Blockchain Based Monitoring on Trustless Supply Chain Processes

Kwame O.-B Obour Agyekum[1,2], Emmanuel Boateng Sifah[1], Ali Shahzad[1], Hanlin Yang[1], Smahi Abla[1], Jianbin Gao[3], and Qi Xia[1,2]

[1] University of Electronic Science and Technology of China, National Engineering Laboratory for Big Data Application on Improving Government Governance Capabilities, Chengdu, China
[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[3] School of Resource and Environment, Center for Digital Health, University of Electronic Science and Technology of China

Abstract. Current trends in global business and enterprise typically force companies that create, distribute and sell products to employ long and diverse supply chain networks. While delivering value these networks also have several challenges with data management due to individual companies use of disparate ledgers vulnerable to fraudulent human interventions. As a result errors in transmission, intentional or otherwise, of just one parameter can consume resources of manufactures, retailers, their respective banks and legal representatives and cause problems that require formidable arbitration to resolve, further diminishing trust in an industry that has little of it. In this paper we present a blockchain solution that facilitates secure, transparent transactions between parties in the supply chain. We point out the merits of our design, the smart contracts that assist its operations and the significant boost it gives to restoration of trust in business activities.

Keywords: Supply chain management · Sovereign blockchain · Off-chain processes · Business continuity.

1 Introduction

Traditional supply chain functions in relation to payment processing and order fulfillment suffer considerably from several challenges. Lack of transparency and trust in the supply chain are two major problems. Buyers and sellers have to have reliable systems for verifying and validating the different transactions. Currently supply chains potentially involve hundreds of stages, actors and different geographic locations. When an actor in the supply chain conducts illegal activity, investigations become a difficult task and it is often hard to identify the offender [1–4]. The payment for orders are usually handled by banks in the time frame agreed to by the transacting parties. This arrangement consequently leaves the customer access to only such information as when the ordered product will be delivered or when the proposed service will be rendered.

Modern supply chain methods continue to seek more cost-effective means of operation [5, 6] (which pertains to in-house supply chain process), greater transparency and efficiency in the enterprise value-chain [7, 8](by ensuring fairness in a trustless business environment enabling parties to trust visibility achieved by blockchain monitoring on business processes). While large, centralized systems have been created to manage the flow of goods and data generated as a result, a single challenge remains to be addressed[9, 10]. The data generated as a result of transactions can be changed from its original form[11], causing some parties to perceive the supply chain as not being fully transparent with suppliers', manufacturers' and logistics' processes[12, 13].

One solution to this problem would be to develop a system that helps the customer gain access to more information on the origin of a product. This includes specific supplier information, mode of payment and shipment which ensure significant transparency between customers and suppliers. In this work we propose a blockchain system to enhance the transparency of activities and processes carried out between customers and suppliers. With its emphasis on distributed[14], immutable[15], timestamped transaction logs, the technology can enable entities in the supply chain identify fraudulent activities more easily. This gives supply chain processes an advantage over entities that seek to undermine legitimate system activities[16–18].

2 Related Work

Works carried out in this direction include Saveen et. al's paper on "Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger"[19]. He reviewed some of the main characteristics of the Blockchain technology and discussed potential application domains. The proposed system facilitates a vast amount of data to be collected about the products and users in manufacturing industry, which can prove to be beneficial to a range of different people, organizations, governments and researchers. For example, this allows consumers to readily access accurate data specific to any product that has been manufactured through a blockchain enabled supply chain, thus allowing them to make better buying decisions.

Feng Tian also proposed an agri-food supply chain traceability system based on RFlD (Radio-Frequency Identification) and blockchain technology in his paper "An Agri-food Supply Chain Traceability System for China Based on RFID and Blockchain Technology"[20]. The system proposed covers the whole process of data gathering and information management of every link in the agri-food supply chain. This realizes the monitoring, tracing and traceability management for the quality and safety of the food "from farm to fork".

The final work reviewed was achieved by Ingo Weber and his team in "Untrusted Business Process Monitoring and Execution Using Blockchain" paper[21]. He proposed a technique to integrate blockchain into the choreography of processes in such a way that no central authority is needed, but trust is maintained.

Their solution comprises the combination of an intricate set of components, which allow monitoring or coordination of business processes.

The modern supply chain system continues to seek more cost savings and greater transparency and efficiency in all processes. While large, centralized systems have been created to manage the flow of goods and data, a single problem remains. The data can be altered in ways that beat current detection methods causing some parties to assert the supply chain process is not fully transparent with all relevant parties.

In this paper we propose a system that makes use of secure technologies to enhance the transparency of transactions taking place between customers and suppliers using blockchain and smart contracts technologies in the supply chain system. These technologies enable supply chain companies to identify attempted fraud more easily.

3 System Architecture

The system has three main components that interact with one another to achieve system goals. These are the Users, Smart Contract Network and the Blockchain Network. The Users consist of nodes of participating manufacturers, suppliers and others who request goods and/or services. These users each run nodes that participate in the network and take part in the creation, transmission, reception or forwarding of services/goods requests. The process of providing a service begins when any one party contacts the other with a specific service request. Typically this group comprises retailers, manufacturers, distributors and suppliers. We also include logistics services providers, consultancy firms, financial institutions, government and regulatory agencies, machine and equipment providers, and indirect materials suppliers. These, often in pairs, initiate and perpetrate actions in the system.

The blockchain system consists of a smart contracts center and a blockchain network. They work together to generate smart contracts after transacting parties have agreed on service provision requests. They also ensure quantities and timelines are transparent to all parties that may be required to observe the transaction. The smart contracts center checks received requests (contracts) for attributes necessary to permit the service request to proceed. It inspects the signatures on the requests for correctness and checks quantities and delivery conditions along with penalties for non-compliance. There may be other checks as transacting parties and system demands may deem appropriate. When complete the service request is timestamped and a hash of it is also generated and forwarded to the blockchain network. Once here, all active nodes receive a copy of the transaction. We allow for two transactions between any pair of users at a time: an "opening" transaction and a "closing" one. We allow for multiple transactions and messages between parties but to keep the network traffic light and performance optimal we require only opening and closing transactions. The intervening messages may pass freely back and forth between the parties through dedicated off-blockchain channels setup for the purpose. Since the system is ar-

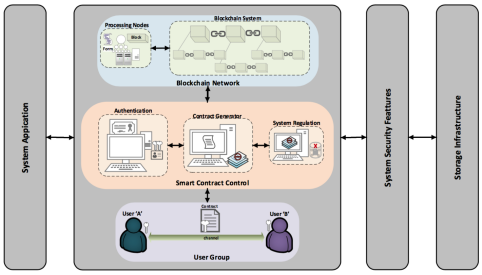

Fig. 1. Complete system architecture.

chitected on the blockchain and has consequences (legal sanctions and financial penalties) for all involved it should be the case that opening and closing transactions reconcile easily.

The system has three main entity types. These are the users (transacting parties), the smart contract network and the blockchain network. Each supply chain party runs nodes that after setup participate in the system protocol, checking submitted contracts and flagging infractions to them as required.

1. Users: Users consist of individuals and/or organizations that request services or goods so as to initiate a two-way flow of information, materials and payments. On the basis of the products or services in question, the scope of requirements and particular demand and supply attributes, the users or participants in the supply chain network may be geographically dispersed. For our purposes we consider retailers, distributors, manufacturers, and suppliers of raw materials.
2. Smart contract network: A Smart Contract, broadly speaking, is an autonomous program that encodes the terms of a binding agreement between transacting parties and is capable of facilitating, verifying and enforcing the terms of the agreement without the involvement of third parties. The document (service or supply of materials request) is "smart" because of the clever use of self-executing computer programs for the enforcement of agreements and its a "contract" in the true sense of the word in that it specifies the legal relationship of obligations between the transacting parties. For the system we propose we use smart contracts to enforce checking of contract conditions necessary for carrying out specified actions. For example the Smart Contracts Center checks that the document is signed by both transacting parties. This is trivially done as the center has the public keys of all active entities on the network. It would then check for conflicting conditions in the contract document. When this passes successfully it sends the hash and timestamp of the document to the blockchain network.
3. Blockchain network: The blockchain is a distributed public ledger technology that permits the recording, verification and validation of transactions in a timestamped tamper-proof manner. Its design permits all active nodes in the network access to all the data available in the network. Protocols for

validating and ordering transactions permit all nodes to arrive at one version of transaction history that is consistent with other active nodes across the network. For the supply chain system under construction the blockchain receives the hash of the signed, timestamped document for keeping. The transacting parties are free at anytime revisit the stored document (signed and timestamped) as proof of service or product request.

4 Session Generation

To establish a secured communication channel between two transacting parties in the business environment, there is the need to generate session keys between the transacting parties. The blockchain system aids in authenticating various transacting parties and eventually creates the channel.

For our illustration, we denote transacting parties A and B by Tx_A and Tx_B respectively. For a business transaction between the two transacting parties, party A chooses a random number Rn_{k_A} concatenates this with his identity ID_A in addition to transacting party B's identity ID_B and sends $Enc(ID_A||Rn_{k_A}||ID_B)$ to the authenticator. Transacting party B correspondingly chooses a random number Rn_{k_B} concatenates this with his identity ID_B in addition to transacting party A's identity ID_A and sends $Enc(ID_B||Rn_{k_B}||ID_A)$ to the authenticator. This is done to identify the various transacting parties and whom business relations for a particular transaction is being formed. The random number is generated to verify the authenticity of the authenticator.

The authenticator replies to both parties, A and B with a verified message $Auth(T_{k_A}||H(Rn_{k_S}||Rn_{k_A})||Sq_n)$ and $Auth(T_{k_B}||H(Rn_{k_S}||Rn_{k_B})||Sq_n)$ respectively where Sq_n is used to identify the sequence number of the message. T_{k_A} and T_{k_B} are token allocated to A and B which will be used to establish the session between the parties. The token is made up of $H(Rn_{k_A}||ID_B||S_k||Rn_{k_S}||T_s)$ and $H(Rn_{k_B}||ID_A||S_k||Rn_{k_S}||T_s)$ for A and B respectively. S_k denotes the session key to be used for the business process. Parties A and B compute and sign $\sigma(T_{k_A}||Src_A||Mac_A)$ and $\sigma(T_{k_B}||Src_B||Mac_B)$ respectively which is sent to the authenticating unit. Src_A and Src_B are used to directly connect the parties together for the business transaction processes. Mac_A and Mac_B denotes the message authentication code used to authenticate the messages sent from the transacting parties to the authenticator to specify the message came from the stated parties and has not been changed. The authenticator verifies Mac_A and Mac_B which denotes $Mac_A = (ID_A||Sq_n||Rn_{k_S}||Sn_k)$ and $Mac_B = (ID_B||Sq_n||Rn_{k_S}||Sn_k)$ and sends the Mac to respective communicating parties A and B which contains the session keys used to encrypt and decrypt transactional messages.At this point, a channel is established and parties will use the session key to hold communication on a business transaction. The session is closed when both parties agree to end a transaction and the result of the business transaction has been broadcast into the blockchain network.

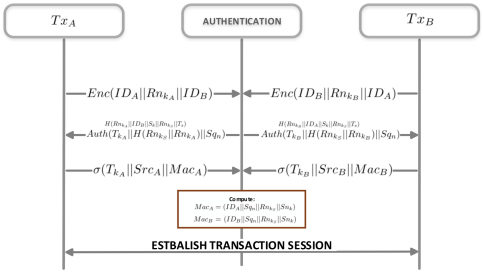

Fig. 2. Establishing session between two transacting parties.

5 Smart Contract Design

A Smart Contract is an autonomous software (a self-executing contractual state) that resides on the blockchain, controlled collectively and has its actions easily verifiable. In terms of a supply chain, the smart contract permits convenient carrying out of payments once contractual obligations have been met. They are particularly relevant to supply chains because of the sheer size a chain can assume in order to facilitate the provision of requested services. Ideally transactions are initiated and carried out between pairs of requesters and suppliers but because of trust issues each party in the transaction calls in their respective banks and legal representatives. These are trusted entities whose fees usually lead to bloated service charges. The smart contract permits the removal of these in a manner that reduces costs while maintaining the security of the transaction process for all the interested parties. Smart contracts can be written to include all conditions, legal clauses, penalties and sanctions of a contract and still be enforced in a manner that the transacting parties respect. The blockchain smart contract in this paper is initialized into two categories called the **agreement contract** and **processing monitoring contract**. The first sets out the terms of the requested services while the last, as implied in the name, systematically checks the status of the requested service till it is concluded.

6 System Performance Analysis

For our system we make use of opening and closing transactions which leave intervening messages to reside only on the information systems of directly transacting parties. We transform existing business processes as implemented in supply chain environments through a translator implemented as scripts through smart contracts in solidity.

For our blockchain-based monitoring system for supply chain processes, we design an implementation as a proof-of-concept on Ethereum in order to evaluate the system. We initiate smart contracts in our system by the use of solidity which is a state-based scripting language with no restrictions on data size stored in the blockchain network. Transacting participants in the system are accounted

Algorithm 1 Agreement Contract

Require: Contract parameters
Initialise: *Consumer.ID, ServiceProvider.ID, SpecifyBy, Specify.Status, Confirmby.Status*
Retrieve: *Contract.Formulation*
for $Specify.By == Consumer.ID$ and $Specify.Status == Active$ **do**
$Consumer.ID \rightarrow Contract.Term$
$Retrieve.PublicKey \rightarrow Customer.ID$
$Customer.ID \rightarrow Contract.Sign$
$Send.Contract \rightarrow ServiceProvider.ID$
$Timestamp \rightarrow BlockchainNode$
$Log \rightarrow BlockchainNode$
Terminate process
end for
for $ConfirmBy == ServiceProvider.ID$ and $Confirm.Status == Active$ **do**
$Confirm.ConsumerSig \leftarrow ServiceProvider.ID$
$Retrieve.PublicKey \rightarrow ServiceProvider.ID$
$ServiceProvider.ID \rightarrow Contract.Sign$
$Timestamp \rightarrow BlockchainNode$
$Log \rightarrow BlockchainNode$
Terminate process
end for
Initiate: Smart Contract Protocol

as external which define actions on each user based on smart contracts. Smart contracts are therefore activated by initiation of a business process and agreement between transacting parties. Enforcing these contracts are achieved using Truffles which is used to initialise contract processes involving contract management, deployment and contract migration with several testing frameworks enabled through truffle boxes. These serve as the substructure fit setting up environments for testing owing to pre-configured web-development environments. Mining in our work is achieved by processing nodes through pre-processing transactions and validating business process to ensure correctness before blocks are created and broadcast into the blockchain network. This is backed by smart contract data.

Interfaces connecting transacting parties to the blockchain system use web3.js library enabled through RPC calls. In addition, javascript libraries connect all entities in the blockchain structure by use of COAP to allow interactions among the processing nodes, blockchain system and contract generator. Partcipants in the system interact with each other in a business process through javascript web-client libraries implemented. We in-turn generate private and public key pairs to uniquely identify participants registered to the system. We initialize the number of active requests from 50 transactions to 2500 transactions as active users to varied periods of time for between 10 minutes to allow for generation, processing and broadcasting of blocks.

Algorithm 2 Process Monitoring Contract (A)

Initialize: *Agreement.Contract, Session.ID, Timestamp.Start, Timestart.End, Transaction.Status*
commit.Customer: *Customer.ID → Network*
commit.Customer = Amt.Payable + Amt.Penalty
commit.ServiceProvider: *ServiceProvider.ID → Network*
commit.ServiceProvider = Amt.Penalty
if Transaction.Status == Active and End.Time == Elapsed **then**
Terminate procedure
Retrieve.Contract → Network
Allocate.Amt.Payable → Consumer.ID
for violation == Service.Default **do**
Allocate.Amt.Penalty: ServiceProvider.ID → Consumer.ID
Retrieve.PublicKey → ServiceProvider.ID
ServiceProvider.ID → Contract.Violation
end for
Retrieve.PublicKey → ServiceProvider.ID
ServiceProvider.ID → Contract.Violation
Retrieve.PublicKey → Customer.ID
Customer.ID → Contract.Sign
Timestamp → BlockchainNode
Log → BlockchainNode
else if Transaction.Status == InActive **then**
Retrieve.Contract → Network
Check: *Service.Contract*
for Service.Contract == Reached **do**
Allocate.Amt.Payable → ServiceProvider.ID
Retrieve.PublicKey → ServiceProvider.ID
Retrieve.PublicKey → Customer.ID
Customer.ID → Contract.Sign
end for
for Service.Contract == Target.Unachieved **do**
Allocate.Amt.Payable → Customer.ID
Retrieve.PublicKey → Customer.ID
Retrieve.PublicKey → ServiceProvider.ID
ServiceProvider.ID → Contract.Sign
end for
Timestamp → BlockchainNode
Log → BlockchainNode
else {Transaction.Status == Active and End.Time == Elapsed and Time.Elapsed == ignore}
Continue business process
end if

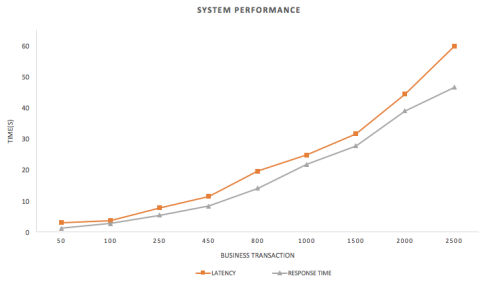

Fig. 3. System performance based on latency and response time.

In our system, benchmarking is achieved through jMeter which is an apache benchmarking tool. The main metrics to establish the efficiency of our system is latency, query time and response time. Latency is based on the delay before the processing of blocks are completed based on the initialization of a transaction completed by participants in the system. This includes all steps required to process a request by all participants in the supply chain system. Response time is evaluated on the length of time taken by the entire system to achieve concurrency in blocks generated per participant which directly affect the efficiency of the entire system. In considering response time, we account for query time which is based on the rate at which system requests are completed in initiating a business process in addition to generating smart contracts pertaining to business transactions. Given variations to the number of active transactional requests, we analyse the time it takes for each system to execute the total number of requests. The number of requests in the system is therefore represented as active transactions.

As represented in ***table 1*** and ***Figure 3***, a close observation shows a considerable increase in latency as business transactions per group of participants increase in the system. This however is controlled by optimization on the methods of processing transactions by the processing nodes. This enables us to achieve an acceptable and efficient response time for business processes. Query time proves to be controlled since all transactions are completed within seconds of being created. The increase in latency is due to the trade-off of achieving security on business processes.

7 Conclusion

The global nature of large supply chains results in difficult management and costly executions. Errors in requests do not just result in product delays and costly financial penalties. They can have a domino-effect on several other industries. It is therefore very important that erroneous information in supply chain systems be eliminated altogether or drastically minimized. The blockchain with its emphasis on immutability of transactions and chronological ordering of events

Table 1. Performance analysis for blockchain based monitoring on trustless supply chain processes.

Active Transactions	Latency(sec)	Response Time(sec)
50	3.018	1.25
100	3.71	2.73
250	7.89	5.42
450	11.46	8.35
800	19.65	14.09
1000	24.83	21.81
1500	31.61	27.73
2000	44.39	39.01
2500	59.83	46.54

presents a way to ensure entities in a supply chain can place requests for services whose fulfillment and payment terms can be supervised by smart contracts. By so doing we eliminate inaccurate information from participating nodes and ensure a more efficient service process. Since the system removes the necessity of third parties such as banks the service process also becomes significantly less expensive.

References

1. A. Brinkhoff, . zer, and G. Sargut, All you need is trust? An examination of inter-organizational supply chain projects, Prod. Oper. Manag., vol. 24, no. 2, pp. 181200, 2015.
2. A. Singh and J. T. C. Teng, Enhancing supply chain outcomes through Information Technology and Trust, Comput. Human Behav., vol. 54, pp. 290300, 2016.
3. J. Peng, J. Quan, G. Zhang, and A. J. Dubinsky, Mediation effect of business process and supply chain management capabilities on the impact of IT on firm performance: Evidence from Chinese firms, Int. J. Inf. Manage., vol. 36, no. 1, pp. 8996, 2016.
4. A. Capaldo and I. Giannoccaro, How does trust affect performance in the supply chain? the moderating role of interdependence, Int. J. Prod. Econ., vol. 166, pp. 3649, 2015.
5. T. Tsiakis and P. Tsiakis, Managing the Risks of Outsourcing IT Security in Supply Chain, in OUTSOURCING MANAGEMENT FOR SUPPLY CHAIN OPERATIONS AND LOGISTICS SERVICE, 2013, pp. 477495.
6. Q. Lu, F. Meng, and M. Goh, Choice of supply chain governance: Self-managing or outsourcing?, Int. J. Prod. Econ., vol. 154, pp. 3238, 2014.
7. E. Crisan, I. Parpucea, and L. Ilies, Models for Supply Chain Governance, Proc. 7th Eur. Conf. Manag. Leadersh. Gov., pp. 535537, 2011.
8. A. I. Pettersson and A. Segerstedt, Measuring supply chain cost, in International Journal of Production Economics, 2013, vol. 143, no. 2, pp. 357363.
9. S. Qrunfleh and M. Tarafdar, Supply chain information systems strategy: Impacts on supply chain performance and firm performance, Int. J. Prod. Econ., vol. 147, no. PART B, pp. 340350, 2014.

10. J. Bastian and J. Zentes, Supply chain transparency as a key prerequisite for sustainable agri-food supply chain management, Int. Rev. Retail. Distrib. Consum. Res., vol. 23, no. 5, pp. 553570, 2013.
11. N. u-Babi, D. Rebolj, M. Nekrep-Perc, and P. Podbreznik, Supply-chain transparency within industrialized construction projects, Comput. Ind., vol. 65, no. 2, pp. 345353, 2014.
12. J. H. Trienekens, P. M. Wognum, A. J. M. Beulens, and J. G. A. J. Van Der Vorst, Transparency in complex dynamic food supply chains, Adv. Eng. Informatics, vol. 26, no. 1, pp. 5565, 2012.
13. P. M. Wognum, H. Bremmers, J. H. Trienekens, J. G. A. J. Van Der Vorst, and J. M. Bloemhof, Systems for sustainability and transparency of food supply chains - Current status and challenges, Adv. Eng. Informatics, vol. 25, no. 1, pp. 6576, 2011.
14. Q. Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments, Information, vol. 8, no. 2, p. 44, 2017.
15. Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, MeDShare: Trust-less Medical Data Sharing Among Cloud Service Providers Via Blockchain, IEEE Access, 2017.
16. R. Hull, V. S. Batra, Y. M. Chen, A. Deutsch, F. F. T. Heath, and V. Vianu, Towards a shared ledger business collaboration language based on data-aware processes, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, vol. 9936 LNCS, pp. 1836.
17. Don Tapscott and Alex Tapscott, How Blockchain Will Change Organizations, MIT Sloan Manag. Rev., no. December 7, 2016.
18. D. Tapscott and A. Tapscott, The Impact of the Blockchain Goes Beyond Financial Services, 2016.
19. S. A. Abeyratne and R. P. Monfared, Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger, Int. J. Res. Eng. Technol., vol. 5, no. 9, pp. 16, 2016.
20. F. Tian, An Agri-food Supply Chain Traceability System for China Based on RFID and Blockchain Technology, 2016 13th Int. Conf. Serv. Syst. Serv. Manag., pp. 16, 2016.
21. I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, Untrusted business process monitoring and execution using blockchain, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, vol. 9850 LNCS, pp. 329347.