International Conference on Smart Data Intelligence
(ICSMDI 2021)

Blockchain and AI in Pharmaceutical Supply Chain

Sherwyn D'souza[a], Darlene Nazareth[b], Cassia Vaz[c], Prof. Monali Shetty[d]

Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Bandra-West, Mumbai, 400050, India
[a]sherwyndsouza1999@gmail.com
[b]darlenenaz99@gmail.com
[c]cassiahvaz@gmail.com
[d]monali_shetty@fragnel.edu.in

A B S T R A C T

At present, counterfeit drugs pose a serious threat as it is difficult for people to know the true value of purchased medicines due to a significant lack of transparency in the current system. Also, tampering within the supply chain is difficult to investigate when suspicion of illegal or unethical practices. Our solution is an amalgamation of two powerful technologies - Blockchain and AI. Blockchain is an open, distributed ledger that can efficiently record transactions between two parties in a verifiable and permanent way. Since blockchains are decentralized, distributed, transparent, and immutable, they can easily solve counterfeit medicines. AI in pharmacology helps improve customer service, loyalty and enables easy access to blockchain-based medical intelligence. This paper proposes a system that uses blockchain and AI for the safe supply of medical drugs throughout the supply chain. Each product within the chain can be transferred between authenticated entities of the chain using an event request-response mechanism. All transactions between entities are recorded into the blockchain using smart contracts with the help of which a product can be traced to its source. We built a Rasa chatbot integrated into a Flutter app enabling ordering, tracing back medicines, and enhancing blockchain-based credit evaluation. A DApp was then developed using React Framework. The smart contracts were deployed on a local blockchain provided by Ganache. Using Web3.js and Truffle framework, DApp is connected to the blockchain. The experimental results show that our solution is feasible and comparatively more secure than existing systems.

Keywords: Blockchain, Ethereum, Smart contracts, Product traceability, Supply chain, Drug counterfeiting, Artificial intelligence, Rasa

1. Introduction

The pharmaceutical supply chain network is the mechanism by which manufactured prescription medications are distributed to patients. However, this supply chain is very complex and consists of multiple stages, which could span over months or longer, across multiple regions around the world. A primary supply chain consists of many entities like suppliers, manufacturers, transporters, wholesalers, distributors, retailers, etc. Thus, keeping track of each medical drug throughout the chain and tracing it back to its source becomes a tedious job. Drug counterfeiting is a global concern. The Health Research Funding organization reported that nearly 10-30% of the drugs are fake in developing countries. Counterfeit products are a matter of great concern as they can produce different side effects to human health. According to the World Health Organization, about 30% of all medication sold in Africa, Asia, and Latin America is counterfeit. The key problem is not counterfeiting, but rather the fact that, as opposed to real medications, counterfeit drugs have different side effects on human health.

Due to lack of transparency in the current system, it is extremely difficult for customers or buyers to know the value of the products. It is also very difficult to investigate the tampering within the supply chain when there is suspicion of illegal or unethical practices. Hence such supply chains are highly inefficient as vendors, suppliers, etc. try to establish a link among the entities and identify who needs what, when and how. Customers and buyers are currently unable to determine the true worth of goods due to a major lack of clarity in the current system. When there is evidence of improper or immoral activities, this tampering within the chain is exceedingly difficult to investigate. They can also be unreliable as retailers and manufacturers struggle to figure out who wants what, when, why, and how. Blockchain is a revolutionizing solution. Blockchain provides a distributed hyperledger with no centralized authority over the system. Each transaction into the blockchain is immutable, which means there is no way sensitive data like Drug/Customer information can be tampered with. Blockchain provides complete transparency, which also brings trust between the various main entities of the Supply Chain, such as, Manufacturers, intermediaries like Distributors and Suppliers, and the end-users like Customers/Retailers/Hospitals. Each product within the chain can be transferred between the different authenticated entities of the chain using an event request-response mechanism. All transactions between the different entities are recorded into the blockchain using smart contracts.

For any business, customer experience is their top-most priority, and thus happy and satisfied customers give them better returns. Hence with the use of Artificial Intelligence, companies improve their customer service, customer loyalty and brand reputation which thereby enables employees to focus on their higher-value tasks and get great returns. AI powered bots can proactively start conversations with customers, provide relevant information, and help each touchpoint throughout the entire customer life cycle. It therefore enables customers to fetch the answers that they need, without any dependency on the company's employees, which thus improves customer satisfaction. The proposed system does this job with the help of the Rasa framework. Rasa is a framework for developing AI-powered, industrial-grade chatbots. It is incredibly powerful and is used by developers worldwide to create chatbots and contextual assistants. The chatbot is then integrated with an app developed using the Flutter framework. Flutter helps build cross-platform (Android/iOS) mobile applications. Customers can communicate with the chatbot to place orders, track their order, find the source of their order, etc., using the mobile app.

Finally, a DApp (Decentralized App) is developed using the React Framework. The smart contracts are deployed on a private local blockchain provided by Ganache [16]. Using Web3.js and the Truffle framework, the DApp is connected to the blockchain.

The rest of the paper is organized as follows. Section II states the review of literature, and Section III introduces the preliminaries used in this paper. In Section IV, the system architecture is described in detail. The smart contract's concrete implementation is exhibited in Section V, and the experiment and evaluation are discussed in Section VI. Finally, the conclusion is presented.

2. Review of Literature

Over the years, many scholars and entrepreneurs have made many discussions and research on how to combine blockchain technology to improve and manage the current situation of the supply chain.

G. Perboli et al. [1] describes the standard methodology for developing and validating the overall Blockchain solution and designing a strategy for integrating it into Business Strategy. They also highlight how blockchain can tackle security challenges in IoT, such as identifying different devices and managing trust, information tracking, authentication and access control, and accountability in IoT-based applications.

Jamil et al. [3] proposed an application based on a user service framework that uses smart contracts and a distributed ledger as middleware. The proposed system is based on a permissioned blockchain that allows only valid participants to participate and enroll in the blockchain network, thus separating it from the other blockchain-based systems. The proposed smart contract-based application uses the transaction and executes several queries and updates the ledger state by appending the transaction blocks, and returns the updated result as a response to the application.

Q. Zhu et al. [5] focus on linking the product deletion decision-making process, which is a process of deleting a product and all its related information from the company's portfolio with Blockchain technology. The four main stages in the product deletion process have been proposed: recognition, analysis and revitalization, evaluation and decision-making, and implementation. Blockchain technology improves communication and collaboration among various entities in the supply chain and thus increases information efficiency, effectiveness and reduces conflicts.

Prachi Shrikant et al. [6] focuses on the importance of Blockchain technology to trace and detect counterfeit products in the supply chain. Every time the medicinal drug moves from one entity to another, the information stored in the blockchain makes it easy to track the drug, and thus the threat of counterfeit products is reduced. Blockchain technology helps solve the main issues, as each new transaction is time-stamped, enabling companies to track their products in the supply chain. Allowing stakeholders to take actions in case of any issues by identifying the exact location of the drug.

K. Toyoda et al. [7] proposed the need to develop anti-counterfeit systems that will work when the RFID tag's information will be cloned in the post supply chain. Hence through the paper, they proposed The Possession of Products scheme. The counterfeits can be detected if any entity is unable to prove the possession of the particular product. They have suggested blockchain, as Bitcoin allows users to prove their ownership without the need for any authentication and centralized authority.

Huang et al. [8] narrate the importance of Drugledger, a fully scenario-oriented blockchain system for drug traceability and regulation. Drugledger uses the UTXO-based transaction model combined with the supply chain to construct the entire workflow that includes drug packaging, repackaging, unpackaging and drug transaction cancellation, the arrival and exit of the drug supply chain, and so on. Thus, the Drugledger manages to separate the drug traceability service from data modification, ensuring data authenticity and privacy.

J. Ma et al. [9] describe the need for a fully-functional blockchain system to prevent product counterfeiting to ensure that the products can be identified and traced in the supply chain. They propose using Ethereum, a Blockchain platform and suggest that users write smart contracts using Solidity and deploy them on the network. Through the proposed anti-counterfeit system, the users do not have to be concerned regarding acquiring counterfeit products.

Leng et al. [11] proposed a public Blockchain with a double chain architecture to enhance agricultural supply chain systems' efficiency. They showed that their solution provides adaptive rent-seeking and matching mechanisms for public service platforms. It guarantees the transparency and security of transaction information and privacy of enterprise information. The main drawbacks are the size of the underlying Blockchain network and the related performance issues.

Mao et al. [12] proposed a Blockchain-based credit evaluation system to strengthen supervision and management effectiveness in the food supply chain. In particular, they gather credit evaluation text from traders by smart contracts on the blockchain. Further, the text is analyzed by a deep learning method named Long Short-Term Memory (LSTM). The system's drawback is that they show the effectiveness of their method, but they do not consider the overall system costs and benefits explicitly.

R. Kumar et al. [2] proposed a framework that represents a blockchain-based secure infrastructure for the medical supply chains among valid participants, which can provide drug security and the authenticity of the manufacturer. It is based on PKI and digital signatures to prevent replay and man-in-the-middle attacks.

Hackius et al. [4] have described four use cases of blockchain in logistics, and SCM explored both theoretically and practically. The use cases are easy paperwork processing in ocean freight, identifying counterfeit products, facilitating origin tracking, and operating the Internet of Things.

Q. Ding et al. [10] have presented a chatbot to search the price, the cap, and the volumes of stock based on RASA NLU using iex-finance API. The proposed system uses a spaCy model for a language known as en_core_web and the RASA NLU pipeline spacy_sklearn that uses pre-trained word vectors from GloVe. The general steps proposed in the system are as follows: sending a message to the financial chatbot, analysis of the sentence, fetching stock information from iex-finance API, returning the possible intent of the message chatbot received by regular expression and keywords, and responding to the messages according to the intents and current states based on the state machine.

3. Preliminaries

This section reviews some of the relevant background knowledge and detailed concepts that will be used throughout this paper.

3.1. Blockchain

Blockchain is a new technology that is gradually emerging with the increasing popularity of digital currencies such as Bitcoin. It is essentially a distributed ledger database. Blockchain records transactions that have occurred by establishing a database maintained by all network nodes and the entire process is open, transparent, and irreversible. According to the participants, it can be divided into the public blockchain, consortium blockchain, and private blockchain, and the consortium blockchain and the private blockchain are collectively referred to as the permissioned blockchain. The public blockchain is completely decentralized in the real sense. At any time, any node will join or leave the decision to build a new block. In the permissioned blockchain, though, the decision to build a new block is made by certain trusted nodes. It has been applied to copyright management, identity authentication, and data storage services.

3.2. Ethereum and Smart Contracts

Bitcoin's universality is low since it is built only for virtual currency scenarios and is not Turing-complete. As a result, several other blockchain-based technologies have evolved, and various kinds of software will now be represented on the blockchain as smart contracts. Ethereum first realized the complete fit of blockchain and smart contracts. If a smart contract is deployed on the blockchain, it would be implemented in accordance with predefined rules, and no one will be able to alter it. The smart contract in Ethereum is written in a stack-based low-level bytecode language called EVM code, which is executed by the Ethereum Virtual Machine.

3.3. Rasa

Rasa is an open-source machine learning framework for automated text and voice-based conversations. Rasa can be used to understand messages, hold conversations, and connect to messaging channels and APIs. Following are the steps of how an assistant built with Rasa responds to a message:

1. The received message is routed into the interpreter, which translates it into a dictionary of the original text, intent, and entities discovered. This is handled by NLU.
2. The tracker object records the current state of the conversation. It receives information that a new message has come in.
3. The policy receives the tracker's current state.
4. The policy determines what action to do next.
5. The tracker logs the chosen action.
6. A response is sent to the user.

4. System Architecture

This section introduces the system model of the product traceability process, which is realized by the decentralization and data immutability of blockchain technology. In the process, any node can have both the attributes of demand and supply. In the product transaction formation process, an event response mechanism is designed in order to ensure that both parties of the transaction agree on the receipt and delivery of medical drugs. The transaction data will be permanently stored in the blockchain using smart contracts. Furthermore, we design a mobile application that is integrated with the Rasa chatbot for customer service.

4.1. System Overview

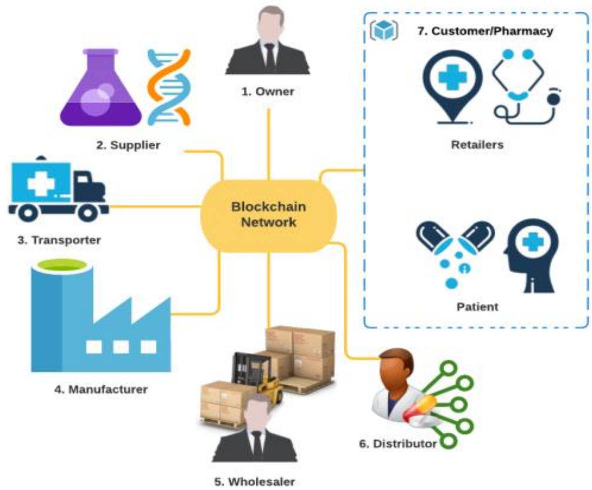

Fig. 1 - Overview of the Proposed Architecture.

As shown in Fig. 1, the proposed system consists of many entities like suppliers, transporters, manufacturers, wholesalers, distributors, and customers/retailers and is connected through a decentralized network. Each of the above entities of the supply chain is a node on the public blockchain. Each of these nodes has its own Ethereum account, which is used for representing its identity. The specific roles and functionalities of each are discussed below:

1. Owner
   a. CREATE a new user to be added to the chain.
   b. READ the information of any user.
   c. UPDATE the roles of a user.
   d. DELETE a user from the chain.
2. Transporter
   a. Verify the package (Raw Material or Medicine).
   b. Pick the package from an entity (based on transporter type).
   c. Deliver the product to an entity.
3. Supplier
   a. CREATE a Raw Material.
   b. GET the addresses of the Raw Materials created.
4. Manufacturer
   a. Receive the Raw Material from the Supplier through the Transporter.
   b. Verify the source of the product received.
   c. CREATE a new Medicine using received raw materials.
5. Wholesaler
   a. Receive the medicine from the manufacturer through the Transporter.
   b. Verify the source of the medicine.
   c. Transfer the ownership of the medicine.
6. Distributor
   a. Receive the medicine from the Wholesaler through the Transporter.
   b. Verify the source of the medicine.
   c. Transfer the ownership of the medicine.
7. Customer
   a. Receive the medicine from the Distributor through the Transporter.
   b. Verify the source of the medicine.
   c. Place orders using the Rasa chatbot.
   d. Get medical drug information.

4.2. Smart Contract Design

Supply Chain Contract: This contract is deployed by the Owner of the chain. It consists of many entities associated with the supply chain, i.e., Owner, Supplier, Transporter, Manufacturer, Wholesaler, Distributor, Customer. It also consists of various Solidity events used to communicate with the front end in real-time. Each function in the contract can only be accessed by its respective role assigned to it. This is done with the help of "modifiers" in Solidity. Thus, no entity without a particular role can access a specific function. This helps to increase the security and accessibility of data stored or queried from the blockchain.

Raw Material Contract: A respective Supplier deploys the Raw Material Contract. Once a raw material is created physically, it is then added to the chain by the supplier that created the raw material. While creating a raw material to be added to the chain, data such as EA (Ethereum Address) of the Supplier, DateTime, EA of Transporter, Transaction Contract Address, etc. are requested from the supplier. It also contains events that can compute the whereabouts of the package in real-time. The EA of Receiver (Manufacturer) is later updated based on the event request-response mechanism. It also stores the current status of the medicine, i.e., which entity currently has the raw material.

Medicine Contract: The respective manufacturer deploys the Medicine Contract. Once a medicine is created physically, it is then added to the chain by the manufacturer that created the medicine. While creating medicine to be added to the chain, data such as EA (Ethereum Address) of Raw Material used to create medicine, DateTime, EA of Transporter, Transaction Contract Address, etc., is requested from the manufacturer. It also contains events that can compute the whereabouts of the package in real-time. The EA of Wholesaler, EA of Distributor, and EA of Customer are updated later based on the event request-response mechanism. It also stores the current status of the medicine, i.e., which entity currently has the package.

Transaction Contract: The Transaction Contract is deployed automatically by the Raw Material and Medicine smart contracts whenever created. The contract takes data such as DateTime, sender EA, receiver EA, location, transaction hash, and the hash of the previous transaction. The transaction hash is 32 bytes. The previous transaction hash is stored for entities to verify the source of products in the chain—an example of transaction data in the smart Transaction contract.

Table 1 - Example of data in Transaction contract.

TxnHash	From	To	Prev. TxnHash	Lat	Lng	Timestamp
0xf3...e0	0xCc...06	0x72...9D	0xf3…e0	19.07	72.81	1616358305
0xd7...5e	0xCc...06	0x9a...64	0xf3...e0	18.52	72.85	1616651750

4.3. Product Traceability and Source Verification

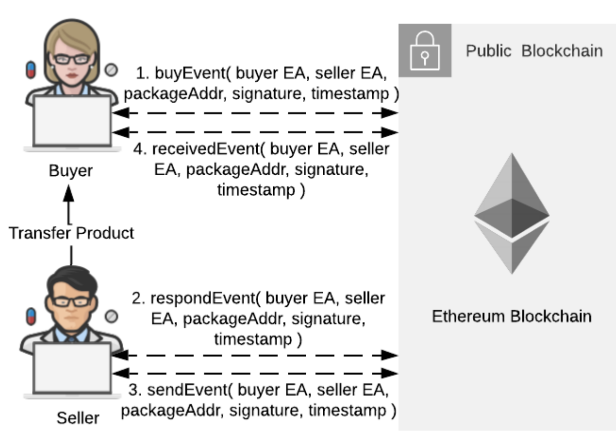

Fig. 2 - Event Request-Response Mechanism.

As shown in the above Fig. 2, the process of agreeing on the delivery and receipt of goods by both parties of the transaction is shown using the event request-response mechanism [13]. It is discussed in detail below.

1. First, the buyer initiates a purchase request. The buyEvent() event in the Supply Chain contract is then triggered. The event includes the buyer's and seller's Ethereum address (Buyer EA and Seller EA), the address of the raw material/medicine to be purchased, along with the signature which is signed with the private key of the requester(buyer) and timestamp of the request. The signature is sent along in the events to confirm the identity of both parties and the authenticity of the request. The Seller addresses are indexed so that each seller can query their records based on their Ethereum Address.
2. Then, the Seller queries the log records related to himself according to his Ethereum address and verifies the validity of the signature contained in the events. If the verification is passed, an event respondEvent() is triggered by the seller to respond to the buyer's request along with a signature which is signed with the private key of the seller.
3. Next, the Seller sends the product to the buyer through the Transporter. An event sendEvent() is triggered to prove that the raw materials/medicines have been shipped, including the Seller's and buyer's Ethereum address (Seller EA and Buyer EA), the product address, along with the signature which is signed with the private key of the Seller and timestamp of the transfer of the product.
4. Finally, an event receivedEvent() is triggered by the buyer upon receipt of the goods to certify that the goods have been received.

For example, suppose the manufacturer requires a raw material to create new medicine. In that case, the manufacturer acts as the buyer, and the supplier, which requires the raw material, acts as the seller. Once the above process is completed, the Supplier will update the transaction information according to the product address in the corresponding Transaction contract, and the new receiver of the raw material is updated in the Raw Material contract. The system assumes that only after both parties of the transaction have truthfully activated the above events, the transaction details will be changed. The product's source will be regarded as trustworthy.

5. System Implementation

5.1. Blockchain Implementation:

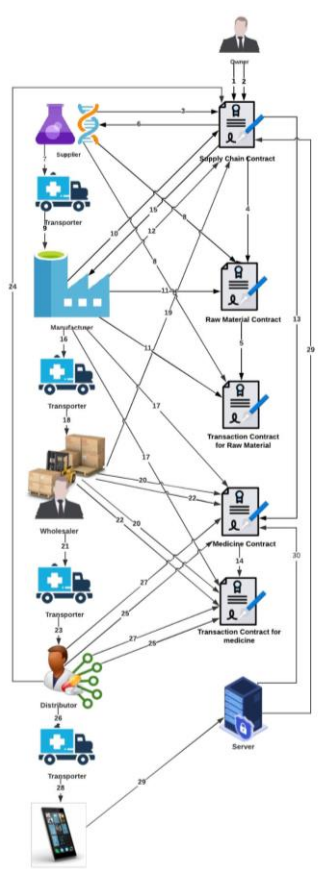

Fig. 3 - Process Flow Diagram.

Table 2 – Process Flow Table.

Sr. No	Explanation
1.	Owner deploys the smart contracts to the Ethereum Blockchain.
2.	The Owner authenticates and registers the entities of the chain.
3.	Supplier registers a new Raw Material.
4.	Raw Material Contract is deployed for the newly created raw material.
5.	Corresponding Transaction Contract is also deployed for the newly created raw material.
6.	Raw material registered successfully.
7.	Supplier transfers the raw material to the Transporter.
8.	Supplier updates the product status and creates a Transaction in the Transaction Contract.
9.	Transporter transfers the raw material to the manufacturer.
10.	Manufacturer verifies the source of the raw material.
11.	Manufacturer updates the product status and creates a Transaction in the Transaction Contract.
12.	Manufacturer registers a new medicine.
13.	Medicine Contract is deployed for the newly created medicine.
14.	Corresponding Transaction Contract is also deployed for the newly created medicine.
15.	Medicine registered successfully.
16.	Manufacturer transfers the raw material to the Transporter.
17.	Manufacturer updates the product status and creates a Transaction in the Transaction Contract.
18.	Transporter transfers the raw material to the Wholesaler.
19.	Wholesaler verifies the source of the raw material.
20.	Wholesaler updates the product status and creates a Transaction in the Transaction Contract.
21.	Wholesaler transfers the raw material to the Transporter.
22.	Wholesaler updates the product status and creates a Transaction in the Transaction Contract.
23.	Transporter transfers the raw material to the Distributor.
24.	Distributor verifies the source of the raw material.
25.	Distributor updates the product status and creates a Transaction in the Transaction Contract.
26.	Distributor transfers the raw material to the Transporter.
27.	Distributor updates the product status and creates a Transaction in the Transaction Contract.
28.	Distributor transfers the raw material to the Customer.
29.	Customer verifies the source of the medicine through a temporary node.
30.	Customer updates the status of the product through a temporary node.

The proposed system is discussed in this section. The truffle framework [14] helps in better the compilation and deployment of smart contracts. Truffle is a development tool for the Ethereum Solidity language and provides a test framework, making DApp development easier. The test environment for the proposed solution is listed in the above Table. Visual Studio Code is a powerful, free, and open-source text editor that can build a solidity IDE by downloading some plugins. Web3.js [15] is a JavaScript library provided by Ethereum. It offers a complete set of JavaScript objects and functions that help the client side interact with the blockchain. We compile and migrate our contracts, which gives us a build version of our contracts that integrates the contract abstraction into the script to make use of the contract abstraction provided by truffle directly in the JavaScript code. The key to DApp development is to write the smart contracts and interact with them through the front-end web pages. The code for the front-end, along with the connection establishment and interaction logic, is written in JavaScript with the ReactJS framework's help. Users can interact with contracts through the UI to use the different functionalities provided by the DApp.

Table 3 – System Test Environment.

Software	Version
Solidity	0.6.6
Ganache	2.5.3
Truffle	5.1.58
Metamask	8.1.9
Web3	1.3.1
ReactJS	17.0.1

5.2. Rasa Implementation:

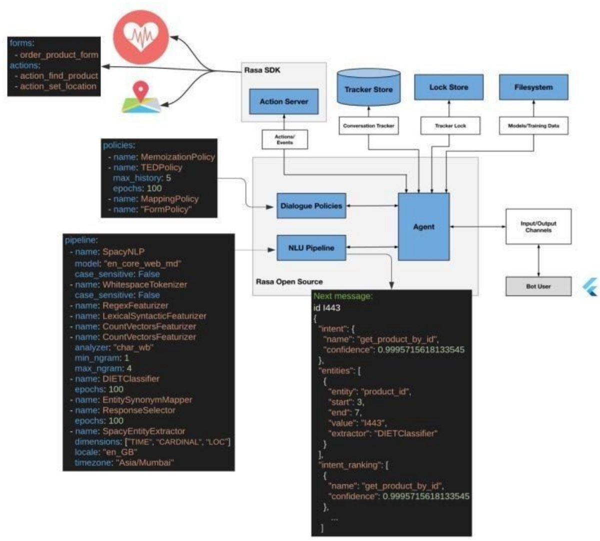

Fig. 4 – Rasa Implementation.

Fig. 4 provides an overview of the Rasa Open-Source architecture. NLU Pipeline processes user utterances using an NLU model that is generated by the trained pipeline. Based on the context, the dialogue management aspect defines the next action in a conversation. This is depicted in the diagram as the Dialogue Policies. The pipeline used consists of the following components:

1. **Spacy core model:** This is a pre-trained model to predict named entities, assign part-of-speech tags (POS) and parse syntactic dependencies.
2. **Whitespace Tokenizer:** Creates a token for every whitespace-separated character sequence used for intents and responses
3. **Regex Featurizer:** A feature will be set marking whether a *regex* in the training dataset was found in the user's message or not.
4. **Lexical Syntactic Featurizer:** This takes an input of tokens and creates features for entity extraction.
5. **Count Vector Featurizer:** Converts collection of text documents to a sparse matrix of token counts. It creates features for message intent classification and response selection.
6. **DIET Classifier:** DIET (Dual Intent and Entity Transformer) is a multi-task architecture that can handle both intent classification, and entity recognition together predicted through a Conditional Random Field (CRF) tagging layer. It generates an output of entities, intent, and intent ranking.
7. **Entity Synonym Mapper:** This component will ensure that the detected training entity will be mapped to the same synonym value.
8. **Response selector:** Used to predict bot's response from a set of candidate responses. It embeds user inputs and response labels into the same space.
9. **Spacy Entity Extractor:** This component predicts the entities of a message.

Fig. 5 – Rasa Interactive Stories.

The model is trained on more than 12k web scraped and preprocessed drug names. This web scraping script is written using Selenium and Beautiful Soup. When a customer requests for a drug name similar to the ones in this dataset, the model extracts the correct medicine name entity with the help of DIET Classifier along with Regex Featurizer and queries a list of similar drugs and details like its manufacturer name, components, etc.

The mobile app is built using Flutter, which is a hybrid app development framework that can create apps for both Android and iOS at the same time. This is connected to our rasa bot using sockets, and messages are streamed to and from the application to the server. Each message is then sent to the server, and the rasa bot determines the intent from the text. A relevant message is then returned to the user based on the intent. Rasa Chatbot main Features Implemented are:

1. **Form for Ordering Medicines:** Customers can select the drug and specify the quantity needed. Based on the users' inputs and validity check, the order is placed, and a bill is generated. These order details and a timestamp are then stored on the firebase cloud firestore while the *user's* conversation history and trackers are stored in MongoDB.
2. **Setting users' nearby location:** Customers can set their nearby location by text. This generates a query to fetch and store their location coordinates in the database.
3. **Trace orders:** Customers can scan the ID (address) of the purchased medicine to know if it is a verified drug and view its supply chain transaction history, *tracing* it back to the source of even its component raw materials used in the making of that drug. Thus, allowing pharmacies/customers to know the true value of the medicine.

6. Results

6.1. Experimental results of a smart contract deployment

The smart contracts were deployed on a local blockchain provided by Ganache [16]. The contracts and the gas cost for deploying them are displayed in the table below

Table 4 – Smart contract deployment cost

Contract	Gas Cost	Actual Cost (Ether)
Supply Chain contract	5055356	0.10110712 ETH
Raw Material contract	1015651	0.00142702 ETH
Medicine contract	1405550	0.0023604 ETH
Transaction contract	574042	0.0012762 ETH

Analysis of the throughput and latency of blocks mined in the blockchain is shown in Fig. 6.

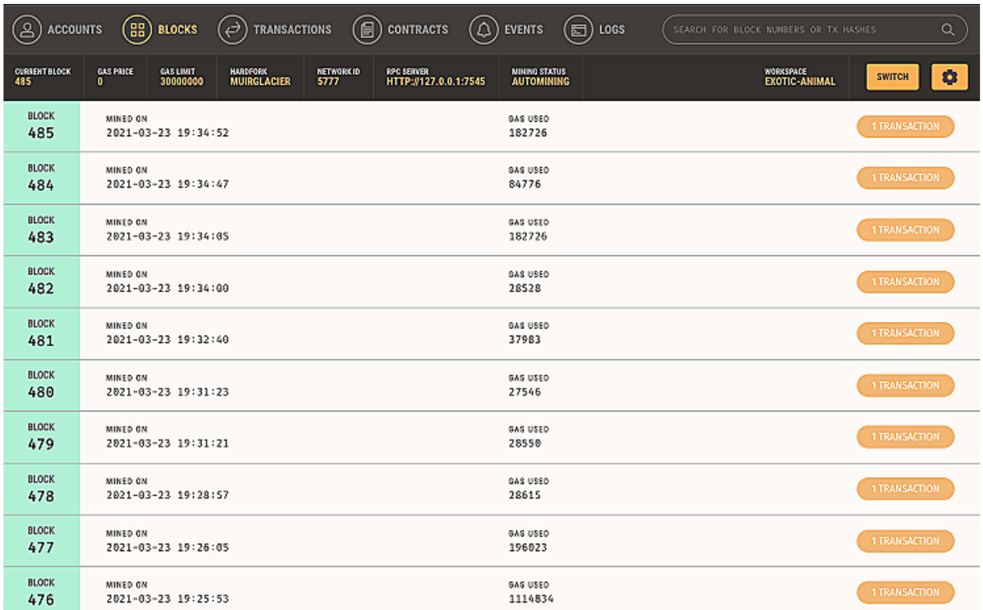

Fig. 6 – Analysis of local blockchain on Ganache GUI.

To check the performance of the system that deployed on Ethereum and accessed through Ganache, capacity is calculated using TPS (Transaction per second).

$$Throughput = capacity = TPS = (block\ size) * (block\ time)(Note: considered\ average\ values) \quad (1)$$

$$TPS = (transactions\ per\ block) * (block\ per\ second) \quad (2)$$

So, average TPS ranges between 0.01 to 0.209

Network's average latency rate is also referred to as block time, which is the time required to generate the next block of transactions in the chain. So, it is the amount of time a user has to wait after execution of transaction.

So, average latency for the proposed system is 0.01 sec to 1.07 seconds

The contracts are deployed on a local blockchain provided by Ganache [16] which is part of the Truffle suite. The front-end (client) app is connected to the blockchain using Web3.js.

As shown in Fig. 7, we can see a raw material being created by the chain's Supplier entity. Once a raw material is created, its details are added to the blockchain, and it is assigned an address. A new Transaction contract is deployed, and this contract address is stored in the Raw Material smart contract of the created raw material.

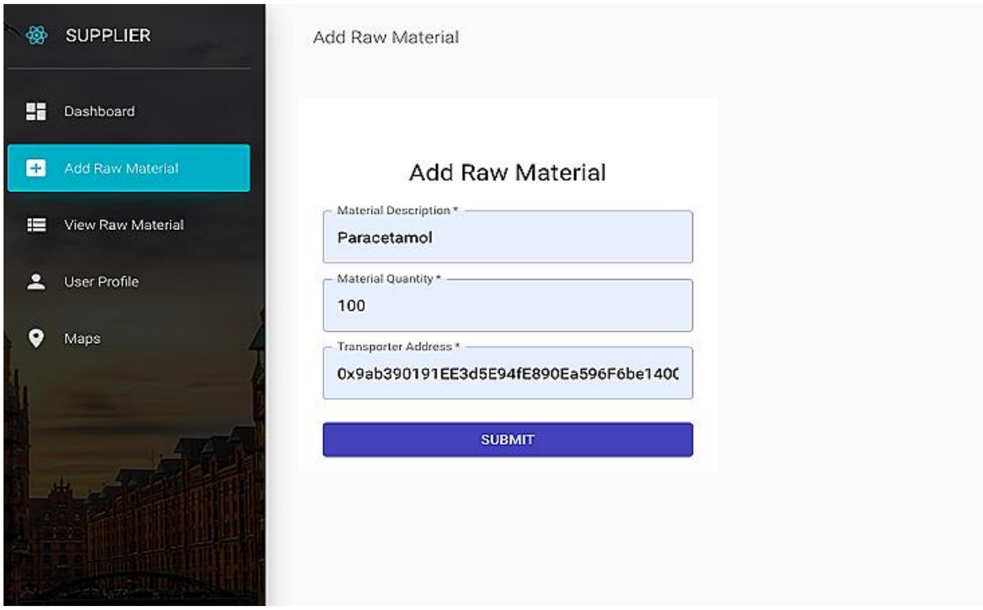

Fig. 7 – Creating a Raw material.

The Manufacturer entity can then request raw material by choosing the raw material he is interested in buying. Once he clicks the "Submit" button, the event request-response mechanism starts between the Manufacturer (Buyer) and Supplier (Seller). Once the raw material buyer is confirmed, the Raw Material contract is updated with the new buyer's address. The Transporter can then pick up the product, and the supplier hands the package to the Transporter, and a new transaction is then recorded in the Transaction contract of the raw material. When the Manufacturer entity receives the raw material, he verifies the signature associated with the transaction and then creates a new transaction in the Transaction contract. Thus, any further entity can view the transaction contract's records and verify the origin of the product, as shown in Fig. 8. The supplier can also view that the product reached the manufacturer by viewing the status, which updates in real-time on the Raw Material details page (Fig. 9). The same response mechanism is used to transfer the products between different entities throughout the chain.

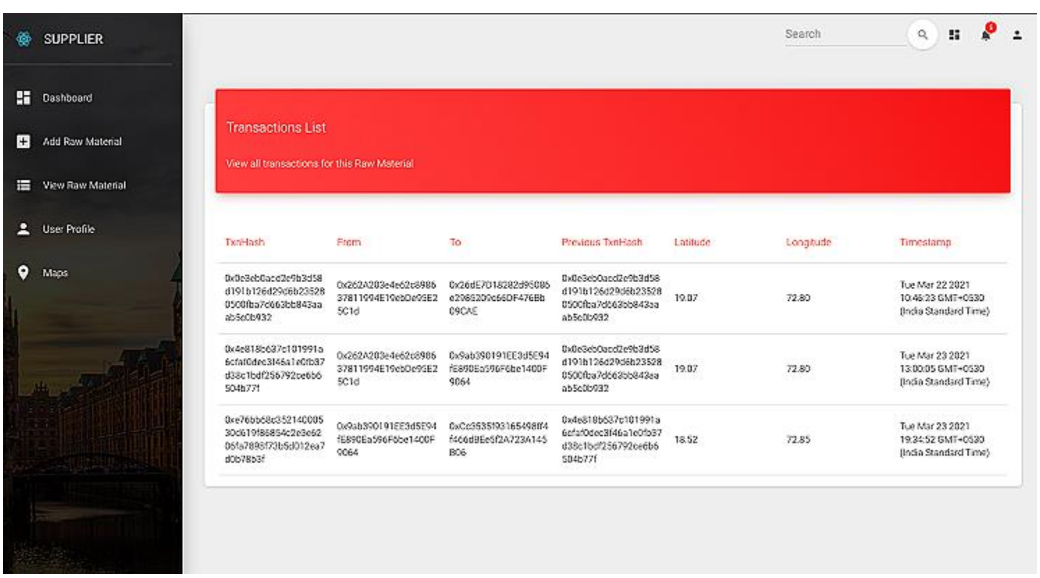

Fig. 8 – Transaction's Page.

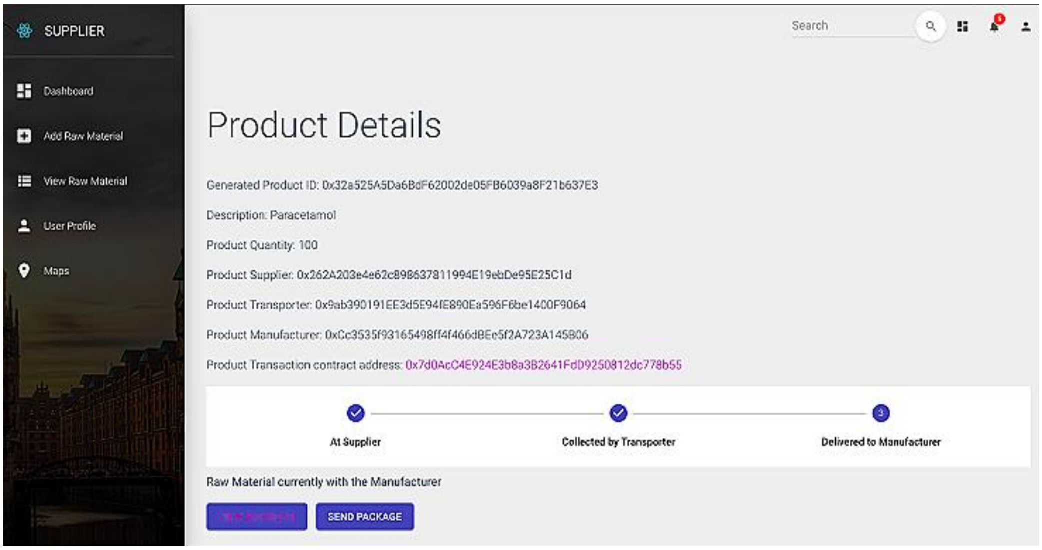

Fig. 9 – Raw Material details page.

6.2. Experimental results of Rasa Chatbot

Table 5 – RASA NLU Report

Weighted Average	Metrics		
	Precision	**Recall**	**F1-score**
Entity Extractor	0.97	1.0	0.98
Intent Classifier	1.0	1.0	1.0

In Table 5, the rasa test NLU report shows the weighted average scores of entity extractor (DIET classifier) and intent classifier's precision, recall, and f1-scores.

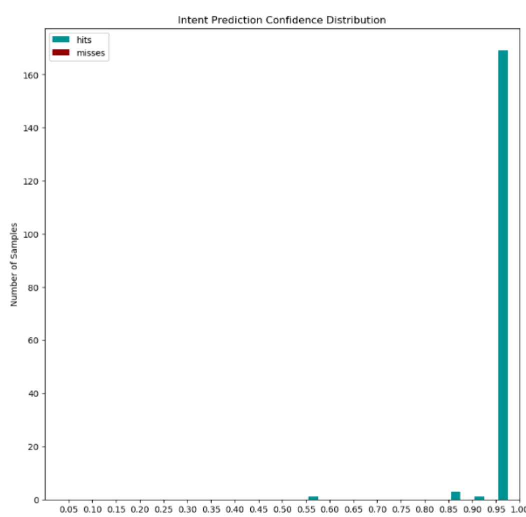

Fig. 10 – Intent Prediction Confidence Distribution.

With the above histogram plot (Fig. 10), we can visualize all predictions' confidence, the correct and incorrect predictions displayed by green and red bars, respectively. The confusion matrix results shown below highlight that no intents were mistaken for others by our NLU model.

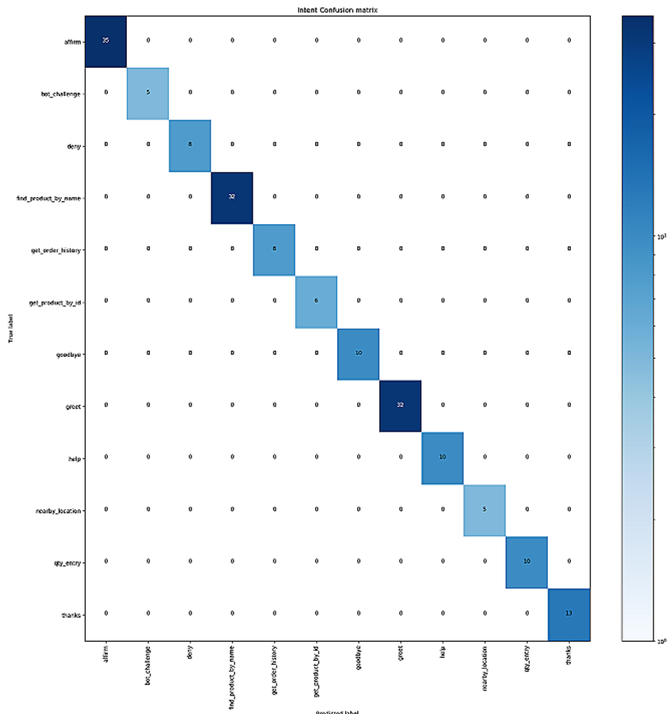

Fig. 11 – Intent Confusion Matrix.

7. Conclusion

The paper proposes a smart anti-counterfeit drug supply chain system based on blockchain technology and artificial intelligence. By using smart contracts and product registration and transferring, all product transferring records are permanently registered in the unchangeable ledger. The collaboration of smart contracts allows Tracking of the products. Consumers can also participate in the process of maintaining information flows. The proposed system has prominent decentralized characteristics, which significantly reduces the possibility of privately tampering with data. The Rasa chatbot handles improved customer service and transparency of drug details, order movements down the supply chain.

Furthermore, an event request-response process was created to verify the identity of all parties and the signature found in the event in order to decide if the event is authentic. All events can be recorded and stored in the blockchain as a log, which can be viewed in real-time. Finally, a decentralized application (DApp) was built based on the Truffle framework, deploy the smart contract, test the contract code through a local blockchain provided by Ganache, and implement a decentralized web app interaction interface based on the prototype. The system is characterized by data accessibility, tamper-proofing, and resistance to man-in-the-middle attacks.

References

[1] G. Perboli, S. Musso and M. Rosano. "Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases". in IEEE Access. vol. 6.pp. 62018-62028, 2018. DOI: 10.1109/ACCESS.2018.2875782.

[2] R. Kumar and R. Tripathi. "Traceability of counterfeit medicine supply chain through Blockchain." 2019 11th International Conference on Communication Systems & Networks (COMSNETS). Bengaluru, India. 2019. pp. 568-570. DOI: 10.1109/COMSNETS.2019.8711418.

[3] Jamil, F.; Hang, L.; Kim, K.; Kim, D. "A Novel Medical Blockchain Model for Drug Supply Chain Integrity Management in a Smart Hospital". Electronics 2019. 8. 505.

[4] Hackius, Niels & Petersen, Moritz. (2017). “Blockchain in Logistics and Supply Chain: Trick or Treat?”.10.15480/882.1444.

[5] Q. Zhu and M. Kouhizadeh. "Blockchain Technology, Supply Chain Information, and Strategic Product Deletion Management." in IEEE Engineering Management Review. vol. 47. no. 1. pp. 36-44. First quarter. March 2019. DOI: 10.1109/EMR.2019.2898178.

[6] Prachi Shrikant, Pokharkar Pooja Machhindra, Pokharkar Monika Baban."Traceability and detection of the counterfeit medicine supply chain through Blockchain." Department of Computer Engineering Jaihind College of Engineering. Kuran. Volume 21 Issue 8. November 2019.

[7] K. Toyoda, P. T. Mathiopoulos, I. Sasase, and T. Ohtsuki. "A Novel Blockchain-Based Product Ownership Management System (POMS) for Anti-Counterfeits in the Post Supply Chain." in IEEE Access. vol. 5. pp. 17465-17477. 2017.DOI: 10.1109/ACCESS.2017.2720760.

[8] Huang, Yan & Wu, Jing & Long, Chengnian. (2018). "Drugledger: A Practical Blockchain System for Drug Traceability and Regulation." 10.1109/Cybermatics_2018.2018.00206.

[9] J. Ma, S. Lin, X. Chen, H. Sun, Y. Chen and H. Wang. "A Blockchain-Based Application System for Product Anti-Counterfeiting." in IEEE Access. vol. 8. pp. 77642-77652, 2020. DOI: 10.1109/ACCESS.2020.2972026.

[10] Q. Ding, S. Gao, J. Zhu, and C. Yuan. "Permissioned Blockchain-Based Double-Layer Framework for Product Traceability System." in IEEE Access. vol. 8. pp. 6209-6225, 2020. DOI: 10.1109/ACCESS.2019.2962274.

[11] Leng, Kaijun, Y. Bi, Linbo Jing, Han-Chi Fu and I. Nieuwenhuyse. "Research on agricultural supply chain system with double chain architecture based on blockchain technology." Future Gener. Comput. Syst. 86 (2018): 641-649.

[12] Mao, Dianhui & Wang, Fan & Hao, Zhihao & Li, Haisheng. (2018). "Credit Evaluation System Based on Blockchain for Multiple Stakeholders in the Food Supply Chain." International Journal of Environmental Research and Public Health. 15. 1627. 10.3390/ijerph15081627.

[13] Wang, Shangping; Li, Dongyi; Zhang, Yaling; Chen, Juanjuan (2019). "Smart Contract-Based Product Traceability System in the Supply Chain Scenario." IEEE Access. 1–1. doi:10.1109/ACCESS.2019.2935873

[14] Truffle. Accessed: Aug. 28, 2020. [Online]. Available: https://www.trufflesuite.com/truffle

[15] web3.js - Ethereum JavaScript API. Accessed: Aug. 28, 2020. [Online]. Available: https://web3js.readthedocs.io/en/1.0/

[16] Ganache. Accessed: Aug. 28, 2020. [Online]. Available: https://www.trufflesuite.com/ganache

[17] Vivekanadam, B. "Analysis of Recent Trend and Applications in Block Chain Technology." Journal of ISMAC 2, no. 04 (2020): 200-206.

[18] Sivaganesan, D. "Smart Contract Based Industrial Data Preservation on Block Chain." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 2, no. 01 (2020): 39-47.