) DEALER
The dealer is a certified entity that sells the vehicles of the franchised automaker to the customer. In this approach, it is responsible of receiving the automobile parts shipped by the automaker and performing the vehicle’s service. It also communicates to the customer the readiness of the serviced vehicle. The dealer is also responsible for off-chain communication about the logistics arrangement of the vehicle.

5) AUTOMAKER
The automaker is the prime stakeholder in the overall recall process and is accountable for the overall automotive product recall campaign. The automaker initiates the recall based on the inspection outcome and sends the recall notices to all the registered customers and dealers. It is also liable for coordinating with the automobile parts supplier to purchase new automobile parts and receive the parts shipments.

6) AUTOMOBILE PARTS SUPPLIER
The automobile parts supplier is responsible for accepting purchase orders from the automaker, updating the inventory database, and shipping the automobile parts to the automaker.

7) SMART CONTRACTS
Smart contracts contain programmable logic with functions that can be executed and events that can be triggered. They ensure the execution of the rules set by the stakeholders. Our system architecture comprises four smart contracts: the Registration Smart Contract, the Recall Handler Smart Contract, the Corrective Action Smart Contract, and the Inspection Handler Smart Contract. These smart contracts are encoded with business operational logic and trigger events when a particular function is initiated.

8) DISTRIBUTED STORAGE
Distributed storage systems like IPFS can store big-sized files, images, and records to counter the storage space constraints of blockchain. In our approach, big-sized files relate to the pictures of faulty parts, inspection reports, vehicles, and certification documents. These documents are placed on the IPFS system, and the hash values of these documents are available on the blockchain ledger. As blockchain technology is immutable, stakeholders can verify the stored information and access it to perform business operations.

B. SEQUENCE OF OPERATIONS

In this subsection, we present the sequence diagrams of all the system operations in the form of functions and events. The sequence diagrams also explain the interactions among various stakeholders and the smart contracts.

The sequence diagram displayed in Figure 3 presents the interaction of the customer, NHTSA, and the inspection department with the smart contracts. All the stakeholders are registered on the blockchain through the registration smart contract deployed by the NHTSA. A unique identification
code is created upon the successful registration of the stakeholders, which is used as a reference to the participants and their actions. Once the stakeholders are registered on the network, the customer deploys the recall handler smart contract using the respective function. The recall handler smart contract accepts or rejects the requests placed by the customer depending on the rules encoded into it. Once the recall handler smart contract is deployed successfully, the customer stakeholder calls the function *PlaceDefectComplaint*. In response, the NHTSA calls the function *ConfirmDefectRequest*. The NHTSA executes the function involving the Ethereum Address of the Customer (Customer EA), Vehicle Identification Number (VIN), Functional Module Name, Vehicle Model Name, and Vehicle Build Year. After successfully accepting the complaint request, the NHTSA deploys the inspection handler smart contract and raises an inspection request to the inspection department using the function *PlaceInspectionRequest*. The inspection department executes the function *ConfirmInspectionRequest* to accept or reject the request. Once the inspection is accomplished, the inspection department calls the function *CreateInspectionReport* to create the inspection report.

Figure 4 illustrates the interaction between the inspection department, NHTSA, automakers, IPFS, and smart contracts. Upon successfully creating the inspection report, the NHTSA calls the function *RequestInspectionReportUpload* placing a request to upload the inspection report to IPFS. After accepting the request to upload the inspection report, the inspection department calls the function *UploadInspectionReport* to upload the inspection report and notify the authorised stakeholders. The IPFS hash of the file gets stored on the blockchain to enable the authorised stakeholders to access the inspection report. The inspection report contains important information such as test results, simulation results, and images of the defective parts.

Figure 5 depicts the sequence diagram of the mutual interaction among the customer, dealer, automaker, smart
contracts, and NHTSA. Based on the results available in the inspection report, the automaker calls the function `CreateRecallNotice`. To execute this function, the function caller must provide the inputs, such as the Ethereum addresses of the customer, dealer, and the inspection report ID. The customer accepts the request by calling the function `ReceiveRecallNotice` and accepts or rejects the recall notice.

After successfully receiving the automobile parts from the automaker for vehicle servicing at the dealer’s facility, the automaker calls the function `CreateShipmentNoticeToDealer` to create a shipment notice and ship the required parts to the dealer. After receiving the parts and servicing the vehicle, the dealer calls the function `CreateServicedVehicleNotice` to create a notice about the serviced vehicle and notify the customer about the readiness of the vehicle. Upon accepting the serviced vehicle, the customer calls the function `ReceiveServicedVehicle` to notify the stakeholders about successfully receiving the serviced vehicle. The NHTSA calls the function `UpdatePenaltyStatusResult` to issue penalties to the automaker in case of any non-compliance with the regulations during the recall. In this paper, we considered a service time of 60 days as the threshold time and a deciding factor in issuing a penalty. If the threshold time is exceeded, the recall handler smart contract issues a penalty to the automaker, as the automaker is entirely responsible for the entire process.

Figure 6 illustrates the sequence diagram of the mutual interaction among the automaker, automobile parts supplier, and smart contracts.

Once the recall notice is sent to the customer, the automaker deploys the corrective action handler smart contract and calls the function `PlacePurchaseOrder` to place an order for the new replacement parts. The smart contract triggers the notification to the automobile parts supplier with parameters such as Ethereum address of automaker, automobile parts supplier, part number, the quantity of the parts, etc. The automobile parts supplier calls the function `UpdateInventoryDatabase` where it updates the inventory quantity and related details. Based on the available inventory status, the automobile parts supplier confirms the purchase order using the function `ConfirmOrder` to notify the automaker about the confirmation of the purchase order.

Upon successfully confirming the purchase order, the automobile parts supplier calls the function `CreateShipmentNotice`. The smart contract creates a parts shipment notice to ship the parts to the automaker. Upon receiving the parts, the automaker calls the function `ConfirmShipmentReceive` to inform the stakeholders on receiving the shipment.

IV. IMPLEMENTATION DETAILS

In this section, we implement our blockchain-enabled solution and explain the details of the algorithms. The smart contracts are executed in the Ethereum platform to operate the business operations of the stakeholders. The smart contracts are created, deployed, and run in the Remix IDE environment. Our objective is to digitalize the participating stakeholders’ product recall operations, including Registration, Recall handling, Inspection handling, and Corrective action handling.

Algorithm 1 exhibits the method of placing a defect complaint request to the NHTSA by the customer. The caller is required to provide the Ethereum address of the receiver of the defect complaint, the Ethereum address of the requester, vehicle identification number (VIN), functional module name, vehicle model name, and vehicle build year as the inputs, to successfully create a defect complaint request and notify the participating stakeholders as per the proposed system architecture. The stakeholders must be aware of this
information to conduct an inspection and initiate a recall successfully. In this approach, the functional module name depicts the functional system of an automobile, such as the steering system, engine system, etc. Similarly, the vehicle model name represents the model name of the vehicle. The build year describes the year of manufacturing of the vehicle. These details help the automaker and other stakeholders target the defect easily. It assures that only registered customers can report the defect and deny any request raised by unregistered customers or stakeholders. It records the defect complaint details: VIN, vehicle model name, functional module name, Ethereum addresses of the Receiver and the Requestor, build year, and status of the complaint. It deploys the Ethereum based cryptographic hash function called Keccak256 [36] that encrypts the input parameters to create a unique ID. We use the abi.encodePacked function with the Keccak256 function to create a hash that is used as an ID [36]. It is collision-free and irrevocable. This unique ID of the complaint request can track and trace the defect complaint request’s status during the recall management operations. In addition, it is more efficient as any slight change in the input creates a significant difference in the output due to the avalanche effect. The abi.encodePacked function generates a string by padding the input data with variables such as
caller EA, time, VIN, vehicle model name, functional module ID, and complaint status. The Keccak256 hash algorithm creates the DefectComplaintRequestID as the algorithm’s output. Then a notification about the awaiting/pending status of the defect complaint request is sent to the authorised stakeholders and recorded in the blockchain.

Algorithm 2 demonstrates the process of creating a request for defect inspection by the NHTSA to the inspection department after deploying the Inspection Handler smart contract. The function caller provides the Ethereum address of the receiver of the inspection request, the Ethereum address of the requestor, and DefectComplaintRequestID as inputs to create an inspection request and send a notification to the participating stakeholders successfully. It authorizes only the NHTSA to initiate a request for the defect inspection. It records the defect complaint details, such as the Ethereum addresses of the receiver and the requestor. To generate an inspection request ID, the Keccak256 hash algorithm [36] is
Algorithm 1 Submitting a Complaint About Defect

1. **Input**: Customer EA, NHTSA EA, Vehicle Identification Number, Functional Module Name, Vehicle Model Name, Vehicle Build Year
2. **Output**: Create DefectComplaintPlaced Event
3. **if** FunctionCaller is not a registered Customer **then**
   1. The Ethereum address of customer is incorrect.
   2. **else**
      1. **if** Function Caller EA = Customer EA **then**
         1. Create DefectComplaintRequestID using Keccak256.
         2. Update Defect complaint status to Pending.
         3. Alert the stakeholders by emitting an event regarding the creation of a defect complaint request using DefectComplaintRequestID, ReceiverEA, FunctionCallerEA, VehicleIdentificationNumber, Vehicle ModelName, FunctionalModuleName, and VehicleBuildYear.
      2. **else**
         1. Display an error notification and revert to the original state of contract.
   3. **end**
4. **end**

Algorithm 2 Submitting a Request for Defect Inspection

1. **Input**: NHTSA EA, Receiver EA, and Defect Complaint Request ID
2. **Output**: CreateInspectionRequestPlaced Event
3. **if** FunctionCaller is not NHTSA **then**
   1. The address of NHTSA is incorrect.
   2. **else**
      1. **if** Function Caller EA = NHTSA EA **then**
         1. Create DefectInspectionRequestID using Keccak256 function.
         2. Update Inspection request status to Pending.
         3. Alert the stakeholders by emitting an event regarding the creation of defect inspection request using DefectInspectionRequestID, ReceiverEA, RequestorEA, InspectionDepartmentEA, FunctionalModuleName, and VehicleBuildYear.
      2. **else**
         1. Display an error notification and revert to the original state of contract.
   3. **end**
4. **end**

Algorithm 3 Creating Inspection Report and Uploading It to IPFS

1. **Input**: Inspection Department EA, Automaker EA, Inspection Request ID, NHTSA EA, IPFS hash String
2. **Output**: CreateInspectionReportUploaded Event
3. **if** FunctionCaller is not InspectionDepartment **then**
   1. This Inspection department is not authorized to upload inspection report.
   2. **else**
      1. **if** Function Caller EA = Inspection Department EA **then**
         1. Create InspectionReportID using Keccak256.
         2. Update InspectionReportCreate status to Pending.
         3. Update InspectionReport status to Uploaded.
         4. Alert the stakeholders by emitting an event regarding the uploading of inspection report to IPFS server using the InspectionReportID, FunctionCallerEA, NHTSAEA, AutomakerEA, IPFSSString.
      2. **else**
         1. Display an error notification and revert to the original state of contract.
   3. **end**
4. **end**

Algorithm 4 highlights the process of initiating and sending the Recall notice to the customer and dealer. Based on the information provided in the inspection report by the inspection department, only the automaker is authorized to recall the automobile part. The recall notice is identified by a unique RecallNoticenumber that gets generated by the Keccak256 hash function. The parameters such as Automaker EA, InspectionRequestID, and NHTSA EA are provided as inputs to the abi.encodePacked function, which produces the encoded byte string. This unique identifier helps track and trace the Inspection report availability to the Automaker who is the Recall event initiator.

The address of the NHTSA is incorrect. The Ethereum address of customer is incorrect. Request is accepted, the inspection department inspects the faulty system and generates an inspection report.

In our approach, the Inspection report availability status is referenced by a unique InspectionReportID that gets generated by the Keccak256 hash function. The parameters such as Inspection department EA, Automaker EA, InspectionRequestID, and IPFS Hash are provided as inputs to the abi.encodePacked function, which produces the encoded byte string called Inspection Report ID, when the Keccak256 hash function [36] hashes the encoded byte string. This unique identifier helps track and trace the Inspection report availability to the Automaker who is the Recall event initiator.

The address of the NHTSA is incorrect. The Ethereum address of customer is incorrect. Request is accepted, the inspection department inspects the faulty system and generates an inspection report.
Algorithm 4 Initiating and Sending Recall Notice

| Line | Code                                      | Description |
|------|-------------------------------------------|-------------|
| 1    | **Input:** Customer EA, Dealer EA, Inspection Report ID, and Vehicle Identification Number |             |
| 2    | **Output:** RecallNoticeCreated Event     |             |
| 3    | **if** FunctionCaller is not Automaker EA |             |
| 4    | **then**                                  |             |
| 5    | The Automaker is invalid and initiating a recall notice is rejected. |             |
| 6    | **else**                                  |             |
| 7    | **if** Function Caller EA = Registered Automaker EA |         |
| 8    | **then**                                  |             |
| 9    | Create RecallNoticeNumber.                |             |
| 10   | Update Recall request status to Pending.  |             |
| 11   | Create an event and informing the stakeholders regarding the recall request using the RecallNoticeNumber, CustomerEA, AutomakerEA. |       |
| 12   | **else**                                  |             |
| 13   | Display an error notification and revert to the original state of contract. |         |
| 14   | **end**                                   |             |
| 15   | **end**                                   |             |

RecallNoticeNumber. The Recall notice number is used as a reference for all servicing actions of the product.

In our approach, in case of recall initiated due to faulty automobile components, the automaker places a purchase request to the respective registered automobile component supplier. The request is done to buy the parts by providing the required quantity of the parts and the part number ID. The Purchase order gets recognised by a unique Purchase order ID formed by the Keccak256 hash [36] function. The parameters Automaker EA, Automobile parts supplier EA, part number, and requested parts quantity are passed to the abi.encodePacked function, which generates the encoded byte string. This encoded byte string is hashed using the Keccak256 hash function [36] to generate the PurchaseOrderID. Upon getting the purchase order ID, the automobile parts supplier checks the parts availability based on the inventory status. Algorithm 5 highlights the process of inventory management and confirming the purchase order of the automobile parts. Based on the requested quantity of the specified part number, the automobile parts supplier checks the inventory and updates the available stock, excluding the parts order quantity. Once the supplier is ready to ship the parts, the corrective action handler smart contract generates a ShipmentNoticeID using the parameters PurchaseOrderID, the quantity of the parts to be shipped, and the Ethereum address of sender and receiver.

Algorithm 6 explains auditing the complete recall process and issuing a penalty in case of any deviation concerning government standards. It uses the time difference in days between the start of the recall to completion of the recall. As per the standard guidelines, the NHTSA sets the standard time to complete the recall process. Suppose the time taken to finish the recall is higher than the threshold time to service. In that case, the algorithm issues a penalty to the automaker, as the automaker is responsible for managing all the activities of the recall service process. After successful execution, it notifies the automaker about the penalty status.

V. SYSTEM TESTING AND VALIDATION

In this section, we demonstrate the outputs of the events and functions of our proposed solution. We executed all the smart contract functions in the Remix IDE and explained their results. We employed modifiers in the smart contracts to prevent access of unauthorised entities to restricted functions. The system generates an error in case of any unauthorised access. We included events in the smart contracts, to maintain data provenance and ease traceability. Every stakeholder and smart contract gets assigned with a
Algorithm 6 Auditing the Stakeholders Activities to Issue Penalties

1. **Input**: Automaker EA, NHTSA EA, Threshold Time to Service, Initiate Time, Completion Time
2. **Output**: Trigger PenaltyStatusUpdated Event and notify about the penalty status.
3. **if** FunctionCaller is not NHTSA **then**
   4. Only NHTSA is authorized to execute this activity.
5. **else**
6. **if** FunctionCaller EA = NHTSA EA **then**
   7. Calculate the Time taken to finish the Recall.
   8. **if** CompletionTime - InitiateTime ≥ ThresholdTimeToService **then**
    9. Set PenaltyStatusResult to True.
   10. **end**
   11. **else**
   12. Set PenaltyStatusResult to False.
   13. **end**
   14. Trigger an event to impose the penalty to Automaker and inform the stakeholders.
15. **end**
16. **else**
17. Display an error notification and revert to the original state of contract.
18. **end**
19. **end**

### TABLE 1. Participant’s Ethereum address.

| Name of Participant | Ethereum Address |
|---------------------|------------------|
| NHTSA               | 0x3B38Da6a70c568545dC6cB03F3d251f556beb3C4 |
| Customer            | 0xAb8483F64d9C6d1Ecf9B849Ae677dD3315835cb2 |
| Dealer              | 0x4B309953b58177ec7B9571ceC95Ae22C02db |
| Inspectiondepartment | 0x78731D3Ce6b7B3B4C63242a7eC2A1F3e2caB8 |
| Automaker           | 0x17F02E2D72F50D53315835cb2 |
| Automobilesupplier  | 0x17FaGAD85f982397579C203699C1D6b5B544c372 |

unique Ethereum address that can help identify the entities. This system can track and trace the status of the vehicles and products from the start of the recall process until the end. The use of penalties ensures better control of the process for any non-compliance of standards. A unique Ethereum address identifies each stakeholder in the system. The details of the Ethereum addresses of each participant are provided in Table 1. For example, the Ethereum address “0xAb8483F64d9C6d1Ecf9B849Ae677dD3315835cb2” has been assigned to a customer.

### A. COMPLAINT RAISING AND INSPECTION

This subsection presents a few outputs of the executed functions, starting from defect complaint raising till inspection completion.

Figure 7 displays the execution results of the PlaceDefectComplaint function that is operated by the RecallHandler smart contract. The PlaceDefectComplaint function permits the customer to register a complaint regarding the defect found in his/her vehicle. The system verifies the Ethereum address of the customer to validate that the requester is a registered user. The smart contract ensures that the NHTSA can only execute this function. The smart contract initiates an event named DefectComplaintPlaced upon the successful execution of the PlaceDefectComplaint function. The event also records the blockchain transaction details that include key information details such as DefectComplaintRequestID, VIN, functional module name, vehicle model name, build date, and address of the requestor and receiver.

The successful execution of the PlaceDefectComplaint function also changes the transaction status to Pending so that the stakeholders can view the latest status of the complaint request. The receiver of the request can verify the details through the system. Figure 8 explains the output of the execution of ConfirmDefectRequest function. This function is called by the NHTSA. The output of this function contains the acceptance or rejection status of the defect complaint request based on the standard guidelines. With the successful execution of the ConfirmDefectRequest function, the response of NHTSA gets recorded on the blockchain, and the participants get notified about the latest defect complaint status by calling an event named UpdateStatus.” This event ensures the documentation of the status of the complaint request along with the DefectComplaintRequestID.

Figure 9 highlights the results of the functions and demonstrates the key elements of the DefectComplaintRequestID. After confirming the defect complaint request successfully, the NHTSA requests an inspection by the inspection department. Figure 10 shows the details of the key elements of the inspection request ID that gets generated when the NHTSA calls the PlaceInspectionRequest function in the InspectionHandler smart contract. PlaceInspectionRequest function enables the NHTSA to request an inspection regarding the defect complaint raised by the customer. The system verifies the Ethereum address of the inspection department to validate that the requestor of the inspection request is a registered user. An event named “InspectionRequestPlaced” gets released on the successful execution of the function and the transaction details get recorded on the blockchain. In this operation, the transaction log contains details such as InspectionRequestID, the address of the requestor, and the receiver.

After the successful execution of PlaceInspectionRequest function, the smart contract changes the transaction status to Pending. It also displays the present status of the inspection request to the authorised stakeholders. The inspection department calls the ConfirmInspectionRequest function to notify its response by either confirming or rejecting the inspection request.

After the successful execution of ConfirmInspectionRequest function, the inspection department’s response gets
recorded on the blockchain. To notify about the current status of the inspection request, the event “Statusupdated” is emitted. This event ensures the recording of the inspection request’s updated status as well as the InspectionRequestID generated by the `PlaceInspectionRequest` function.

After confirming the inspection request successfully, the inspection department conducts the inspection and creates an inspection report.

Figure 11 displays the results of the successful execution of the `CreateInspectionReport` and `UploadInspectionReport` functions. The `CreateInspectionReport` function enables the inspection department to create the inspection report. An event entitled “InspectionReportCreated” gets generated with the successful execution of the function.

**B. RECALL INITIATION AND PARTS SHIPMENT**

This subsection demonstrates a few output results of the executed function, starting from recall initiation to parts shipment.

Based on the outcome of the inspection, the automaker decides whether to initiate a recall or not. To initiate a recall, the automaker calls the function `CreateRecallNotice` which is part of the Recall handler smart contract to create the recall notice for the affected vehicles. Figure 12 displays the result in executing the `CreateRecallNotice` function. Once this function gets executed successfully, an event titled `RecallNoticeCreated` is emitted. This event also documents the transaction details such as RecallNoticeNumber, InspectionReportID, InspectionreportID, VIN, Ethereum address of the customer, dealer, and automaker on the blockchain.

With the successful execution of the `CreateRecallNotice` function, the customer responds on the acceptance of the recall notice using the function `ReceiveRecallNotice`. Figure 13 exhibits the transaction details of the
ReceiveRecallNotice function. When this function is successfully executed, an event called RecallNoticeReceived is triggered, which maintains a record of the transaction details on the blockchain.

Figure 14 relates to the business operation where the purchase order for the automobile parts is placed by the automaker. It shows the results of executing the PlacePurchaseOrder function after successfully deploying the Correctiveactionhandler smart contract. The successful execution of this function causes an event to be triggered, which records the transaction details such as the requestor’s and receiver’s Ethereum addresses, PurchaseOrderID, Partnumber, and requestedpartquantity. Upon receiving the Purchase order, the automobile parts supplier calls the UpdateInventoryDatabase function. The successful execution of this function triggers an event named UpdatedInventory which stores the details of the PartID and Available inventory.

Following a successful inventory update, the automobile parts supplier invokes the ConfirmPurchaseOrder function of the Correctiveactionhandler smart contract by accepting or rejecting the purchase order by analyzing the available parts inventory. When this function is successfully executed, it generates the desired transaction output and stores the automobile parts supplier as the receiver of the purchase order, as well as the quantity of the parts to be shipped.
To initiate the shipment of the parts to the automaker, the automobile parts supplier executes the `CreateShipmentNotice` function to create a shipment notice. When this function is successfully executed, an event termed as `ShipmentNoticeCreated` gets created that documents the transaction features such as the sender’s and receiver’s Ethereum addresses and the `ShipmentNoticeID`. As part of the smart contract rules, the `ShipmentNoticeCreated` event gets triggered that records the transaction details containing `ShipmentNoticeID`. The `ShipmentNoticeID` contains the details of Ethereum addresses of requestor and receiver, Part number, requested part quantity, quantity to ship and the shipment status. After successfully receiving the automobile parts, the automaker informs the automobile parts supplier about receiving the shipment using the function `ConfirmShipmentReceive` which has been illustrated in Figure 16. Figure 17 displays the execution result of the `ConfirmShipmentReceive` function. When the function is successfully executed, it emits an event that allows the transaction details such as `ShipmentNoticeID` and event status to be recorded on the blockchain.

To fix the defective parts and replace them with the faultless parts at the dealer service centers, the automaker ships the replacing parts to the dealer by calling the `CreateShipmentNoticeToDealer` function. Figure 18 shows the results of the successful creation of parts shipment notice to the dealer. When this function is successfully executed, the Recall handler smart contract creates an event that records the transaction details and stores key details such as the sender’s and receiver’s Ethereum addresses, `ShipmentNoticeNumber`, and `PartNumber`. The `DealerPartShipmentCreated` event is generated as a result of the execution of this function. The event records the transaction details on the blockchain.

### C. VEHICLE SERVICING AND PENALTY

Figure 19 presents the creation of serviced vehicle notice to the customer, automaker, and NHTSA by the dealer. It also shows the results of executing `CreateServicedVehicleNotice` function. When this function is successfully executed, an event is generated. As part of the transaction records, the event stores the Ethereum addresses of the sender and receivers, as well as the `ServicedVehicleID`, `ShipmentNoticeNumber`, and `VIN`. Following the successful execution of this function, the `CreateServicedVehicleNotice` event is triggered, which records the transaction details in the blockchain. When the customer receives a Serviced vehicle, the customer calls the `ReceiveServicedVehicle` function. Following the flawless execution of this function, an event is generated that allows for the documentation of the transaction details by saving the Ethereum addresses of the sender and receivers, the `ServicedVehicleID`, and the status.
FIGURE 18. Highlighting the successful creation and receiving of shipment notice number.

execution of this function, the ServicedVehicleReceived event gets triggered to the prime stakeholders such as: automaker, NHTSA and dealer to notify about the customer satisfaction of the service.

Figure 20 and Figure 21 present the results in which a penalty in terms of cost has been issued to the automaker as the time of recall service is higher than the ThresholdTimeToService. The results obtained on executing the UpdatePenaltyStatusResult function by the NHTSA are based on the recall initiation time and completion time. The time taken to finish the recall is obtained by subtracting the recall initiation time from the recall completion time. The time also records the status of penalty in terms of Boolean values. “1” indicates the status of penalty as True/applicable. In contrast, “0” indicates the status of penalty as False/not applicable. This status accompanied with an event PenaltyStatusUpdated triggered to the automaker.

VI. DISCUSSION
This section presents the cost and security analyses of our proposed solution to show its affordability and robustness against well-known security attacks and vulnerabilities.

A. COST ANALYSIS
Each operation performed on the Ethereum blockchain utilizes a certain amount of gas. To execute a transaction, the user must specify the maximum amount of gas. All the functions in the smart contracts are run on the Ethereum virtual machine (EVM). The transaction and execution cost of the operations are obtained in terms of gas cost. The EVM might run out of gas if the execution costs exceed the maximum threshold. Users can also offer higher gas prices to miners in order for them to prioritize their transactions. Gas prices use the unit Gwei. The gas cost analysis can be very helpful in developing cost efficient smart contracts.

Ethereum Gas Station determines the gas cost for the transactions performed on the Ethereum platform. It provides varying speeds for completing transactions, which are heavily influenced by the user’s gas price. The Ethereum Gas Station can calculate the price of gas units based on transaction execution speed, which can be fastest, fast, average, or slow. As a result, the price of gas may vary depending on the time of the day. For the cost analysis of our proposed solution, we use the ETH Gas station’s prices as of June 11, 2021 [30].

For the gas units of 21000, we use a transaction fee of 9.30 Gwei. In addition, we considered the conversion rate of Ether to USD regarding the value available on 08-Oct-2020 to denote a steady and practical value.

Table 2 displays the gas unit consumption by the list of transactions and functions of our proposed approach. The gas unit values are obtained when the smart contracts are run on
the Remix IDE. The table also shows the transaction fee in Ether for the fastest and slowest transaction execution speeds for each function. The slow transaction execution requires fewer Ethers than the fastest execution. In Table 2, `PlaceDefectComplaint` and `ShipmentNoticeCreation` require more gas to execute the functions successfully due to the complexity of the smart contract logic and business operations. From the cost analysis table, the total cost of operation with slow execution speed is approximately 5.4 USD, which indicates that our solution is quite affordable and cost-efficient.

### B. SECURITY ANALYSIS

Herein, we present the security analysis of the proposed solution. Our solution offers a high level of security, robustness, and resilience. The essential security factors and details of our proposed system are provided below.

1) **DATA INTEGRITY AND TAMPER PROOF**

One of the most significant risks to data integrity is data exploitation. Data integrity is preserved in the blockchain via cryptographic functions. Because of hashing algorithms, blockchain transactions are immutable. In our proposed approach, stakeholders digitally sign transactions before they are updated on the blockchain. As a result of the immutability of the transactions, it is impossible to amend, modify, or delete these transactions, maintaining the data integrity of our proposed solution intact.

2) **AVAILABILITY**

Unlike centralized systems, the decentralization feature has made blockchain technology very popular. Decentralization makes the system significantly available to all the stakeholders in a secure manner. Therefore, the risk of denial of service (DoS) attacks is almost impossible. As our system is built on the Ethereum blockchain, its operations are always securely accessible to the stakeholders.

3) **ACCOUNTABILITY**

Accountability is necessary for emerging technologies like blockchain to help users perform their activities in
TABLE 3. Comparison with the existing blockchain-based solutions.

| Study            | Topic Domain                        | Prototype | Security Analysis | Cost Analysis | Decentralized | Traceability |
|------------------|-------------------------------------|-----------|------------------|---------------|---------------|--------------|
| [26]             | Pharmaceutical product recall       | Yes       | No               | No            | Yes           | Yes          |
| [27]             | Automotive parts counterfeit        | Yes       | No               | No            | Yes           | Yes          |
| [28]             | Automotive supply chain traceability| Yes       | No               | No            | Yes           | Yes          |
| [29]             | Spare parts traceability in manufacturing | Yes     | Yes             | Yes           | Yes           | Yes          |
| [Our Study]      | Automotive product recall management| Yes       | Yes             | Yes           | Yes           | Yes          |

The tool can report vulnerabilities like integer underflow and overflow, callstack depth attack vulnerabilities, Transaction-Ordering Dependence (TOD), timestamp dependency, and re-entrancy attacks. The summary result of Oyente for our system smart contracts is displayed in Figure 22. All of the vulnerabilities analyzed by the Oyente tool were marked as “false”. This indicates that our system’s smart contract is completely free from any vulnerabilities.

C. COMPARISON WITH THE EXISTING BLOCKCHAIN-BASED SOLUTIONS

Table 3 presents a comparative analysis of our proposed solution against the existing solutions proposed in [26]–[29]. The solutions proposed in these studies attempt to address the challenges without cost and security analyses, except [29]. In [26], the domain of the topic is the pharmaceutical recall supply chain, which is different from the automotive recall supply chain. The study conducted in [27] predominantly focuses on the counterfeiting of automotive parts, one of the major causes of the automotive product recall. However, challenges like collaborating with all stakeholders and bringing them under one common platform to share the information transparently have not been addressed. In [28], the proposed approach provides solutions for tracking and tracing automotive products using blockchain. However, the stakeholders involved in the proposed recall supply chain are restricted only to the OEM, supplier, and logistics provider. Our approach includes the other stakeholders like the customer, dealer, NHTSA, inspection department, etc., to cover the broader aspect of automotive recall management. Likewise, the proposed solution presented in [29] also demonstrates the blockchain-based approach for tracking and tracing spare parts in the automotive manufacturing environment. Although it presents cost and security analyses of the solution, the domain of this approach is not related to automotive recall management.

Unlike the studies mentioned above, our proposed solution involves the most vital automotive recall supply chain stakeholders. It also establishes a blockchain-enabled approach to improving existing business operations in a automotive product recall supply chain. We also used Remix IDE to implement and test the proposed framework and conducted the necessary analysis to demonstrate the feasibility of our solution.

D. GENERALIZATION

Our proposed system is developed, tested, and validated on a public Ethereum platform to meet the traceability, transparency, and security requirements of the automotive...
product recall supply chain. As the blockchain platforms can efficiently encrypt the intended data, our proposed approach delivers a secured solution to the business stakeholders to record their transactions. Any industry that deals with product recalls can modify the proposed smart contracts according to their business requirements.

The proposed system can effectively trace and track the activities and elements of the proposed automotive recall supply chain. It can also be adopted by other product industries that deal with product recalls, such as food, pharmaceuticals, toys, etc. The proposed solution provides efficient tracking of the stakeholders, their functions, and activities in the automotive product recall supply chain. Hence, this research can handle product recall supply chains for other industries as well. For example, pharmaceutical manufacturers can use this method to trace and track their medical products, customers, pharmacies, and distributors during a recall crisis. Therefore, the pharmaceutical company can handle the overall recall with better coordination, accuracy, and minimal impact.

We executed the four smart contracts on the Ethereum-based blockchain platform using solidity. Our solution is general and covers the broader aspects of the automotive product recall process. Therefore, it can be easily customized to be implemented on different blockchain platforms. Our solution enhances the existing product recall process in the automotive industry. Hence, automakers can adopt this solution to improve their current practices.

VII. CONCLUSION

In this paper, we have addressed the problem of tracking and automating automotive products in the existing automotive recall supply chains. We proposed a blockchain-based approach that enables the overall automotive recall management process in a decentralized, secure, trusted, transparent, and reliable manner. We developed four smart contracts to automate the process, record the events of the automotive recall supply chain, and maintain data provenance. Our system architecture, algorithms, sequence diagrams, and system testing approaches are general and applicable to different industrial use cases dealing with product recall management. We presented the cost and security analyses to show that our solution is cost-efficient and secure enough against cyberattacks and vulnerabilities. We compared our solution with blockchain and non-blockchain-based solutions to show the novelty of our proposed solution. In future work, we plan to deploy and test our solution on the real Ethereum network and build dApps for various stakeholders.

REFERENCES

[1] K. Hill, D. Menk, J. Cregger, and M. Schultz. (Jan. 20, 2020). Contribution of the Automotive Industry to the Economies of All Fifty States and the United States. Center for Automotive Research. Accessed: Jun. 14, 2021. [Online]. Available: https://www.cargroup.org/publication/contribution-of-the-automotive-industry-to-the-economies-of-all-fifty-states-and-the-united-states/

[2] S. Kapadia. (Feb. 20, 2018). Moving Parts: How the Automotive Industry is Transforming. Supply Chain Dive. Accessed: Jun. 14, 2021. [Online]. Available: https://www.supplychaindive.com/news/moving-parts-how-the-automotive-industry-is-transforming/516459/

[3] C. Chandra and A. K. Kamrani. “Knowledge management for consumer-focused product design,” J. Intel1. Manuf., vol. 14, no. 6, pp. 557–580, 2003, doi: 10.1023/A:1027358721819.

[4] Y. G. Gencer and U. Akkocuk, “Product recalls as an important category of reverse logistics,” DesTech Trans. Environ., Energy Earth Sci., 2016.

[5] K. Ahsan and I. Gunawan. “Analysis of product recalls: Identification of recall initiators and causes of recall,” Oper. Supply Chain Manage., Int. J., vol. 7, no. 3, pp. 97–106, Dec. 2014.

[6] M. Bernon, M. Bastl, W. Zhang, and M. Johnson. “Product recalls: The effects of industry, recall strategy and hazard, on shareholder wealth,” Int. J. Bus. Sci. Appl. Manage., to be published. Accessed: Jun. 15, 2021. [Online]. Available: https://epublications.marquette.edu/mgmt-fac/332/

[7] A. Aragon, U. Huber, T. Möller, and H. Nick. (Feb. 13, 2019). Return to Sender: Resolving the Automotive-Recall Resurgence. McKinsey & Company. Accessed: Jun. 14, 2021. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/return-to-sender-resolving-the-automotive-recall-resurgence

[8] C.-F. Chi, D. Sigmund, and M. O. Astardi, “Classification scheme for root cause failure mode and effects analysis (FMEA) of passenger vehicle recalls,” Rel. Eng. Syst. Saf., vol. 200, Aug. 2020, Art. no. 106929, doi: 10.1016/j.ress.2020.106929.

[9] K. Ahsan, “Trend analysis of car recalls: Evidence from the U.S. market,” Int. J. Manag. Value Supply Chains, vol. 4, no. 4, pp. 1–16, Dec. 2013, doi: 10.5121/jmvsc.2013.4401.

[10] (Aug. 27, 2012). Ford Cruise Control Deactivation Switch Recalls and History, The Center for Auto Safety. Accessed: Jun. 14, 2021. [Online]. Available: https://www.autosafety.org/ford-cruise-control-deactivation-switch-recalls-and-history/

[11] Capgemini. (Jul. 2017). Vehicle Recall Management—A Proactive Approach to Brand Loyalty and Happy Customers. Accessed: Jun. 14, 2021. [Online]. Available: https://www.capgemini.com/wp-content/uploads/2017/07/vehiclerecallmanagement.pdf

[12] N. G. Rupp and C. R. Taylor. “Who initiates recalls and who cares? Evidence from the automobile industry,” J. Ind. Econ., vol. 50, no. 2, pp. 123–149, Mar. 2003.

[13] P. N. Navin, “Vehicle recall management—Show customers you care,” LTI Let’s Solve, to be published.

[14] J. Ni and X. Huang, “Discovery-to-recall in the automotive industry: A postmortem-solving perspective on investigation of quality failures,” J. Supply Chain Manage., vol. 54, no. 2, pp. 71–95, Dec. 2017.

[15] S. Nassar, T. Kandil, M. E. Kara, and A. Ghadge, “Automotive recall risk: Impact of buyer-supplier relationship on supply chain social sustainability,” Int. J. Productiv. Perform. Manage., vol. 69, no. 3, pp. 467–487, Oct. 2019.

[16] Datapm. Recalls in the Automobile Industry—Creating A Million Dollar Dent. Accessed: Jun. 14, 2021. [Online]. Available: https://www.progress.com/docs/default-source/default-document-library/progress/documents/ebook auto recall acer.pdf

[17] M. T. Wynn, C. Ouyang, A. M. ter Hofstede, and C. J. Fidge, “Workflow support for product recall coordination,” Tech. Rep., 2009.

[18] P. Fragua-Lamas and T. M. T. M. Fernández-Caramés, “A review on blockchain technologies for an advanced and cyber-resilient automotive industry,” IEEE Access, vol. 7, pp. 17578–17598, 2019, doi: 10.1109/ACCESS.2019.2895302.

[19] P. Dutta, T.-M. Choi, S. Somani, and R. Butala, “Blockchain technology in supply chain operations: Applications, challenges and research opportunities,” Transp. Res. E, Logistics Transp. Rev., vol. 142, Oct. 2020, Art. no. 102067.

[20] K. Salah, M. Rehman, N. Nizamuddin, and A. Al-Fuqaha, “Blockchain for AI: Review and open research challenges,” IEEE Access, vol. 7, pp. 10127–10149, 2019, doi: 10.1109/ACCESS.2018.2895057.

[21] R. Beck, J. S. Ciepluch, N. Lollike, and S. Malone. “Blockchain-the gateway to trust-free cryptographic transactions,” in Proc. ECIS, May 2016, p. 153.

[22] M. A. Benatia, V. E. De Sa, B. Baudry, T. Möller, and M. F. Amin, “A framework for big data driven product traceability system,” in Proc. 4th Int. Conf. Adv. Technol. Signal Image Process. (ATISP), Mar. 2018, pp. 1–7, doi: 10.1109/ATISP.2018.8364340.

[23] B. Dai, Y. Xu, X. Xie, and J. Li, “Interactions of traceability and reliability optimization in a competitive supply chain with product recall,” Eur. J. Oper. Res., vol. 290, no. 1, pp. 116–131, Apr. 2021.

[24] Z. Shafi, S. Shahwan, N. A. Muhamed, H. Hashim, M. F. M. Amin, A. A. Rahim, Y. A. Aziz, K. Jusoff, and S. Z. Zaib, “Product recall management for halal product,” Middle-East J. Sci. Res., vol. 13, pp. 85–92, Jan. 2013.
PRATYUSH KUMAR PATRO received the B.E. degree in mechanical engineering from the College of Engineering Guindy, Anna University, India, in 2016. He is currently pursuing the master’s degree in engineering systems and management with Khalifa University, United Arab Emirates. Previously, he worked as a Supply Chain and BoM Engineer for new product launches at Ford Motor Private Ltd., India, for four years and two months and handled the import parts supply chain operations for various business geographies. In addition, he worked as a Research Intern at IJ Kakehashi Services Pvt. Ltd., India, for eight months and a certified six sigma green belt professional from ISI Delhi, India. He is enthusiastic about research, especially in Industry 4.0 technologies. His research interests include blockchain, the IoT, supply chain management, and project management.

RAJA WASIM AHMAD received the master’s degree in computer science from COMSATS University Islamabad (CUI), Abbottabad Campus, in 2011, and the Ph.D. degree in computer science from the Center for Mobile Cloud Computing Research (C4MCCR), Faculty of Computer Science and Information Technology (FSKTM), University of Malaya, Malaysia. Currently, he is working as a Postdoctoral Research Fellow at Khalifa University, Abu Dhabi, United Arab Emirates. He has published several research papers in leading journals and conferences. Most of his research contributions are published in top-cited journals, such as IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE SYSTEMS JOURNAL, IEEE ACCESS, Journal of Network and Computer Applications, Journal of Grid Computing, Cluster Computing, International Journal of Communication Systems, International Journal of Information Management, Journal of Systems and Software, The Journal of Supercomputing, IET Intelligent Transport Systems, and Renewable and Sustainable Energy Reviews. His research work is well acknowledged in national and international conferences. One of his research articles has received the Best Paper Award in SPECS Symposium held in France, in 2018. He is also serving as a reviewer for several journals and conferences.

IBRAR YAQOOB (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017. He is a Research Scientist at the Department of Electrical Engineering and Computer Science, Khalifa University, United Arab Emirates. Previously, he worked as a Research Professor at the Department of Computer Science and Engineering, Kyung Hee University, South Korea, where he completed his postdoctoral fellowship. He worked as a Researcher and a Developer at the Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya. His numerous research articles are very famous and among the most downloaded in top journals. He has been listed among top researchers by Thomson Reuters (Web of Science) based on the number of citations. He has been involved in a number of conferences and workshops in various capacities. His research interests include big data, blockchain, edge computing, mobile cloud computing, the Internet of Things, healthcare, and computer networks. He is currently serving/served as an guest/associate editor in various journals.

KHALED SALAH received the B.S. degree in computer engineering with a minor in computer science from Iowa State University, USA, in 1990, and the M.S. degree in computer systems engineering and the Ph.D. degree in computer science from the Illinois Institute of Technology, USA, in 1994 and 2000, respectively. He is a Full Professor at the Department of Electrical and Computer Engineering, Khalifa University, United Arab Emirates. He has over 220 publications and three U.S. patents, has been giving a number of international keynote speeches, invited talks, tutorials, and research seminars on the subjects of blockchain, the IoT, fog and cloud computing, and cybersecurity. He is now leading a number of projects on how to leverage blockchain for healthcare, 5G networks, combating deepfake videos, supply chain management, and AI. He served as the Chair of the Track Chairs for IEEE Globecom 2018 on Cloud Computing. He is an Associate Editor of IEEE BLOCKCHAIN TECH BRIEFS and a member of IEEE Blockchain Education Committee.

RAJA JAYARAMAN received the bachelor’s and master’s degrees in mathematics in India, the Master of Science degree in industrial engineering from New Mexico State University, and the Ph.D. degree in industrial engineering from Texas Tech University. He is an Associate Professor with the Department of Industrial and Systems Engineering, Khalifa University, Abu Dhabi, United Arab Emirates. His expertise is in multi-criteria optimization techniques applied to diverse applications, including supply chain and logistics, healthcare, energy, environment, and sustainability. His research interests include using blockchain technology, systems engineering, and process optimization techniques to characterize, model, and analyze complex systems with applications to supply chains, maintenance operations planning, and healthcare delivery. His postdoctoral research was centered on technology adoption and implementation of innovative practices in the healthcare supply chains and service delivery. He has led several successful research projects and pilot implementations in the area of supply chain data standards adoption in the U.S. healthcare systems. His research has appeared in top rated journals, including Annals of Operations Research, ISE Transactions, Energy Policy, Applied Energy, Knowledge-Based Systems, IEEE ACCESS, Journal of Theoretical Biology, and Engineering Management Journal.

VOLUME 9, 2021
167775