International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

Satchain: Secured Autonomous Transactions in Supply Chain using Block Chain

B. Ratna Kanth, K. Venkata Ramana

Abstract*: With the increasing of digitization and massive adoption of advanced technologies in the various industries such as automotive, food, electronic goods etc. not only transform the equipment manufacturer's operating mode, but also changing the business models. In particular textile industry, the raw materials are collected from different other industries, end products are manufactured, distributed and sold globally. Supply chain and logistics, in particular, are considered as fertile ground for a blockchain implementation due to the several parties involved in the logistic processes and the lack of trust that usually characterize the industry. In this paper, we addressed the automatic textile industry supply chain as case study, in which we present a non – destructive way of ensuring the traceability of different operational modes of supply chain. We have implemented automatic block chain-based framework, which helps track and trace every mode of operation in supply chain. The proposed framework is simulated in Ethereum platform and the result shows the proof – of – concept of proposed model that can be used for wide range of future smart applications.

Keywords: Smart contract, supply chain, Ethereum, block chain.

I. INTRODUCTION

In the earlier, most of manufacturing companies produced almost everything within the firm. The global competition, forced them to focus on their core competences like engineering and final assembly as original equipment manufacturers thus outsourcing almost the whole manufacturing operations. These higher levels of complexity are the result of dramatic changes in manufacturing and distribution, including globalization and outsourcing. As a result, independent firms manage different parts of global supply chains. Each firm in the supply chain sets strategic and operational goals to maximize its own profit by using local information such as cost structures, profit margins and forecasts. Even though advances in information technology enable firms to collect, process, and share information, firms may be reluctant to do so because of conflicting incentives. Aligning incentives improves firms' profits and sustains the use of information technology.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
B. RATNA KANTH*, Department of CSE, Vignan Institute of Technology and Science, Yadadri Bhuvanagiri Dist., Hyderabad, India. Email: bratnakanth@gmail.com
Dr. K. Venkata Ramana, Department of CS & SE, A.U. College of Engineering(A), Andhra University, Visakhapatnam, India. Email: kvramana.auce@gmail.com

To achieve this state, the manufacturers need to build competitive supply chain. Due to the huge amount of produced and exchanged data needed for the production activities, it is essential to identify the most useful ones and to focus only on the "strategic transaction" leading to potential improvements at the supply chain level. Blockchain technology is generating a big stir in logistics and supply chain management. This particular technology received initial attention for its association with Bitcoin [3] and its capability to create a trusted and transparent ledger of transaction information. Now, as supply chain managers begin to recognize the possibilities of this new technology, there is high potential for elevating transparency. The arrival of this technology is timely because consumers are demanding supply chain transparency.

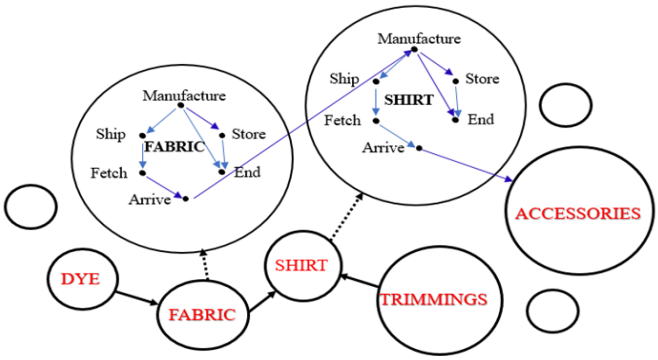


Figure 1: State Transition Graph

The main states of any traditional supply chain life cycle consist of Manufacturing, Storing, Shipping, Arriving and Termination. Consider the state diagram of textile industry, the state of manufacturing product entered by the supplier who currently hold the product as shown in Figure 1. A product could transition between two states only if there is link between those states in state machine. In textile industry the main responsibilities are shared among various manufacturing entities. Dyes, Fabric, Trimmings, Accessories are all manufactured differently and the final manufacturer will manufacture the end product. The validation of transition between two states by the smart contract as shown in the Section …..

In this paper, we design and implement prototype of supply chain application for textile industry. All the functions of supply chain are carried out by the smart contract, which is created and validate in the Ethereum platform. In the section 2, we describe the relation work of supply chain and block chain technology. The overall architecture of the proposed system is shown in the section 3. In Section 4,

we discuss algorithms of various functions in smart contract. Experimental results for creation and validation of smart contract in Ethereum platform are discussed in Section 5 and finally conclusion and future scope are discussed in the Section 6.

II. RELATED WORK

Blockchain technology, popularized by Bitcoin cryptocurrency, is characterized as an open-source, decentralized, distributed database for storing transaction information [3]. This makes transactions considerably more transparent than those provided by centralized systems. Supply chain system is also adopt this technology to improve supply chain transparency [2]. Every product transaction history stored for long period for future purposes. However, much of this history is presently obscured. Often, when negative practices are exposed, they quickly escalate to scandalous, and financially crippling proportions. Pun et al. examine how Block Chain Technology can be used to combat counterfeiting through a consideration of the interplay between a manufacturer and a counterfeiter. Lopez-Pintado et al. propose a blockchain-based collaborative supply chain process[5]. Mattila et al. claim that existing information integration between participants within a supply chain is ineffective and inefficient, and proposes Block Chain Technology as an enabler for effective product-centric information sharing [6]. Tijan et al. [7] noted that this emerging technology provides a basis for transactional data sharing across supply network without requiring trusted participations in food safety management.

Hofmann Et. al describes impact of block chain on both supply chain process and product management, and financial transactions between different network parties. A key potential blockchain supply chain advantage is the disintermediation of financial intermediaries, including payment networks, stock exchanges, and money transfer services [9]. This will make trading processes among partners more efficient. In the view of Ivanov. Et. al [10], the blockchain technology has the potential to improve efficiency through the use of smart contracts. Regulations, contracts, and policies, that can delay supply chain and logistic activities, can automatically be executed by smart contracts. This innovative application helps save time and reduce risk in supply chains.

III. ARCHITECTURE

The modules of the SATChain model are Material Management System (MMS), Quality Control Unit (QCU), Warehouse Management System (WMS), Transport Management System (TMS), Transaction Management System (T_xMS) and Logistic Management System (LMS). The participating entities of this system are Raw Material Supplier, Ancillaries, Manufacturers, Distributors and customers. This section describes the core modules of this architecture which are as follows:

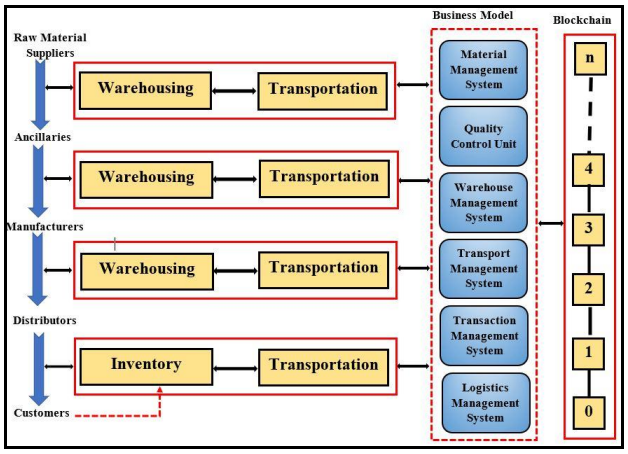


Figure 2: Architecture of the proposed model

Material Management System (MMS): The MMS's responsibility is to collect and fulfil the requirements of various participating entities.

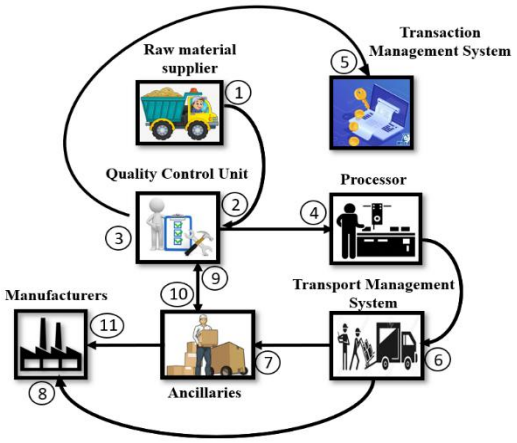


Figure 3 : Material Management System

The Process is as follows:

1. The RMS upon receiving the order from MMS collects all the required raw materials.
2. The raw materials are sent to QCU.
3. QCU verifies the quality of the raw material it received and prepares the required reports.
4. The report generated by QCU is further sent to Processor, where the processor classifies how the material has to be shared with other entities.
5. QCU updates the T_xMS about the quality of the raw material received and T_xMS prepares the amount to be paid for RMS for the material depending on the quality.
6. On further processing the order, Processor sends the raw material to TMS for dispatching them to the entities (Ancillaries and Manufacturers).
7. TMS dispatches the material to Ancillaries.
8. If the material was also requested by Manufacturer the TMS dispatches to Manufacturer also.
9. The Ancillaries process the raw materials, transform them to intermediate output and send them for QCU for quality report of the goods manufactured.

10. The QCU examines the goods and sends the Ancillary the report about their goods.
11. The Ancillaries further send the verified goods to the Manufacturer for completing the final product.

Quality Control Unit (QCU): The responsibility of this module is to check the quality of various raw materials, products manufactured by ancillaries and also the products manufactured by manufacturers. Depending on the quality of the product QCU issues various certificates to the authorities, upon which the payment for the products depend on.

Warehouse Management System (WMS): The responsibility of this module is to govern all the warehouses and inventories maintained by the various entities. This module together with TMS sees that goods are transmitted internally and maintained successfully in the warehouses. The process is as shown below.

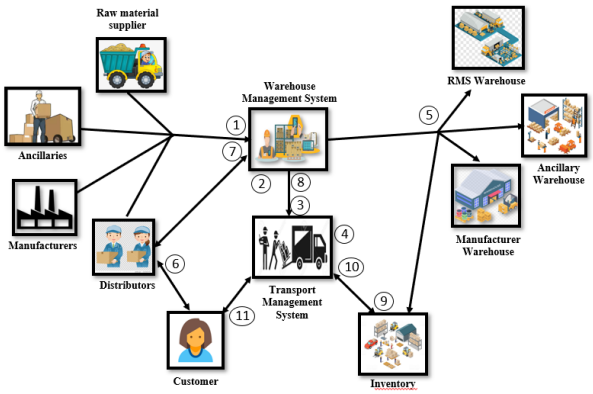


Figure 4: Warehouse Management System

1. WMS receives different products from various entities.
2. It processes the different goods received and prepares them for transport.
3. WMS sends these prepared products to TMS.
4. TMS prepares a plan and further chooses the optimized path for minimization of movement of goods.
5. The goods are transported to their warehouses or inventories.
6. As goods get updated in the inventories the customers place request to the distributor.
7. Distributor upon validating the request forwards it to the WMS.
8. WMS verifies product details with the information it has to choose the correct warehouse/inventory to retrieve the goods and informs the same to TMS.
9. TMS requests for the goods with the corresponding warehouse/inventory.
10. TMS processes the received goods for shipping them to the customer.
11. TMS upon receiving the payment from the customer delivers the goods to them.

Transport Management System (TMS): The responsibility of this module is to transfer goods from various entities to QCU and from QCU to the corresponding warehouse. Upon the request of the customer or upon purchase from other customers the TMS also transfers goods to their location.

Transaction Management System (T_xMS): This is the core module for all the payments. This module looks after all the transactions from collecting the raw materials to the payments done for RMS, Ancillaries, Manufacturers and Distributors. This module also collects payments from customers where required. All these transactions are updated to the blockchain. The process is as shown below.

1. The RMS sends raw material to QCU.
2. QCU sends back the quality report to RMS.
3. QCU updates this certificate to the T_xMS.
4. The T_x MS updates the blockchain about all the transactions updated by QCU.
5. RMS places request to LMS for the transfer of goods to Ancillaries or Manufacturers.
6. LMS conveys the accept/reject response to RMS.
7. TMS gathers raw materials and prepares for transfer to Ancillaries or Manufacturers.
8. LMS delivers the goods to Ancillaries.
9. LMS updates the T_xMS about cost incurred for the inherent transactions.
10. The T_x MS updates the blockchain about all the transactions updated by LMS.

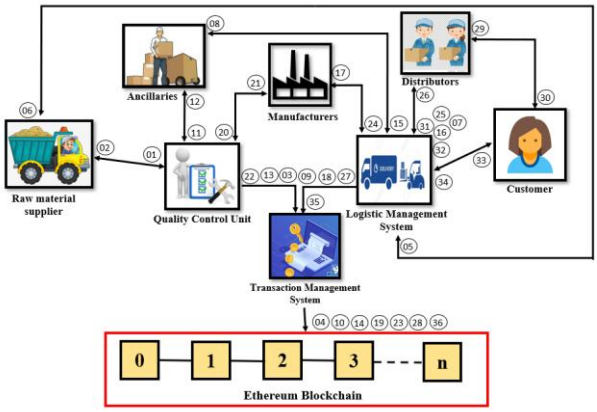


Figure 5: Transaction Management System

11. Ancillaries send their goods for quality check to the QCU.
12. QCU sends them the quality report.
13. QCU updates this certificate to the T_xMS.
14. The T_x MS updates the blockchain about all the transactions updated by QCU.
15. Ancillary places a request to LMS to transfer goods to Manufacturer.
16. LMS gathers the goods from ancillaries and prepares for transfer to Manufacturers.
17. LMS delivers goods to Manufacturers.
18. LMS updates the T_xMS about cost incurred for the inherent transactions.
19. The T_x MS updates the blockchain about all the transactions updated by LMS.
20. The manufacturer sends goods to QCU for quality check.
21. QCU sends back the quality report to Manufacturer.
22. QCU updates this certificate to the T_xMS.

23. The T_x MS updates the blockchain about all the transactions updated by QCU.
24. Manufacturer places request to LMS to transfer goods to Distributor.
25. LMS gathers the goods from ancillaries and prepares for transfer to Distributor.
26. LMS delivers goods to Distributors.
27. LMS updates the T_xMS about cost incurred for the inherent transactions.
28. The T_x MS updates the blockchain about all the transactions updated by LMS.
29. Checking with the information updated on the blockchain, customer places request for goods to the Distributor.
30. Distributor on verifying with the warehouse confirms/declines the customer request.
31. Distributor places request to LMS to transfer goods to Customers.
32. LMS gathers the goods from ancillaries and prepares for transfer to Customers.
33. LMS delivers goods to Customers.
34. Customer makes payment to the LMS.
35. LMS updates the T_xMS about cost incurred for the inherent transactions.
36. The T_x MS updates the blockchain about all the transactions updated by LMS.

Logistics Management System (LMS): LMS works with the integration of TMS, T_xMS and WMS. The LMS operates in a bidirectional way i.e. to the customers and from the customers. The to the customer way includes the operations from receiving an order, processing it, verifying it with WMS, fetching the item from the warehouse if present, Processing and packing it and handing it over to TMS for further shipping and delivering to the customer. The reverse way includes incorrect, damaged or recycling of products.

IV. SMART CONTRACT DESIGN AND WORKFLOW

The smart contract for this textile supply chain management system performs the following operations:

a) Adding Item: Here the smart contract verifies the address of the entity adding item to the repository. Only Seller is allowed to add items to the repository. Other than Seller no other entity or module is allowed to add items. This can be done as:

Algorithm 1: *AddItem algorithm*
Input: *Item_Details*
Output: *Addition of item to list*

```
1. Begin
2. C_addr ←fetch(callerAddress)
3. if(C_addr.isOwner())
            item[] =fetch(itemDetails)
   else
            display ("Only owner can
            perform this operation")
4. items[] = forSale(item);
5. prices[] =update(itemPrice)
6. state[]=mark item's state as 0
7. End
```

b) Purchasing Item: When an item is to be purchased the smart contract verifies the address of the entity intended to purchase the item. This condition checking is done so as to ensure that the entity purchasing the item is buyer and not the seller. Only buyers are allowed to purchase the item by paying price of the item. The money buyer pays can be greater than or equal to the cost of the item but not less. This can be implemented as:

Algorithm 2: *Purchase algorithm*
Input: *Amount payable by buyer,* A_{buyer}, *Item_Id,* I_{id}
Output: *Purchase of item from list*

```
1. Begin
2. B_addr ←fetch(BuyerAddress)
3. if(B_addr.isBuyer())
        if(A_buyer!=0 && A_buyer>=itemPrice)
            if(items.isAvailable())
                item.state = 1
            else
                display("Requested item is
                currently not available")
        else
            display("insufficient funds for
            purchase.")
    else
        display("only buyer can perform
        this operation")
4. items.buyer = B_addr
5. items.prices=A_buyer
6. items.quantity=items.quantity-1
7. items[]=update(item.buyer, item.state,
            item.price,item.quantity)
8. items.Rules_for_transfer_of_Fund()
9. End
```

c) Shipment of Item: In order to ship an item two conditions, have to be validated. First condition to be validated is to check whether the requested item is sold. This can be done by checking with the state of item. On satisfying the first condition the second condition's validation takes place. The condition it has to check is the entity's address that is trying to ship the item. Only seller is given permission to ship the items. This entire operation can be shown in algorithm 3

d) Delivery of Item: The delivery module gets executed after the shipment of item. Upon shipping the item, the state of the item changes and the same will be updated in the blockchain. Depending on the state of the item the functioning of the delivery module takes place. The two important conditions to be validated are 1. For an item to get delivered it has to be shipped prior to delivery. 2. Address of the entity that is trying to ship the item has be verified. Only buyer has the permission to ship the items.

This process is discussed in detail in algorithm 4

Algorithm 3: *Shipment algorithm*
Input: *Item_Id,* I_{id}
Output: *Shipment of item from list*

1. **Begin**
2. C_{addr} ←fetch(callerAddress)
3. if(C_{addr}.isOwner())
   goto step 4
   else
   display("Only owner can perform this operation")
4. if(, I_{id}.forSale())
   goto step 5
   else
   display("Only purchased items in the list can be shipped")
5. item.state =shipped(I_{id})
6. state[]= update(item.state)
7. **End**

Algorithm 4: *Delivery of item algorithm*
Input: *Item_Id,* I_{id}
Output: *Item received by buyer*

1. **Begin**
2. C_{addr} ←fetch(callerAddress)
3. if(C_{addr}.isBuyer())
   goto step 4
   else
   display("Only buyer can perform this operation")
4. if(, I_{id}.Shipped())
   goto step 5
   else
   display("Only shipped items can be delivered …Please try later!!")
5. item.state =received(I_{id})
6. state[]= update(item.state)
7. **End**

e) Fetching Item Status: This module can be executed by any entity participating in this framework. This module takes in the unique id of the item and retrieves all the details corresponding to the item such as name, id, price, state of item, Address of seller and address of buyer. The condition that has to be validated for retrieving the values is the existence of item in the repository.

Algorithm 5: *Fetch item algorithm*
Input: *Item_Id,* I_{id}
Output: *Item received by buyer*

1. **Begin**
2. if(I_{id}.exists())
   goto step 3
   else
   display("Enter a valid item id")
3. return fetch(I_{id}.name, I_{id}, I_{id}.price, I_{id}.state, I_{id}.seller, I_{id}.buyer)
4. **End**

f) Fund Checker: The fund checker module is mainly used to check the balance of the seller.

Algorithm 6: *Fund_checker algorithm*
Input: *Sender id,* S_{id}
Output: *Total fund available in the account*

1. **Begin**
2. if(S_{id}.isValid())
   goto step 3
   else
   display("Enter a valid sender address")
3. return fetch(balance(S_{id}))
4. **End**

g) Item Checker: This module fetches the list of total items represent in the repository. This can be done before an item is sold and even after item is sold.

Algorithm 7: *Item_checker algorithm*
Input: *items[]*
Output: *Count of items available with the seller*

1. **Begin**
2. if(items.state() ==0)
   goto step 3
   else
   display("No items are available with the seller")
3. return count(items)
4. **End**

h) Payment Rules: This module is responsible for the payments that has to be done to various entities. The conditions to be validated by this module are the percentage of sales. If the sales of items are above 60% the transfer of money takes place.

Algorithm 8: *Rules for transfer algorithm*
Input: *itemPrice[],itemCount*
Output: *Transfer of funds*

```
1. Begin
2. Total_price=fetch(total_itemPrice)
3. itemCount=fetch(items.count())
4. soldItems=fetch(items.state()!=0)
5. sales_percentage=soldItems/itemCount * 100
6. if(sales_percentage>=60)
       sender.balance=balance(sender)-Total_price
       buyer.balance= balance(buyer)+Total_price
   else
       display("Sufficient number of items not sold….")
7. End
```

The smart contract interaction diagram is shown below:

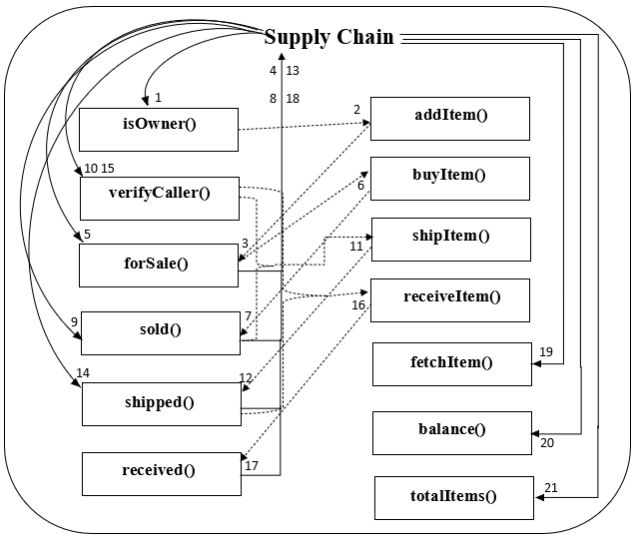


Figure 6: Smart Contract Interaction Diagram

V. EXPERIMENTATION

To deploy our model eight test machines were built for testing which work as light weight and a few also work as virtual nodes. The whole system is developed using the following environment: node of version 12.7.0, npm of version 6.11.2, go-ethereum as the platform for blockchain. The Solidity Framework and IDE for deploying smart contracts – Truffle is of version v5.0.30, the scripting language used to write the contracts – solidity is of version v12.7.0 and Remix IDE. The handling of requests between Ethereum[1] nodes is maintained by web3js is of version v^1.2.0. The hardware configuration of i3 processor, 4 GB Ram, few of 500 GB and others 1TB hard disks.

Table I: 1 ether is equivalent to[12]:

Value	Units
1000000000000000000	Wei
1000000000000000	KWei
1000000000000	MWei
1000000000	GWei
1000000	Szabo
1000	Finney
1	Ether
0.001	KEther
0.000001	MEther
0.000000001	GEther
0.000000000001	Tether

The complexity of this project is calculated in terms of gas units. The metrics used for measuring the cost is Gwei. Table 1 shows the conversions between various price units. The price of 1 unit of gas as measured while executing the project is 5 Gwei. Also 1 Gwei is equivalent to 0.00000022 USD.
The networks where this framework is JavaScript Virtual Machine environment, using Injected Web3 Environment with Ropsten test network and using Web3 Provider with Ganache for testing efficiency of this framework. While executing in various environments SATChain framework produced similar results incurring very less cost. This shows the efficiency of the framework. Table II shows the results produced when the project was deployed in JVM environment. Table III shows the results that were produced in Injected Web3 Environment (Ropsten test network). Table IV shows the results that were produced in Web3 Provider using ganache in linux Operating System. Table V shows the results that were produced in Injected web3 and rinkeby network.

Table II: Results of SATChain model in JVM Environment.

S. No.	Operation being performed	Number of units of gas consumed		Cost Incurred		Total Cost (in GWei)	Total Cost (in USD)
		Transaction (in gas)	Execution (in gas)	Transaction (in GWei)	Execution (in GWei)		
1.	Supply chain Contract deployment	11,27,719	8,18,687	56,38,595	40,93,435	97,32,030	2.1170
2.	Adding item to chain	1,60,251	1,38,019	8,01,255	6,90,095	14,91,350	0.3244
3.	Purchasing item from list	1,54,602	1,33,202	7,73,010	6,66,010	14,39,020	0.3130
4.	Shipment of item	29,823	8,423	1,49,115	42,115	1,91,230	0.0416
5.	Receiving item	29,735	8,335	1,48,675	41,675	1,90,350	0.0414
6.	Checking balance in user account	22,451	1,179	1,12,255	5,895	1,18,150	0.0257
7.	Fetching of item status	30,561	9,161	1,52,805	45,805	1,98,610	0.0432
8.	Total item count calculation	22,296	1,024	1,11,480	5,120	1,16,600	0.0254

Table III: Results of SATChain model in Injected Web3 Environment (Ropsten test network).

S. No.	Operation being performed	Number of units of gas consumed	Transaction Fee	Gas Price	Total Cost (in ether)	Total Cost (in USD)	Total Cost (in INR)
1.	Supply chain Contract deployment	9,32,927	0.000932927	0.000000001	0.000932928	0.23	16.69
2.	Adding item to chain	1,59,667	0.000159667	0.000000001	0.000159668	0.040	2.86
3.	Purchasing item from list	1,36,647	0.000115994	0.000000001	0.000115995	0.029	2.07
4.	Shipment of item	29,615	0.000029615	0.000000001	0.000029616	0.0074	0.53
5.	Receiving item	29,527	0.000029527	0.000000001	0.000029528	0.0074	0.53

Table IV: Results of SATChain model in Web3 Provider using ganache in linux Operating System.

S. No.	Operation being performed	Number of units of gas consumed	Gas Fee	Bytes of Information
1.	Supply chain Contract deployment	1131319	0.022626	4072
2.	Adding item to chain	168051	0.003361	132
3.	Purchasing item from list	98802	0.001976	36
4.	Shipment of item	28623	0.000572	36
5.	Receiving item	28535	0.000571	36

Table V: Results of SATChain model in Injected web3 and rinkeby network

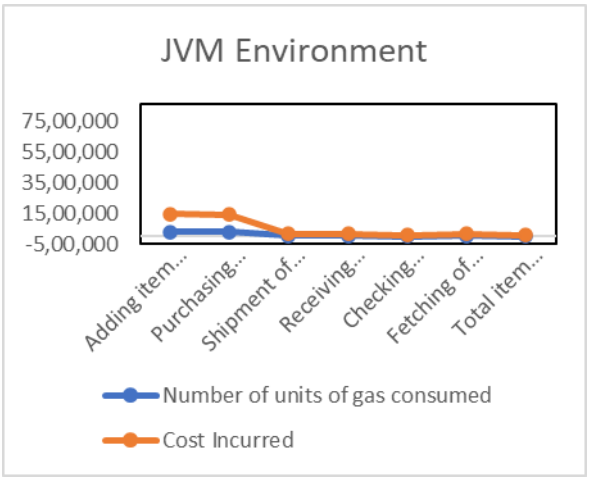
S. No.	Operation performed	Number of blocks mined	Number of transactions mined	Time taken to mine	Amount of information mined (in bytes)
1.	Supply chain Contract deployment	1	5	5s	5,431
2.	Adding item to chain	1	9	5s	2,390
3.	Purchasing item from list	2	14	10s	3,199
4.	Shipment of item	1	18	5s	2,153
5.	Receiving item	1	6	5s	1,548

VI. RESULT ANALYSIS

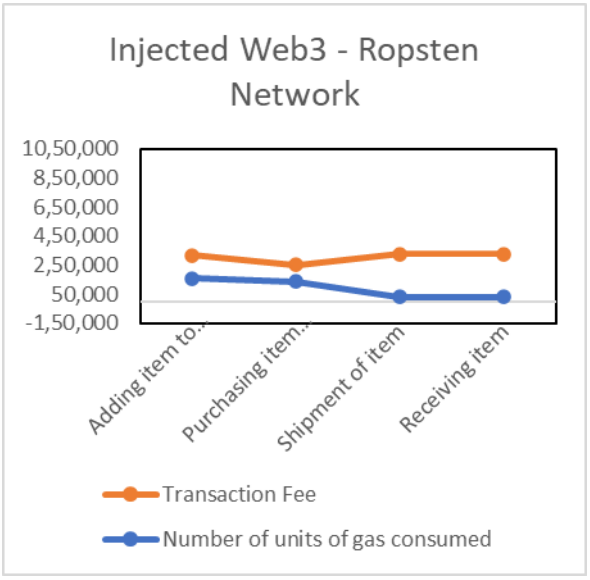
Our system is tested in various ways for measuring the efficiency. Apart from various other parameters measured few important parameters that were measured include Gas fee, Amount of information mined, Number of transactions mined, Time taken to mine a transaction.

Graph 1 shows the cost incurred for deploying our model which has 8 operational codes inclusive in it. To deploy this model 2.8374 USD were used which is equivalent to 0.024 Ether. Graphs 2 and 3 show the deployment of SATChain model in various environments. Graph 4 shows the relation between Bytes of Information mined and Gas Fee incurred as tested in Web3 Provider with Ganache Environment.

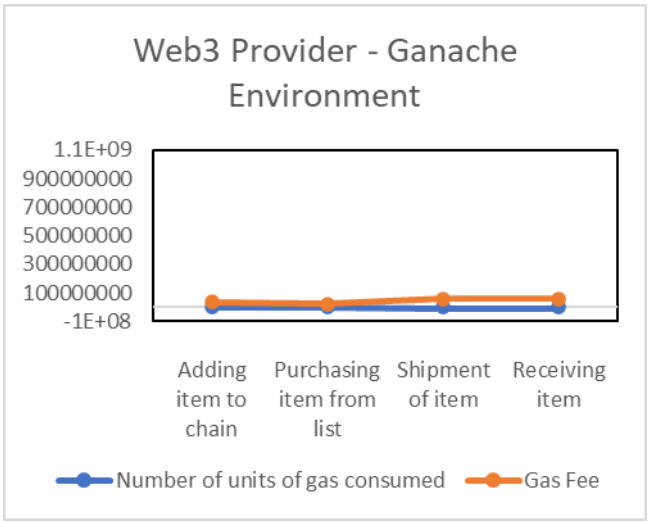
Our model is tested for efficiency using various parameters like number of transactions executed and amount of information mined. The results are shown in the Graph 5 and Graph 6.



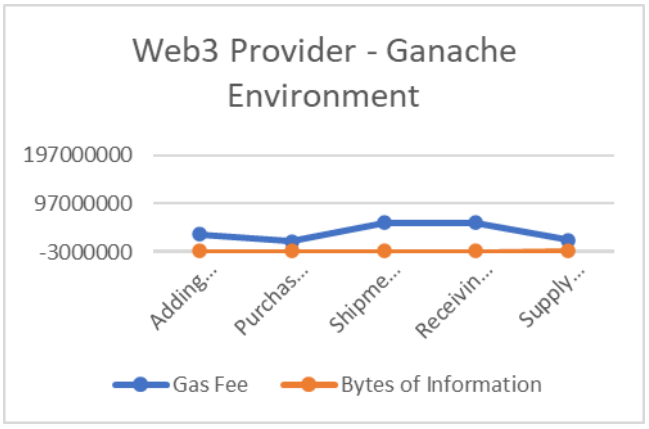
Graph1: Cost calculation graph as tested in JVM Environment.



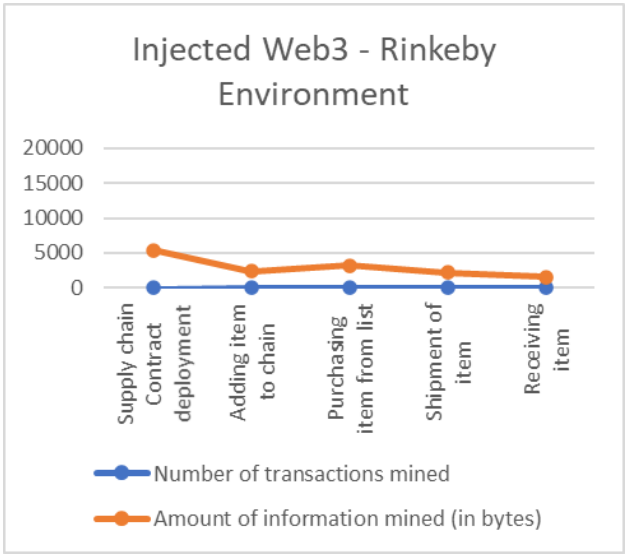
Graph2: Cost graph as tested in Injected Web3 Environment with Ropsten Network.



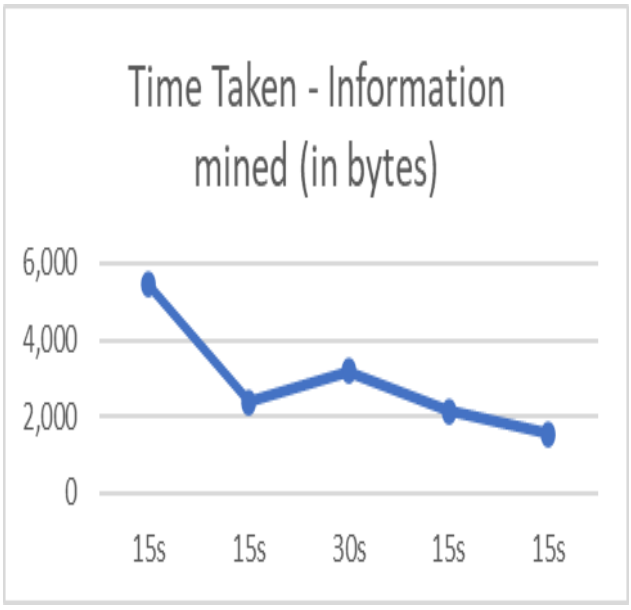
Graph3: Cost graph as tested in Web3 Provider and Ganache Environment.



Graph4: Bytes of Information Vs Gas Fee graph as tested in Web3 Provider with Ganache Environment.



Graph5: Amount of Information mined Vs Number of transactions mined graph as tested in Injected Web3 with Rinkeby Environment.



Graph6: Amount of Information mined Vs Time taken to mine graph.

VII. CONCLUSION

We designed a blockchain based SATChain model for textile supply chain. The operations are all build using an Ethereum smart contract. The transactions to be executed in Ethereum require gas to perform the operations. The cost calculation is done with the help of cryptocurrency called ether. The number of units of gas is responsible for the total cost of the project. Here we implemented various operations for adding, buying, fetching, retrieving, shipping and many other operations for textile supply chain. Though care has been taken for deploying smart contracts and providing security by placing them in a blockchain there are lot many attacks that place on the payment module and in many other places. The prevention of these attacks is the future scope for this paper.

REFERENCES

1. G. Wood: Ethereum: A secure decentralised generalised transaction ledger. In: Ethereum Project Yellow Paper, 2014, vol. 151
2. K Francisco and David Swanson, "The Supply Chain Has No Clothes: Technology Adoption of Blockchain for Supply Chain Transparency" (2018), Journal of Logistics. 2.2.10.3390/logistics2010002.
3. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: http://www.Bitcoin.org/bitcoin.pdf (accessed on 2 September 2017).
4. D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu, "Blockchain application in food supply information security," in the IEEE International Conference on Industrial Engineering and Engineering Management, (2018), 2017-December, pp. 1357– 1361.
5. Lopez-Pintado, O., L. Garc ´ ´ıa-Banuelos, M. Dumas, I. Weber. 2017. Caterpillar: A blockchain-based business ˜process management system. In Proceedings of the BPM Demo Track, 15th International Conference on Business Process Modeling (BPM 2017), Barcelona, Spain .
6. Mattila, J., T. Seppal ¨ a, J. Holmstr ¨ om. 2016. Product-centric information management: A case study of a shared platform with blockchain technology. escholarship.org.
7. Tijan, E. Aksentijevi´c, S. Ivani´c, K. Jardas, M. Blockchain Technology Implementation in Logistics. International journal of Sustainability, Volume 11, Pages 1185, 2019.
8. [126] Hofmann, E., U. M. Strewe, and N. Bosia. 2017. Supply Chain Finance and Blockchain Technology: The Case of Reverse Securitisation. Cham: Springer.
9. Tapscott, D., and A. Tapscott. 2017. "How Blockchain Will Change Organizations." MIT Sloan Management Review 58 (2): 10
10. Ivanov, D., B. Sokolov, and E. A. D. Raguinia. 2014. "Integrated Dynamic Scheduling of Material Flows and Distributed Information Services in Collaborative Cyber-Physical Supply Networks." International Journal of Systems Science: Operations & Logistics 1(1): 18–26.
11. Iansiti, Marco; Lakhani, Karim R.: The Truth About Blockchain. Harvard Business Review. Harvard University, 1–11 (2017)
12. https://www.etherchain.org/tools/unitConverter

AUTHORS PROFILE

B. Ratna Kanth, Associate Professor, Vignan Institute of Technology and Science, his current interests are Security, Machine Learning and Blockchain technology.

Dr. K. Venkata Ramana, Assistant Professor, Andhra University his current interests are Security, Machine Learning, Artificial Intelligence and Blockchain technology.