SECURE PAYMENT SYSTEM IN SUPPLYCHAIN MANAGEMENT USING BLOCKCHAIN TECHNOLOGIES

[1]SAI TEJASWINI GURUGUBILLI, [2] B RATNAKANTH, [3] Dr. K. VENKATA RAMANA

Department of Computer Science &System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, India.

Abstract

In enterprise resource planning a smart contract transaction system has become key area for secure operations in supply chain management a block chain transactions for different parties with specific condition is the main object which needed to be taken into consideration so we need to create an integrated system which can handle the secure transaction between the supplier and retailer. In this paper we have built a prototype to implement a secure payment transaction using block chain transactions and perform the analysis of block chain importance to evaluate the secure payment system here we use the sha256 to generate the hash truncations to create a relations between supply chain parties so that a trust can be built between the parties using smart contract system

Keywords: ***Blockchain, ERP, Supply Chain.***

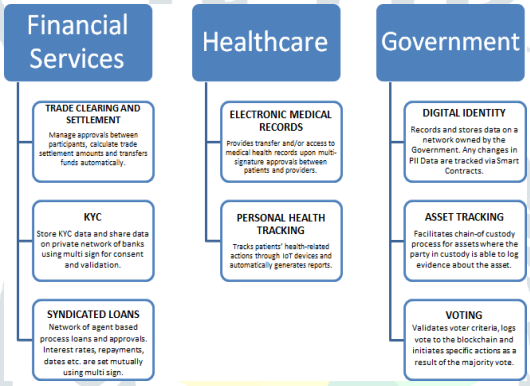
I. Introduction

Block chain is most emerged word in the recent technologies in the recent years earlier people used to work on peer-to-peer ledger but due to much security requirement these technology has been started in different area of platform these was one of major invention by Satoshi Nakamoto. However these keyword as attracted to many sectors such as entrepreneurs, governments, banks, health sector for compunction between different parties and business sector. The block chain was first time implemented in the year 2008 where it was implemented using crypto currency and Bitcoin in health sector [1] where the parties used to perform the transaction on private and public networks with using any online payments system [3]. The blockchain is basically a decentralized open ledger which verifies and store all the secure transaction [4]. The implementation is done using the hash code technique where the data is encrypted and the transaction are created in the from chain blocks[5].These type of technique has made easier for parties since there is no third party involvement like companies and banks so here the main concept is between the importance of the trust between the parties.

Enterprise Resource Planning (ERP) integrates all the parts of business into one system. In any ERP system the centralized server gathers all the information and stores in these server from different organizations and departments so in these types of system the data may be lost on single point of failure A third party is needed as a mediator between the businesses parties in every transaction for trust purpose. so there is need of implementation of secure transaction which can increase the role of the organization in security maintenance and it is benefited to all departments for secure communication of the data and building the channels with secure transaction processing which can hence maintain the quality of data analysis inside the organization [10].with these types of security features the user can have the transparency over the data so that it can make the user to understand and analyze which data is being accessed[11]. In addition to this, blockchain is also considered a new information technology where new data can be added to the ledger without changing historical data transactions. Once a transaction is placed by one of the parties on the chain, it will never be updated or eradicated. This has become the much need in implementation of block chain technology in supply chain management system so that they can get benefited from the trust since the transparency is maintained by the parties so we can implemented a smart payment system using the smart contracts this give much benefit to the parties. A smart contract is the process of implementing a computer programming code which can compute the perforce of an agreement with the blockchain automated system. A smart contract is called as set of rules which is record in the computer ledger these smart contract does not allow any third party involvement. The parties uses the distributed database which can verify the digital process which is written in the form of a computer program which defines the strict rules

Fig 1 : Smart contract System

Fig 2: Blockchain system Use cases



II. Related work

Both in research and in practice the interest in blockchain is rapidly increasing (Lemieux, 2016; Tschorsch and Scheuermann, 2016; Yli-Huumo et al., 2016). A large share of the publications read for this study, even those discussing major flaws of the existing blockchain protocol, state that this technology has the potential to revolutionize business (e.g. Baur et al., 2015; Gupta, 2017; Karame, 2016) and to be disruptive to current practices (e.g. Friedlmaier et al., 2016; Mettler, 2016; Yuan and Wang, 2016). Several authors expect blockchain to change the way transactions are made as much as the internet did in the past (e.g. Brennan and Lunn, 2016; Ito et al., 2017; Reyes, 2016).

Within the three years prior to summer 2016, over 2,500 patents were applied for, and over $1.4 billion was invested into projects within the blockchain field, with the majority of both central banks and commercial banks becoming engaged (McWaters and Bruno, 2016). Already a year ago, PricewaterhouseCoopers (2016: 17) identified 700 companies dealing with blockchain, categorising 150 of them as "worthy to be tracked" and 25 as potential industry leaders. A survey answered by over 800 executives at the 2015 World Economic Forum resulted in more than half the participants expecting 10% of the worldwide GDP to be stored on blockchain before the year 2025. Three quarters of the responders expected taxes to be collected on-chain before 2025 (Global Agenda Council on the Future of Software and Society, 2015). In the academic community, the interest in BCT and its business applications has been steadily growing over the last years [11] which is confirmed by a series of recent literature reviews (see [12, 13, 14, 15, 16]). An overarching research framework for BCT is provided by Risius and Spohrer [17]. In the slipstream of this development, SCM has only attracted minor interest among various application domains (see [18]).

While Korpela et al. [15] Focus on the issue of data integration in digital SCs using BCT, Sternberg and Baruffaldi [19] provide a first summary of potential applications of BCT in SCM from the literature. Hackius and Petersen [20] and Petersen et al. [13] use an expert survey of logistics professionals to explore potential applications and future prospects of BCT in SCM. Petersen et al. [13] synthesize application clusters of BCT in logistics and SCM from publicly available case examples. To the best of our knowledge, there is no comprehensive perspective on use cases of BCT in SCM.

The motivation of this paper arises from the gap highlighted by Banerjee [11], [12]. Banerjee [12] mentioned that research gaps are found regarding supply chain use cases based on the integration of blockchain and ERP, in addition to the research interest in developing a middleware for connecting blockchain and ERP. Banerjee [12] assured that interoperability still remains a gap in adopting blockchain for supply chains. According to Parikh [14], blockchain provides crucial capabilities to supply chains that ERP solely doesn't provide.

The integration of blockchain and supply chain was addressed by Korpela et al. [15], towards the implementation of a digital supply chain. This was applied to benefit from the advantage of security and flexibility at a lower cost of blockchain transaction compared to traditional transactions. Another challenge that is solved by the blockchain integration with supply chain is the transparency of activities, risks and financial flows. If this was applied on a Supply Chain Management system, it will enhance the whole business process and functionality of the system to achieve business objectives faster and more securely.

III. Problem Statement

Implementation of Block chain in Supply chain management plays a very key role between the parties for being involved in untrusted transaction which needed to be addressed. Since the performing sole ERP communication Between the parties does not provide the same level of trust so there need to be a smart contract so that high level trust can be implemented and also we need to see that the transaction between the parties does not affect record to be maintained of distributor and retailer over changes of data unless there is agreement between the parties in a given part of the time and we need to assure that there is no risk on the data which is stored in decentralized server. In this paper we build a block chain smart contract to establish the trust between the parties in supply chain management and end user can have ease of transaction which can allow the product visibility and user transparency using sha256 and Binary Merkel tree Algorithm

IV Methodology

RESEARCH DESIGN

MY implementation of the tool was completely explored since the complete data was collected from the internet resources and different journals and then it was been analyzed in detail and entire design was implemented so that it can cover the maximum information related to block chain transaction in supply chain management we have also undergone the complete structure of smart contract in ERP so that the end user can get the clear cut idea about the implementation and its working model browser version, the OS platform and whether some specific ports are open or closed. This data was then transmitted to the authors over email.

V Existing Approaches

Block chain was found to still face some serious immaturities. Nevertheless, it was acknowledged a promising technology for adoption in fields other than the original industry of application, finance. Some conceptual aspects, particularly regarding the kind of underlying shared ledger, need adjustments, some technological aspects need further development, and the framing conditions in economy and society need to be created. : For a long time, the focus was on the economic aspect, aiming to purchase a product or service at the lowest possible costs, until the environmental and later the social aspect became increasingly important. There are other problems harming companies directly, also caused by the opacity of modern supply chains: As products and their identities cannot be monitored and guaranteed through all the layers in many supply chains, fraud and counterfeit are facilitated and lead to massive direct financial losses for companies.

VI Proposed Approach

One of the crucial enhancements in supply chain processes is applying autonomous actions especially when these actions are taken between different parties who don't have mutual trust. This is achieved by block chain through smart contracts. In a supply chain, the distributor is responsible for dealing with different retailers of different point of sales. Each retailer has distinct requirements, as shown in figure 3. For the distributor to deal with these requirements instantaneously, integrating block chain with the used system will introduce a more efficient solution. We have used the sha256 hash code and binary Markel tree generation and the block chain is been appended using previous transaction and current transactions the proposed solution is applied on the following case scenario. The buyer sells an amount of commodities to the consumer but the buyer will get the complete money of the goods when his percentage of sale reaches 100%. The parties have a smart contract system will activate automatically whenever the point of sale reaches 100% and the total money gets transferred to the buyer account automatically. When the no of volumes of product is not fulfilled then appropriate actions will be implemented based on the rules agreed by both parties using the blockchain. In this paper we have integrated the ERP system with block chain transactions which is an open source ledger Framework for business [18]

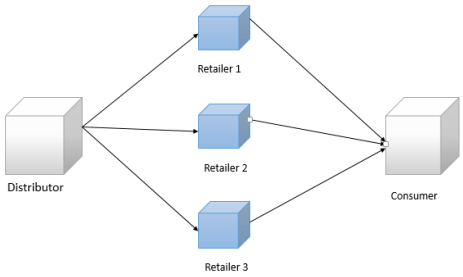


Fig 3: Blockchain network between the distributor and retailers.

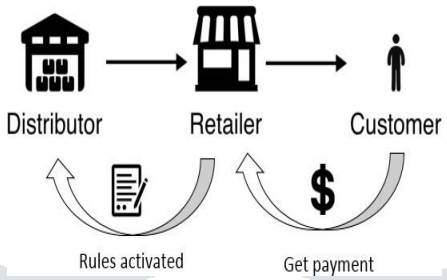


Fig 4: Flow of transactions.

The above Fig 4 represents about the rules agreed upon between the supply chain parties and this is called as smart contract rule one the all the commodities are received by the retailer the 100% payment transaction is made to the distributor by payment automation system

Algorithms:-

Blockchain incorporate *hashing* (transforming data of any size into short, fixed-length values). Hashing also incorporates another old technology called Merkle trees, which take many hashes and squeeze them down to one hash, while still being able to prove each piece of data that was individually hashed. Merkle tree also known as hash tree, is a data structure used for data verification and synchronization. It is a tree data structure where each non-leaf node is a hash of its child nodes. All the leaf nodes are at the same depth and are as far left as possible. It maintains data integrity and uses hash functions for this purpose.

Hash Functions:

So before understanding how Merkle trees work, we need to understand how hash functions work. A hash function maps an input to a fixed output and this output is called hash. The output is unique for every input and this enables fingerprinting of data. So, huge amounts of data can be easily identified through their hash.

This is a **binary Merkle tree**; the top hash is a hash of the entire tree.

- This structure of the tree allows efficient mapping of huge data and small changes made to the data can be easily identified.
- If we want to know where data change has occurred then we can check if data is consistent with root hash and we will not have to traverse the whole structure but only a small part of the structure.
- The root hash is used as the fingerprint for the entire data.

Table 1: time complexity for a binary Merkle tree

OPERATION	COMPLEXITY
Space	O(n)
Searching	O(logn)
Traversal	O(n)
Insertion	O(logn)
Deletion	O(logn)
Synchronization	O(logn)

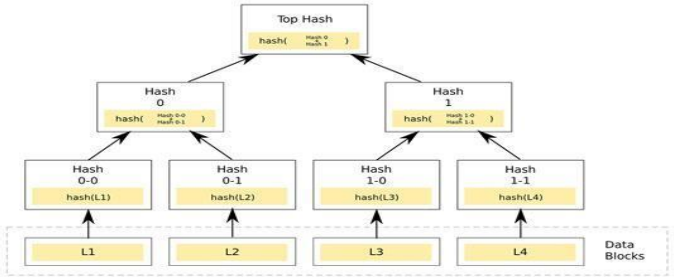


Fig 5: Markel tree

Sha-256 Pseudo code

Note 1: All variables are 32 bit unsigned integers and addition is calculated modulo 232
Note 2: For each round, there is one round constant k[i] and one entry in the message schedule array w[i], 0 ≤ i ≤ 63
Note 3: The compression function uses 8 working variables, a through h
Note 4: Big-endian convention is used when expressing the constants in this pseudocode,
and when parsing message block data from bytes to words, for example,
the first word of the input message "abc" after padding is 0x61626380
Initialize hash values:
(first 32 bits of the fractional parts of the square roots of the first 8 primes 2..19):
h0:= 0x6a09e667
h1:= 0xbb67ae85
h2:= 0x3c6ef372
h3:= 0xa54ff53a
h4:= 0x510e527f
h5:= 0x9b05688c
h6:= 0x1f83d9ab
h7:= 0x5be0cd19
Initialize array of round constants:
(First 32 bits of the fractional parts of the cube roots of the first 64 primes 2..311):
k[0..63] :=
0x428a2f98, 0x71374491, 0xb5c0fbcf, 0xe9b5dba5, 0x3956c25b, 0x59f111f1, 0x923f82a4, 0xab1c5ed5,
0xd807aa98, 0x12835b01, 0x243185be, 0x550c7dc3, 0x72be5d74, 0x80deb1fe, 0x9bdc06a7, 0xc19bf174,
0xe49b69c1, 0xefbe4786, 0x0fc19dc6, 0x240ca1cc, 0x2de92c6f, 0x4a7484aa, 0x5cb0a9dc, 0x76f988da,
0x983e5152, 0xa831c66d, 0xb00327c8, 0xbf597fc7, 0xc6e00bf3, 0xd5a79147, 0x06ca6351, 0x14292967,
0x27b70a85, 0x2e1b2138, 0x4d2c6dfc, 0x53380d13, 0x650a7354, 0x766a0abb, 0x81c2c92e, 0x92722c85,
0xa2bfe8a1, 0xa81a664b, 0xc24b8b70, 0xc76c51a3, 0xd192e819, 0xd6990624, 0xf40e3585, 0x106aa070,
0x19a4c116, 0x1e376c08, 0x2748774c, 0x34b0bcb5, 0x391c0cb3, 0x4ed8aa4a, 0x5b9cca4f, 0x682e6ff3,
0x748f82ee, 0x78a5636f, 0x84c87814, 0x8cc70208, 0x90befffa, 0xa4506ceb, 0xbef9a3f7, 0xc67178f2
Pre-processing:
append the bit '1' to the message
append k bits '0', where k is the minimum number >= 0 such that the resulting message
length (modulo 512 in bits) is 448.
append length of message (without the '1' bit or padding), in bits, as 64-bit big-endian integer
(this will make the entire post-processed length a multiple of 512 bits)

Process the message in successive 512-bit chunks:
break message into 512-bit chunks
for each chunk
create a 64-entry message schedule array w[0..63] of 32-bit words
(The initial values in w[0..63] don't matter, so many implementations zero them here)

copy chunk into first 16 words w[0..15] of the message schedule array
Extend the first 16 words into the remaining 48 words w[16..63] of the message schedule array:
for i from 16 to 63
s0:= (w[i-15] rightrotate 7) xor (w[i-15] rightrotate 18) xor (w[i-15] rightshift 3)
s1:= (w[i-2] rightrotate 17) xor (w[i-2] rightrotate 19) xor (w[i-2] rightshift 10)
w[i] := w[i-16] + s0 + w[i-7] + s1
Initialize working variables to current hash value:
a := h0
b := h1
c := h2
d := h3
e := h4
f := h5
g := h6
h := h7
Compression function main loop:
for i from 0 to 63
S1 := (e rightrotate 6) xor (e rightrotate 11) xor (e rightrotate 25)
ch := (e and f) xor ((not e) and g)
temp1 := h + S1 + ch + k[i] + w[i]
S0 := (a rightrotate 2) xor (a rightrotate 13) xor (a rightrotate 22)
maj := (a and b) xor (a and c) xor (b and c)
temp2 := S0 + maj
h := g
g := f
f := e
e := d + temp1
d := c
c := b
b := a
a := temp1 + temp2
Add the compressed chunk to the current hash value:
h0 := h0 + a
h1 := h1 + b
h2 := h2 + c
h3 := h3 + d
h4 := h4 + e
h5 := h5 + f
h6 := h6 + g
h7 := h7 + h
Produce the final hash value (big-endian):
digest := hash := h0 append h1 append h2 append h3 append h4 append h5 append h6 append h7
The computation of the ch and maj values can be optimized the same way as described for SHA-1.
SHA-224 is identical to SHA-256, except that:

- **the initial hash values h0 through h7 are different, and**
- **the output is constructed by omitting h7.**

SHA-224 initial hash values (in big endian):
(The second 32 bits of the fractional parts of the square roots of the 9th through 16th primes 23..53)
h[0..7] :=
0xc1059ed8, 0x367cd507, 0x3070dd17, 0xf70e5939, 0xffc00b31, 0x68581511, 0x64f98fa7, 0xbefa4fa4
SHA-512 is identical in structure to SHA-256, but:

- **the message is broken into 1024-bit chunks,**
- **the initial hash values and round constants are extended to 64 bits,**
- **there are 80 rounds instead of 64,**
- **the round constants are based on the first 80 primes 2..409,**
- **the word size used for calculations is 64 bits long,**
- **the appended length of the message (before pre-processing), in bits, is a 128-bit big-endian integer, and**
- **the shift and rotate amounts used are different.**

SHA-512 initial hash values (in big-endian):

h[0..7] := 0x6a09e667f3bcc908, 0xbb67ae8584caa73b, 0x3c6ef372fe94f82b, 0xa54ff53a5f1d36f1, 0x510e527fade682d1, 0x9b05688c2b3e6c1f, 0x1f83d9abfb41bd6b, 0x5be0cd19137e2179

VII Implementation

Step 1: each asset which is distributed between the parties contain the unique id, name and price of product, smart contract rule which is agreed between both the parties

Step 2: A unique Id is generated for each party which contains the information like metadata, name, product details and balance value. In this the three parties are been taken in consideration like retailer and distributor where everyone has their own role where they can track their transaction and point of sale (POS)

Step3: so that to track the product a unique id assigned is generated for each all details about the product are shown like, ID, name, buyer, seller, state of product and more. Here the system is atomized where each parties can know about their payment transaction and their smart contract system all the transaction are appended in the chain which will have the information about the sold products and their payment information and smart contract rules based on the smart contract the money transfer is performed

VIII Experiential Results

Deploying contract:

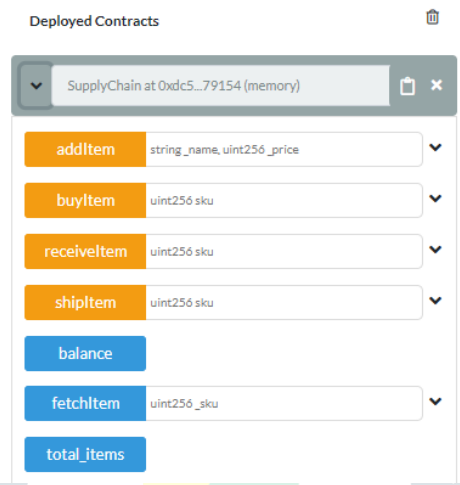


Fig 6: Contract Deployment Screen

Fig 6 depicts the deployment of the smart contract it shows the different functionalities like adding an item, buying an item, fetching item, shipping and receiving and etc… it in the smart contract which helps to negotiates between buyer and seller.

Fig 7 shows, whenever the contract deploys for each every transaction made by buyer and seller it generates the transaction hashcode in remix by default. It also show how much gas consumed to execute the transaction and the status of the transaction.

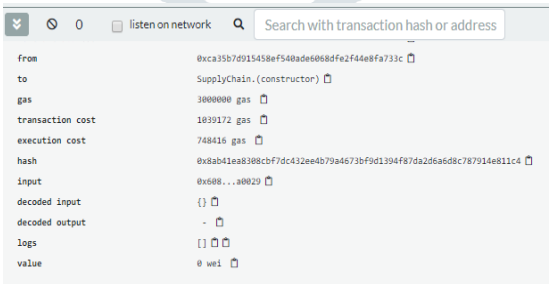


Fig 7: Transaction with Hash of deployed smart contact

Adding Item:

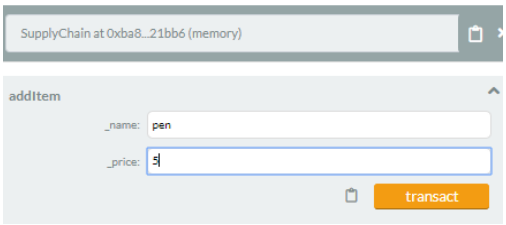


Fig 8: Adding Item

Fig 8 shows us how the items are added in supply chain. Only owner can add the items, whenever item gets added the state of item is shown as "For Sale", buyer address will be the address of owner who added the item and unique id gets generated for each and every item added.

Buy Item:

Fig 9: Buy item Transaction

Fig 9 shows us the transaction hash of the buying item functionality which is performed by the buyer. Customer should have sufficient balance in his account to buy an item and he should allocate some value while buying a product. When a customer buys an item the state of item changes to "Sold" from "For Sale".

Shipping Item:

Fig 9: Shipping Item Transaction

Fig 9 show us the transaction has of shipping functionality when the item is in sold state the owner will start the shipping process and the concerned customer will receive the item and state of item again changes to received.

Fetching Item:

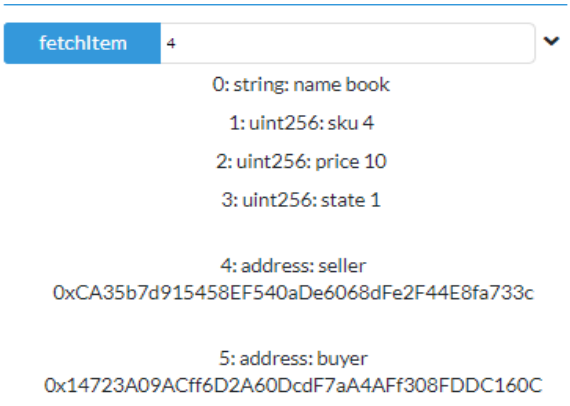


Fig 10: Fetching the item

Fig 10 depicts us the fetch functionality which traces the state of the item and shows the details of item like name, unique id, price of item, seller address and buyer address of the item. Anyone can fetch the item details.

IX. Conclusion

In this paper we have developed a tool to implement an efficient solution in supply chain management in ERP system where we have integrated the block chain payment system with the smart contract which can act a trust between the parties and can give much advantage in secure transactions In this paper we have implemented a distributed transaction system using Open Ledger Block chain system in supply chain management system we can store the huge data in ERP System and transaction are stored securely by implementation of secure hashing algorithm.

X References

[1] M. Risius and K. Spohrer, "A Blockchain Research Framework: What We (don't) Know, Where We Go from Here, and How We Will Get There," Bus. Inf. Syst. Eng., vol. 59, no. 6, pp. 385–409, 2017.

[2] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in Advanced Communication Technology (ICACT), 19th International Conference, pp. 464-467, 2017.

[3] A. W. Reijers, F. O. Brolcháin, and P. Haynes, "Governance in Blockchain Technologies & Social Contract Theories: Open Review," Ledger J., vol. 5980, 2016.

[4] M. Swan, "Blockchain Thinking: The Brain as a DAC (Decentralized Autonomous Organization)," in Texas Bitcoin Conference, pp. 27-29, 2015.

[5] V. L. Lemieux and V. L. Lemieux, "Trusting records: is Blockchain technology the answer?" Rec. Manag. J., 2016.

[6] L. Carlozo, "What is Blockchain?" J. Account., vol. 224, no. 1, p. 29, 2017.

[7] N. Garcia-Sanchez and L. E. Perez-Bernal, "Determination of Critical Success Factors in Implementing an ERP System: A Field Study," Inf. Technol. Dev., vol. 13, no. 3, pp. 293–309, 2007.

[8] Y. B. Moon, "Enterprise Resource Planning (ERP): a review of the literature," Int. J. Manag. Enterp. Dev., vol. 4, no. 3, pp. 235–264, 2007.E. J. Umble, R. R. Haft, and M. M. Umble, "Enterprise resource planning: Implementation procedures and critical success factors," Eur. J. Oper. Res., vol. 146, pp. 241–257, 2003.

[9] V. Morabito, "Business Innovation Through Blockchain," Springer International Publishing, 2017.

[10] G. Zyskind and A. S. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," n Secur. Priv. Work., pp. 180–184, 2015.

[11] A. Banerjee, "Blockchain Technology: Supply Chain Insights from ERP," 1st ed. Elsevier Inc., 2018.

[12] A. Banerjee, "Integrating Blockchain with ERP for a Transparent Supply Chain," Infosys Limited, p. 8, 2017.

[13] Hackius, Petersen "Blockchain in logistics and supply chain : trick or treat?",2017.

[14] T. Parikh, "The ERP of the Future: Blockchain of Things," Int. J. Sci. Res. Sci. Eng. Technol., vol. 4, no. 1, pp. 1341–1348, 2018.

[15] K. Korpela and T. Dahlberg, "Digital Supply Chain Transformation toward Blockchain Integration," Proc. 50th Hawaii Int. Conf. Syst. Sci., pp. 4182–4191, 2017.

[16] J. H. Tseng, Y. C. Liao, B. Chong, and S. W. Liao, "Governance on the drug supply chain via gcoin blockchain," Int. J. Environ. Res. Public Health, vol. 15, no. 6, 2018.

[17] K. A. Clauson, E. A. Breeden, C. Davidson, T. K. Mackey, and T. K. Mackey, "Leveraging Blockchain Technology to Enhance Supply Chain Management in Healthcare," Blockchain Healthc. Today, vol. 1, pp. 1– 12, 2018.

[18] C. Cachin, "Architecture of the hyperledger blockchain fabric," IBM Research. Zurich, 2016.

[19] Henrik Sternberg , Giulia Baruffaldi" Chains in Chains – Logic and Challenges of Blockchains in Supply Chains"

[20] Hackius, Petersen "Mapping the sea of opportunities: Blockchain in supply chain and logistics",2018