Chapter 4

End-to-End Tracing and Congestion in a Blockchain:

A Supply Chain Use Case in Hyperledger Fabric

Kosala Yapa Bandara
https://orcid.org/0000-0002-3616-7764
National University of Ireland Galway, Ireland

Subhasis Thakur
National University of Ireland Galway, Ireland

John G. Breslin
https://orcid.org/0000-0001-5790-050X
National University of Ireland Galway, Ireland

ABSTRACT

Modern supply chain applications are complex systems that play an important role in many different sectors. Supply chain management systems are implemented to handle increasing complexity and flows of goods. However, most of these systems are also increasing the complexity of providing trust and a global view of transactions in a distributed supply chain system. Blockchain technology introduces a new architectural style to support the traceability and trust of transactions performed by participants in a network. This chapter uses this emerging technology to realize a supply chain use case from JLP Meats in the UK with improved transparency, trust, and end-to-end querying while discussing potential challenges of realizing large-scale enterprise blockchain applications. The process of farm-to-fork is implemented and tested for traceability, item recall, block analysis, congestion enabling food safety, and sustainable agriculture. Potential challenges are highlighted in complex supply chains that need heterogeneous trade compliance and scalability.

INTRODUCTION

Supply Chain management is an integration of business processes that are implemented in distributed and heterogeneous systems from end-users to original suppliers (Cooper, Lambert, & Pagh, 1997). Current supply chain management systems have known limitations and the food supply chain is the most complex and fragmented of all supply chains (Martin, 2017). There are many participants involved in a supply chain and they are using distributed and heterogeneous systems increasing the complexity of integration, sharing information, end-to-end tracking, and compliance tracking. Moreover, various systems integrated with a supply chain can be exposed to cyber threats which will result in breaching the integrity of information in the supply chain (Gao, et al., 2018).

Blockchain Technology has emerged as a solution to the double-spending problem that promises traceability, immutability, and transparency of transactions (Nakamoto, 2008). As stated by Consensys (2020), the blockchain technology coupled with smart contracts can enable:

- Transparency of consumer goods from the source point to end consumption
- Accurate asset tracking
- Enhance the licensing of services, products, and software.

The shared IT infrastructure of blockchain can streamline workflows of all participants irrespective of the size of the business network. Moreover, this shared infrastructure enables the auditor greater visibility into the participant's activities along the supply chain.

In the context of supply chain for the food industry, the farm-to-fork food system is a complicated network of isolated systems. There is no widely adopted industry standard regarding how to record and track data for food traceability purposes. Since blockchain technology is emerging as a distributed, trusted, and immutable ledger, it can be used to record transactions in farm-to-fork food systems enabling traceability (Martin, 2017). The number of transactions in a supply chain network is always huge. As an example, Walmart is serving 260 million customers every week across 28 countries in nearly 12,000 stores (Yiannas, 2018). A few of such participants in one blockchain network create millions of transactions and blocks which are continuously growing, challenge the scalability of blockchain networks.

Moreover, product companies in a supply chain network are producing thousands of various types of products before distributing them to their clients. Some of the detail of these transactions is redundant information. For example, thousands of packets of meat are made from one commodity hence only the packet identifier is different. Moreover, most of this information is needed for a certain limited period. Therefore, creating blocks of transactions for these types and have them stored in distributed ledgers of all other participants is a costly process in terms of congestion in the blockchain network and storage. On the other hand, having the same copy of records in all the ledgers support item traceability and auditing. However, end-to-end tracing of items is necessarily required in the modern complex supply chain systems.

The globalisation of the business sector has increased the cross-border movement of commodities and goods, and hence increased the complexity of global supply chains (Martin, 2017). The regulator's role in a blockchain is extremely challenging in current complex supply chains with diverse established old laws, regulations, and institutions distributed in various countries (Kshetri, 2018). Playing a monitoring role as in Gao, et al (2018) is not sufficient since they need to approve or reject transactions providing reasons. Regulators are interested in only the relevant information required for compliance. It is not required for them to know heterogeneous transactions happening in various contexts of regulations.

Current blockchain architecture supports distributed ledgers of the equal state. A regulatory organization to become a participant of all the blockchain networks which need regulatory compliance and maintain ledgers of them is not a practical approach. Hence, the position of a regulatory organization in a blockchain network is still not clear.

The public blockchain frameworks are optimised for transparency hence they create challenges to share private and sensitive information and enable only authorised participants to participate in a supply chain; for example, Bitcoin (Bitcoin, 2020), Ethereum (Ethereum, 2020) and Litecoin (Litecoin, 2020). However, the permissioned blockchain platforms separate transactions into public transactions and private transactions and also enable authorisation, for example, Hyperledger Fabric (Hyperledger, 2020) and Quorum (Quorum, 2020). Hyperledger Fabric has a reliable technology stack to implement supply chain applications compared to other private and permissioned blockchains and that has been highly used for pilots in major organisations such as IBM and Walmart (Kshetri, 2018). Hyperledger Fabric is a modular blockchain framework that supports plug-and-play components that are aimed for use within private enterprises (Kenton, 2020). However, most of these pilots are focussing on end-to-end tracing and still need to elaborate on other challenges such as item recall, blockchain congestion, data redundancy, scalability, regulatory compliance, etc.

In this chapter, the authors discuss a supply chain use case implemented in Hyperledger Fabric (Hyperledger, 2020) for JLP Meats (JLP Meats, 2020) in the UK. Authors have selected Hyperledger Fabric framework because of its promise in developing supply chain applications, modular architecture which supports plug-and-play components required for private enterprises, and reliable technology stack for development and testing. Authors elaborate on the use case for end-to-end tracing, item recall, blockchain congestion, data redundancy, scalability, and position of a regulatory organization in a large scale blockchain network.

The organisation of this chapter is as follows: There are five main sections. Firstly, the authors discuss related work in the literature. Then, the authors discuss end-to-end tracing, blockchain congestion, and regulator's position in a blockchain. In the next section, this chapter describes a use case of JLP Meats, a wholesale meat distributer in London, UK. The fourth section of the chapter describes the design, implementation, and testing using Hyperledger Fabric to elaborate end-to-end tracing, item recall, and congestion analysis. Finally, the chapter summarises the contribution and highlights the future work and directions.

RELATED WORK

In this section, the authors review related work in blockchain technology and supply chain management systems for security, privacy, traceability, item recall, and transaction congestion.

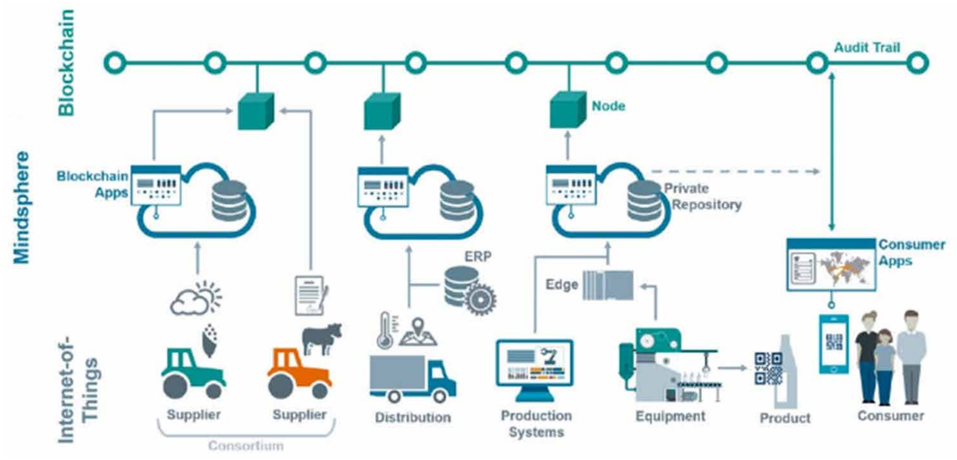
Supply chain management is an integration of business processes that are implemented in distributed and heterogeneous systems from end-users to original suppliers (Cooper, Lambert, & Pagh, 1997). The participants in a large supply chain are generally operating in so many countries under various constraints and legislations. Besides, the food supply chains are trying to provide a more diverse, convenient, and economical source of food while facing enormous new challenges. Moreover, in today's food supply system, the output from one ingredient producer could end up in thousands of new products on a grocery store shelf (Yiannas, 2018). This distributed behaviour and the complexity of food supply chains becomes more complicated when also taking necessary actions for food contamination concerns such as the peanut butter Salmonella outbreak[1] in 2008, the E. coli. illness caused by contaminated flour[2] in 2016, Outbreak of E. coli. infections linked to clover sprouts[3] in 2020, etc. The food supply chain also suffers dynamic costs, prices, and regulatory compliance.

There is no widely adopted industry standard for how each segment of the food supply chain (farmer, processor, distributor, retailer, etc.) tracks and records data for food traceability purposes (Yiannas, 2018). Most of the participants are still recording their data on paper or their legacy systems which do not enable necessary mapping records and communications needed for detailed traceability and transparency. Moreover, current supply chain applications suffer from the integration of isolated systems and security breaches regarding the integrity of transactions. Radio-frequency identification (RFID), telematics, barcode and 2D codes, sensors-enabled technologies, Internet of Things(IoT), and numerous other technologies are used for tracking products through supply chains (Davor & Domagoj, 2018). However, the true potential of tracking data is not fully exploited as the underlying data is available only within companies or partially connected isolated systems. The communication between systems and the lack of trust between the segments of the systems are the main concerns in the current supply chain systems. The supply chain-related sustainability incidents suggest that firms with a global presence struggle to improve environmental, social, and economic outcomes in global supply chains (Esteban Koberg, 2018). The firms should be accountable for the environmental, social, and economic outcomes caused by their internal and supplier operations.

On the other hand, blockchain technology has emerged as a solution to the double-spending problem that promises traceability, immutability, and transparency of transactions (Nakamoto, 2008). The blockchain technology has evolved from cryptocurrency transactions and disrupts constantly enlarging areas of the economy (Davor & Domagoj, 2018). This technology can provide improved traceability, transparency, and tradability for supply chain systems (Consensys, 2020). Thus, blockchain and distributed ledger technologies are becoming increasingly popular in the supply chain applications domain (Martin, 2017). The blockchain technology promises overpowering trust issues and allows a secure and authenticated system for logistics and supply chains. This lead to revolutionise supply chain systems using blockchain technology. Pilot projects already exist within big organisations; for example, the farm-to-fork[4] process can adopt IoT and blockchain technologies to improve control and flexibility while increasing food trust and brand protection.

Siemens merges its Mindsphere platform, private track and trace repositories, and blockchain management applications to implement track and trace use cases as shown in Figure. 1 (Siemens, 2019). These solutions use the blockchain technology to implement trust throughout the process. This framework allows companies to limit information viewing privileges hence sensitive information can be kept behind closed doors while exposing only critical information to other members of the supply chain. This platform further enables leveraging IoT and blockchain for a “digital twin” (Meyvaert, 2020). A digital twin in IoT is a virtual representation of a physical product or process, used to understand and predict the physical counterpart’s performance characteristics (Meyvaert, 2020). Digital twins are used throughout the product lifecycle and blockchain enables them to record all the information in immutable records.

Figure 1. Siemens Mindsphere for Supply Chains



Blockchain technology can enable the creation of a decentralised, distributed, and trusted digital ledger that can be used to record transactions from multiple entities across a complex network (Yiannas, 2018). Data immutability and public accessibility of data streams can support compliance, reliability, and transparency of modern supply chain systems (Perboli, Musso, & Rosano, 2018). A supply chain as a blockchain, is a two-step block construction mechanism as suggested by Gao et al. (2018) classifies participants as ordinary users, third-party users, and supporting entities with different roles. In their approach, regulatory organizations are focusing on monitoring information. A monitoring agent to monitor and analyse blockchain transactions, nodes, blocks, and smart contracts to ensure blockchains operate legally, efficiently, and reliably are discussed in (Ko, Lee, Jeong, & Hong, 2018).

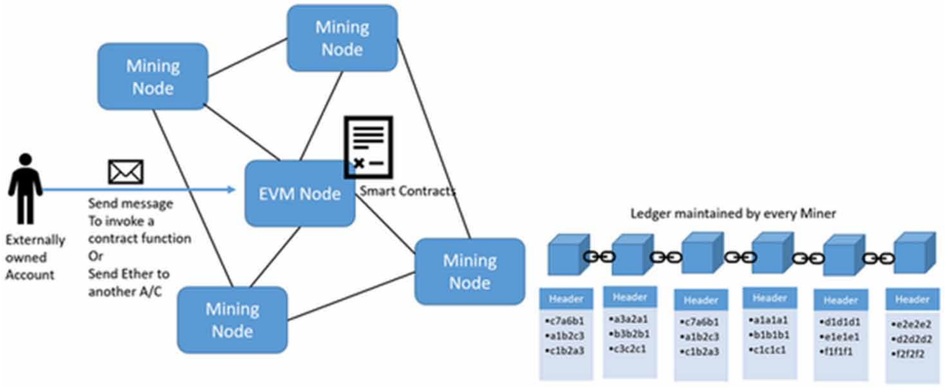
Authors in (Kshetri, 2018) summarize a set of successful use cases of blockchain implementations including: Danish shipping company Maersk (a blockchain application for international logistics), Provenance (a pilot project in Indonesia to enable the traceability in the fishing industry), Alibaba (a blockchain to fight for food fraud), Walmart (tracking produce from Latin America to the US, and Intel's solution to track seafood supply chain, etc.

Traceability in a supply chain is an important area to explore (Westerkamp, Victor, & Küpper, 2018). In the current blockchain architectures, distributed ledgers provide transaction information accessible to all the participants in the blockchain network providing greater transparency (Zheng, Xie, Dai, Chen, & Wang, 2017). However, organizations are reluctant to expose sensitive information in a public ledger, and hence private data collections are introduced by Benhamouda, Halevi & Halevi (2018) to manage sensitive information. Privacy, scalability, and lack of governance are still major concerns for large scale industrial adaptation of blockchain paradigms (Li, Sforzin, Fedorov, & Karame, 2017).

The public (permissionless) blockchain platforms are optimised for transparency, and transactions are public and transparent, for example, Bitcoin (Bitcoin, 2020), Ethereum (Ethereum, 2020), and Litecoin (Litecoin, 2020). However, the permissioned blockchain platforms separate transactions into public transactions and private transactions, for example, Hyperledger Fabric (Hyperledger, 2020) and Quorum (Quorum, 2020). The private transactions share private and sensitive data between participants in a network (Hyperledger2, 2020).

Ethereum is a secure decentralised ledger that is optimized for transparency, hence it is difficult to share secrets on the platform (Ethereum, 2020). The main components of Ethereum are Ethereum Virtual Machine (EVM), miner, block, transaction, consensus algorithm, account, smart contract, mining, Ether, and gas (Modi, 2018). These are illustrated in Figure. 2. The notion of private transactions and public transactions are introduced in Quorum (Quorum, 2020) which extends the transaction model of Ethereum to include an optional privateFor parameter and a new IsPrivate method to deal with such transactions.

Figure 2. Components of Ethereum



On the other hand, Hyperledger Fabric introduces private data collections, which allow a defined subset of organizations on a channel the ability to endorse, commit, or query the private data (Hyperledger2, 2020). The private data is sent peer-to-peer via gossip protocol to only the organisations authorised to see it. The ordering service is not involved here and the orderer does not see the private data. The hash of the private data is endorsed, ordered, and written to the ledgers of every peer on the channel as in Figure. 3 (Hyperledger2, 2020).

Figure 3. Private Data Store in Hyperledger Fabric

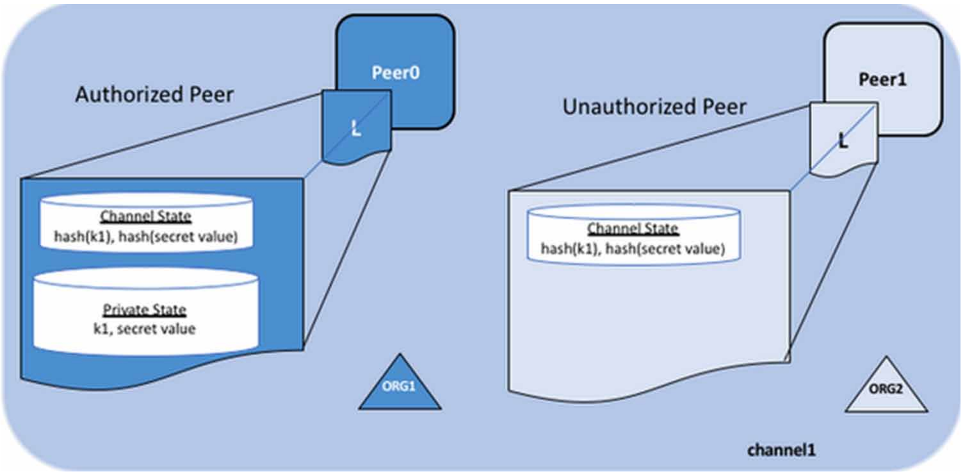


Figure 4. Transactions Invocation in Hyperledger Fabric

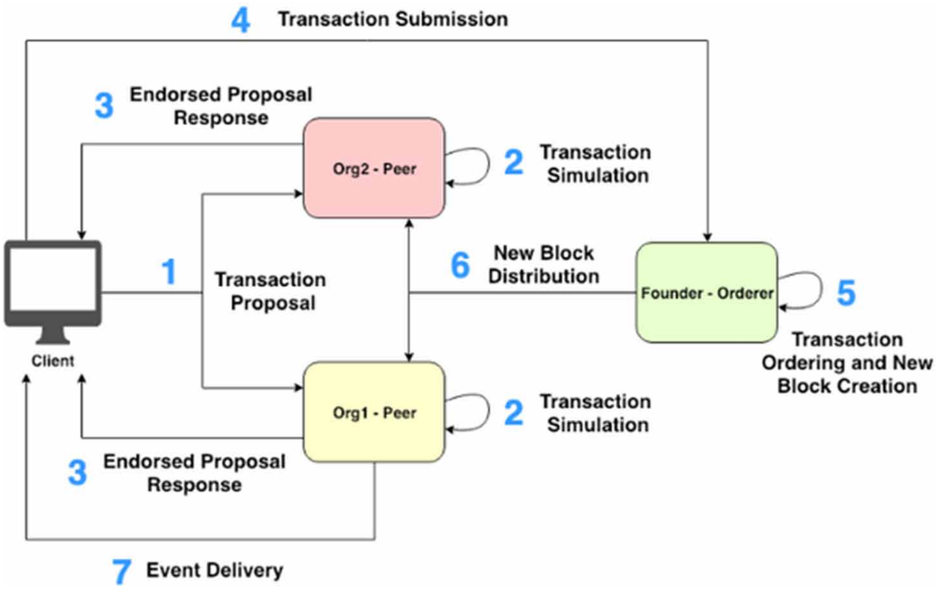
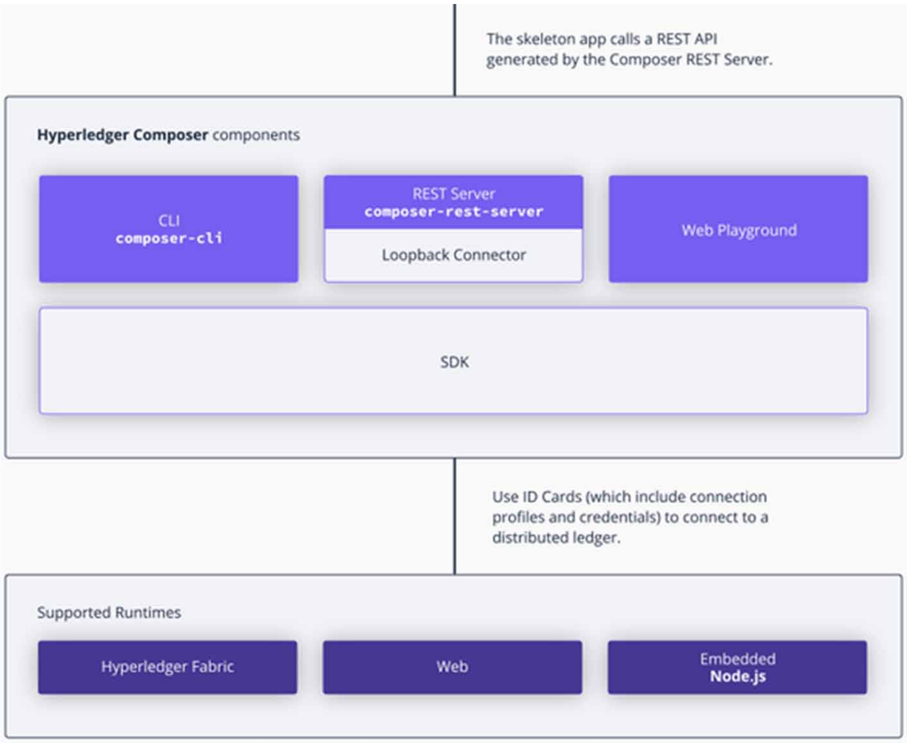


Figure 5. Hyperledger Fabric and Composer Technology Stack



The skeleton app calls a REST API generated by the Composer REST Server.

Use ID Cards (which include connection profiles and credentials) to connect to a distributed ledger.

The hashes of private data go through the orderer to the public ledger and preserve privacy. The hash can be used for state validation and audit purposes. The flow of transactions invocation in Hyperledger Fabric is illustrated in Figure. 4 (Thummavet, 2020).

The Hyperledger fabric technology stack for blockchain applications is presented in Figure. 5 (Composer, 2020).

The technology stack in Figure. 5 enables architects and developers to quickly create "full-stack" blockchain solutions: for example, business logic that runs on the blockchain, REST APIs that expose the blockchain logic to the web or mobile applications, blockchain integration with existing systems, etc. (Composer, 2020). Hyperledger composer has been designed to support pluggable runtimes. The modular architecture of Hyperledger fabric separates the transaction processing workflow into three different processes: smart contracts called chaincode that comprises the distributed logic processing and agreement of the system, transaction ordering, and transaction validation and commitment (Kenton, 2020).

END-TO-END TRACING AND CONGESTION

Blockchain stores data chronologically in blocks that are chained together in a continuously growing series. Participants in the network are contributing to commit transactions and blocks into the blockchain. Adding blocks to a blockchain cannot be predicted and blocks are not sequenced based on transactions. However, all the distributed ledgers have the same state enabling reliable access and supporting end-to-end traceability of records from one ledger.

Figure 6. Transactions and Blocks in a Blockchain

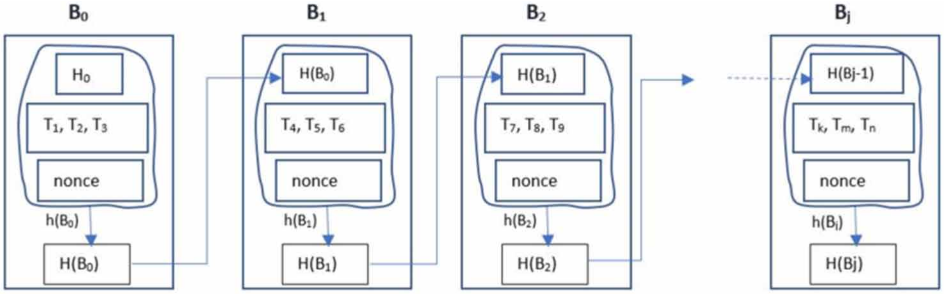
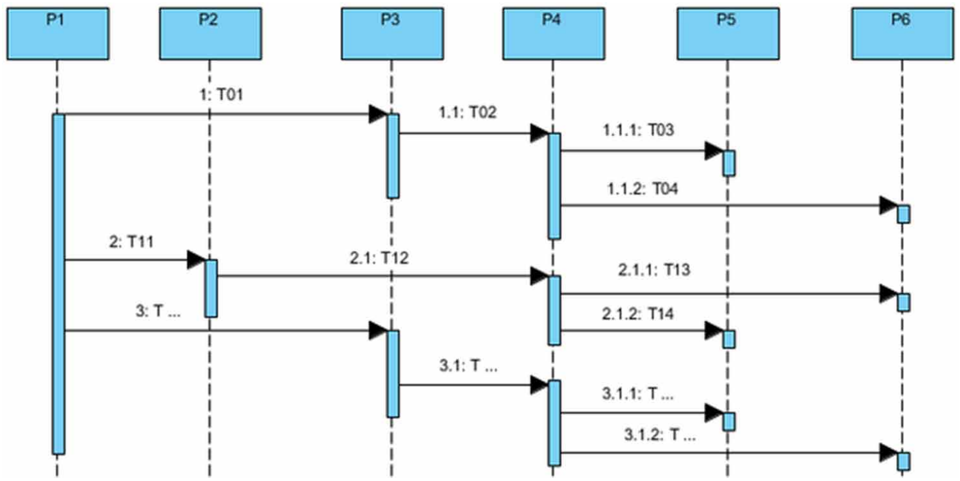


Figure. 6 illustrates a blockchain where B_0 to B_j are constituent blocks in the blockchain. $H(B_0)$ to $H(B_j)$ are hash keys of blocks. T_1 to $T_{k,}$ T_m, T_n are transactions. k, m, n are Integers. h () is the hash function. Here, the authors assume a block is created including three transactions. Blockchain grows continuously based on the transactions created by the participants in the network. Participants will transact based on their own needs and blocks are added to the continuously growing blockchain. Transaction information retrieval is a challenge because there is no relationship between transactions. For example, item traceability needs all the transactions related to one item, auditing needs to audit a set of transactions, etc. However, different blockchain platforms provide various methods to retrieve transaction information. Hyperledger Fabric has the composer-rest-server which provides REST endpoints for each asset.

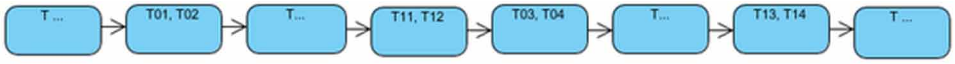
Figure. 7 shows participants P1...P6 and transactions T01 …T04 and T11 … T14. In a supply chain application, P1 to P6 can be considered as, commodity providers, logistic companies, importers, and retailers respectively.

Figure 7. Sequence of transactions in a blockchain



Commodity provider provides commodities continuously of various types through logistic companies to the importer company who will produce batches of products from the commodities and supply them to clients. Participants in the blockchain network contribute to add blocks to the blockchain network as shown in Figure. 8.

Figure 8. State change in the blockchain



The end-users finding end-to-end details about a product, that involves a set of transactions distributed in randomly distributed blocks in a blockchain, are not straightforward in the current blockchain architecture. Though blocks are connected using block hash mechanism, the chain concept and random blocks integration make it complicated to find the evolution of an item in a supply chain. However, the decentralized nature of records and having the same copy of up-to-date ledger allow participants to directly interact with end-to-end details of data.

The authors have created a REST-API which is creating REST endpoints to members on the blockchain, asserts, user-defined queries, and transactions in the blockchain. Queries are defined to extract necessary information from the ledger. Representational state of these resources can be extracted, filtered, and connected to find end-to-end details. The processes should be defined accordingly.

Blockchain Congestion

In supply chain applications, end-user products have gone through a list of transactions in the process of transforming resources to end-user products. There are intermediate participants in the blockchain network who develop the main resource into various products and distribute them to clients. These scenarios will add millions of records to the blockchain. This leads to several challenges regarding block congestion in the blockchain network and data redundancy at the participants. All the transactions happening in the product life cycle are not equally important to all the participants in the blockchain network, hence a compensation model for transaction verification is needed. As in Figure. 7, P4 makes different products from the main resource and distributes them to P5 and P6.

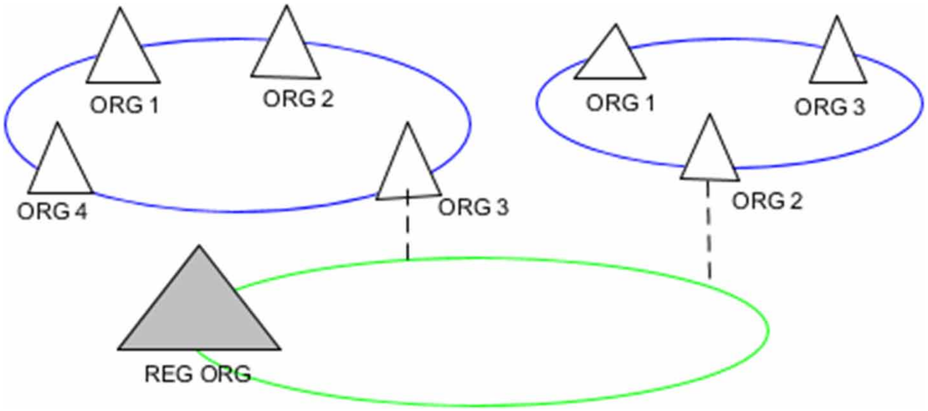
(T_{01}, T_{02}) -> $(T_{1} \ldots T_{n1}), (T_{n1} \ldots T_{n2}), (T_{n2} \ldots T_{n3})$, …… where n1, n2, n3 $\epsilon\ \mathbb{Z}$
(T_{11}, T_{12}) -> $(T_{11} \ldots T_{m1}), (T_{m1} \ldots T_{m2}), (T_{m2} \ldots T_{m3})$, ... where m1, m2, m3 $\epsilon\ \mathbb{Z}$
$(T_{\ldots})$ -> $(T_{21} \ldots T_{k1}), (T_{k1} \ldots T_{k2}), (T_{k2} \ldots T_{k3})$, ... where k1, k2, k3 $\epsilon\ \mathbb{Z}$

At P4, transactions T_{01} and T_{02} result in creating sets of transactions based on product creation and requirements of clients. The same will apply to T_{11}, $T_{12,}$ and $T_{\ldots}$. If there are several intermediate participants of the type P4 who develop sub-products, this will add millions of records to the blockchain developing real-time congestion in the blockchain network. Moreover, this will add redundant data to the blockchain. This is a continuously growing real-time overhead which brings a negative impact to the scalability of blockchain networks. Managing temporal data to reduce transaction redundancy and transaction verification for supply chain applications still need major developments. IoT devices are integrated with supply chains and they generate millions of records throughout the supply chain. Ad hoc solutions can be adopted to manage these records, for example, Hyperledger Fabric support plug-and-play local data stores to record the IoT data, and only the hashes of sets of data are recorded in the blockchain. IoT integration to supply chains is a common scenario, hence new standards are necessary for blockchain integration with IoT.

Moreover, the supply chain applications need to comply with regulations set up in various territories. Regulatory organisations in various territories are interested in only a specific set of information regarding supply chain transactions for compliance checking. In the current blockchain architecture, participants will maintain the complete blockchain hence regulator's role as a participant in the blockchain network is not practical. So far, none of the blockchain architectures provide necessary standards to position regulators in blockchains.

As shown in Figure. 9, the authors propose a private channel for regulatory organizations to connect with only the required participants for compliance checking. Referring to Figure. 9, ORG 1 to 4 and ORG 1 to 3 are connected in two blockchain networks. REG ORG has a private channel connecting ORG3 and ORG 2 of two different blockchain networks. Regulatory organizations in various territories can become authorized members of this private channel and that eases quick validation of supply chain transactions without long delays as in the current system. This can further support transparent compliance checking for anyone who needs to send items through a supply chain. This topic area needs further elaboration because there are no proper standards or mature products so far, to fast track regulatory compliance and blockchain technology has shown a lot of promise in this area.

Figure 9. Regulatory organization in blockchains

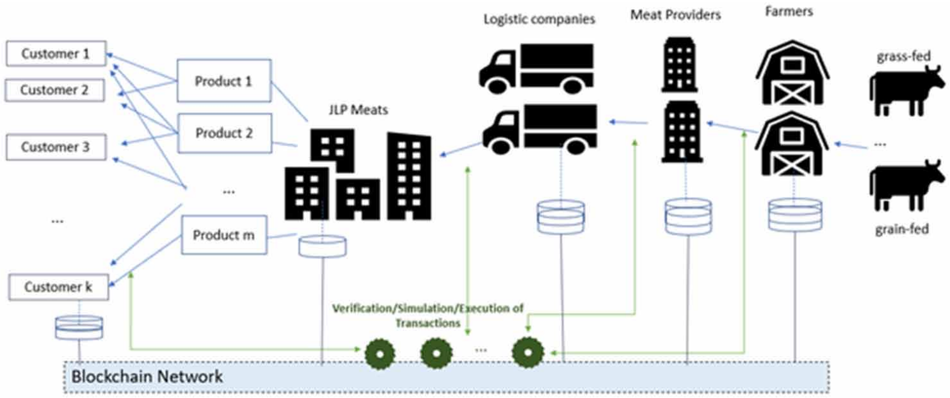


In our proposed architecture, transactions that need regulatory compliance will be directed to the regulatory organization, and distributed ledgers are updated with the blocks of approved or rejected transactions. A block of approved or rejected transactions are kept in the regulator data store as *{hash(block), block}*. The hash of the block is recorded on the blockchain. In this approach, regulators do not need to keep unnecessary records and they only keep records of their processed records (approved or rejected records). Future applications can use this recorded data. For example, since these blocks have necessary hash keys on the blockchain, audit trails can be done connecting to a ledger of a participant in the network.

USE CASE

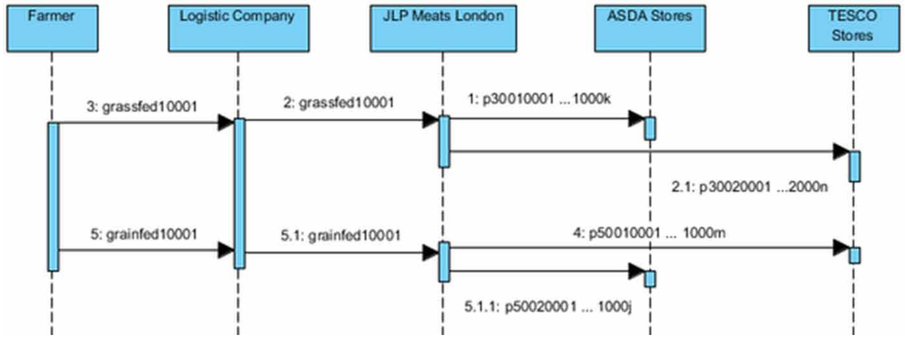
JLP Meats Trading organization (JLP Meets, 2020) imports meat from Australia, South America, and Europe, and sells in the UK. Their customers are restaurants, retail shops, and butchers in the UK. JLP Meats produces hundreds of tailor-made meat products and distributes them to customers. An abstract view of the farm-to-fork process is illustrated in Figure. 10. The authors have implemented a blockchain solution using Hyperledger Fabric to illustrate end-to-end tracing, item recall, transaction congestion, etc.

Figure 10. Use Case of JLP Meats, London



For the illustration purposes, authors considered a specific scenario as follows. Farmer Samex uses the logistic company Sandford group to export beef of the type grass-fed and grain-fed to JLP Meats in the UK. JLP Meats transforms beef into packets of 300 to 500 grams and supply them to their customers ASDA and TESCO stores. A buyer who buys a packet of beef from TESCO (or ASDA) wants to find end-to-end details about the product. Here, a typical set of ttransactions can be modelled as shown in Figure. 11.

Figure 11. Transactions in the Use Case scenario



JLP Meats supplies various types of products that create thousands of transactions improving transaction congestion in the blockchain. Transactions verification is done by permitted verifiers in the network and this process should be compensated. However, this chapter does not cover the compensation process; and the quality control is governed by regulators. This chapter proposes a private channel network of regulators who can connect to blockchain networks to perform regulatory activities and necessary endorsements as illustrated in Figure. 9.

DESIGN, IMPLEMENTATION AND TESTING

The authors use a public and permissioned blockchain so that ledgers are decentralized, and a selected set of nodes participates in consensus procedure. The authors have implemented the above use case using Hyperledger Fabric[5], Composer Playground, Hyperledger Explorer, and Docker on Ubuntu 18.04.

The high-level architecture for the proposed solution using Hyperledger Fabric is illustrated in Figure. 12.

The peers are members of the blockchain network. The business network was developed and installed in the above peers and a REST-API[6] was created to perform end-to-end tracing of items. Hyperledger composer playground was used to implement and test scenarios and Hyperledger blockchain explorer was used to visualize blockchain statistics. Swagger[7] was used to visualise and test REST endpoints. The trading transaction for packets of meat was defined as in Figure. 13. The peers can be authorised miners or participants to mine supply chain transactions and a compensation model should be introduced for transaction verifications. However, there are no proper standards or miners available so far to mine supply chain transactions. These necessary developments can further revolutionise future supply chain systems.

Figure 12. The High-level Architecture Based on Hyperledger Fabric

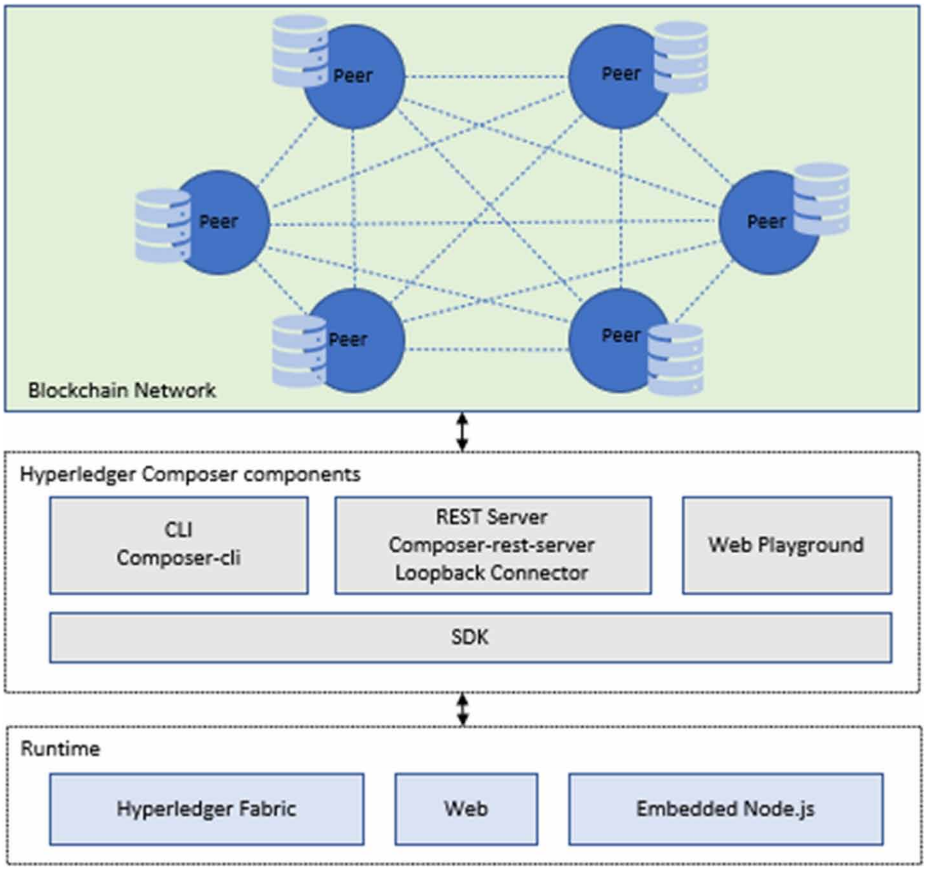
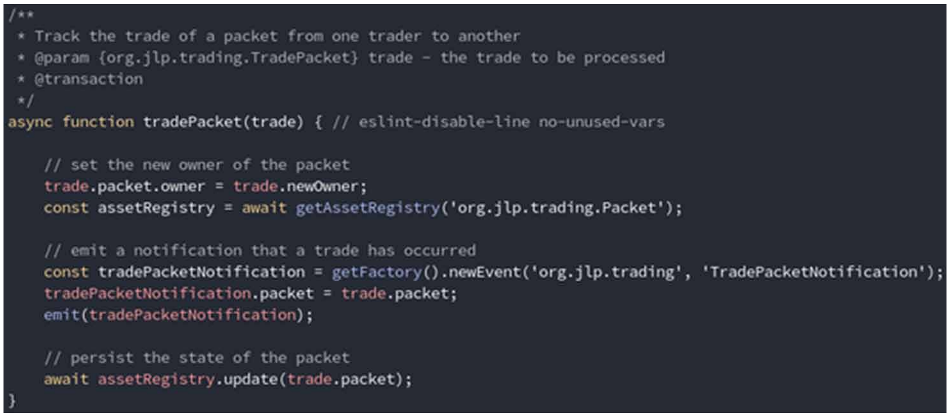


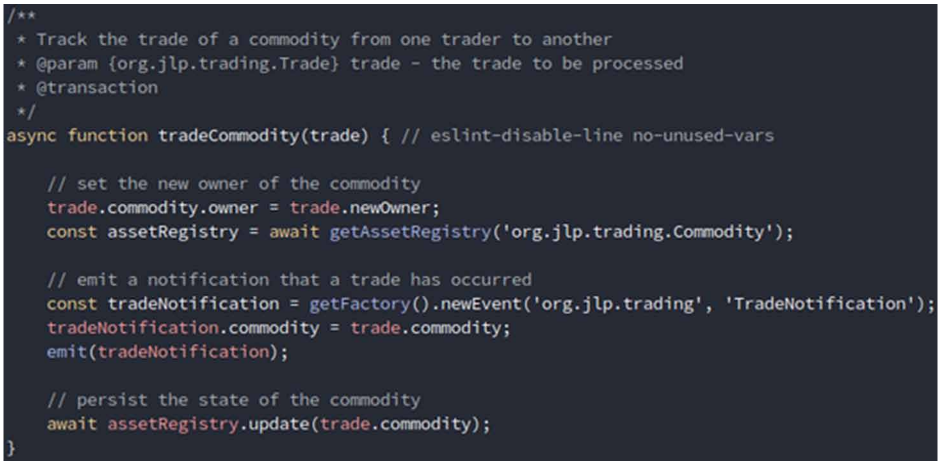
Figure 13. Trading Transaction for Packets



```
/**
 * Track the trade of a packet from one trader to another
 * @param {org.jlp.trading.TradePacket} trade - the trade to be processed
 * @transaction
 */
async function tradePacket(trade) { // eslint-disable-line no-unused-vars

    // set the new owner of the packet
    trade.packet.owner = trade.newOwner;
    const assetRegistry = await getAssetRegistry('org.jlp.trading.Packet');

    // emit a notification that a trade has occurred
    const tradePacketNotification = getFactory().newEvent('org.jlp.trading', 'TradePacketNotification');
    tradePacketNotification.packet = trade.packet;
    emit(tradePacketNotification);

    // persist the state of the packet
    await assetRegistry.update(trade.packet);
}
```

Referring to Figure. 13, the trading transaction has three main functionalities: setting the ownership for a packet of meat, emitting a notification that a trade has occurred, and persisting the state of the packet of meat. These are mostly common functionalities for trading transactions. The trading transaction for a commodity is defined in Figure. 14.

Figure 14. Trading Transaction for Commodities



```
/**
 * Track the trade of a commodity from one trader to another
 * @param {org.jlp.trading.Trade} trade - the trade to be processed
 * @transaction
 */
async function tradeCommodity(trade) { // eslint-disable-line no-unused-vars

    // set the new owner of the commodity
    trade.commodity.owner = trade.newOwner;
    const assetRegistry = await getAssetRegistry('org.jlp.trading.Commodity');

    // emit a notification that a trade has occurred
    const tradeNotification = getFactory().newEvent('org.jlp.trading', 'TradeNotification');
    tradeNotification.commodity = trade.commodity;
    emit(tradeNotification);

    // persist the state of the commodity
    await assetRegistry.update(trade.commodity);
}
```

End-to-End Tracing

Tracing end-to-end details were achieved using REST endpoints. These endpoints and their outcomes to achieve details of a packet, the commodity and its trader are detailed as follows:

```
http://localhost:3000/api/Packet/p3001001:
{
 “$class”: “org.jlp.trading.Packet”,
 “packetID”: “p3001001”,
 “mainExchange”: “GBP”,
 “quantity”: 300,
 “unitprice”: 2,
 “commodity”: “resource:org.jlp.trading.Commodity#grassfedbeef”,
 “owner”: “resource:org.jlp.trading.Trade#tesco”
}
http://localhost:3000/api/Commodity/grassfedbeef:
{
 “$class”:”org.jlp.trading.Commodity”,
 “tradingSymbol”: “grassfedbeef”,
 “description”: “Grassfed Beef”,
 “mainExchange”: “AUD”,
 “slaughterDates”: “20th December”,
 “quantity”: 500,
 “owner”: “resource:org.jlp.trading.Trader#jlp”
}
http://localhost:3000/api/Trader/jlp:
{
 “$class”:”org.jlp.trading.Trader”,
 “tradeId”: “jlp”,
 “name”:”JLP Meat”,
 “address”: “London”
}
```

The authors developed queries to retrieve information and deployed them as REST endpoints. The following REST endpoints output transaction details regarding trading commodities and farmer details. The endpoint to retrieve farmer details is a query-based REST endpoint.

```
http://localhost:3000/api/Trade:
{“$class”: “org.hyperledger.composer.system.AddAsset”,
“resource”:[
        {
          “$class”: “org.jlp.trading.Commodity”,
          “tradingSymbol”: “grassfedbeef”,
          “description”: “Grassfed Beef”,
          “mainExchange”: “AUD”,
          “slaughterDates”: “20th December”,
          “quantity”: 500,
          “owner”: “resource:org.jlp.trading.Trader#samex”
}
]}
http://localhost:3000/api/queries/findAddAssertsTradeTransactions:
{
 “Packet id”: “p3001001”,
 “Type of Beef”: “grass-fed beef”,
 “Trader”: “JLP Meats”,
 “Logistic company used”: “Sandford”,
 “Farmer name”: “Samex”,
 “Slaughtered Date”: “20th December”
}
```

The authors extracted the above details using REST endpoints and arranged them in a process to generate end-to-end details about a specific packet. That is, developers’ intervention is needed to develop workflows connecting the endpoints to filter and provide requested information. This requested information can be a simple user request or a complex audit trail. However, this process enables users, participants and regulatory organisations to view end-to-end details, for example, farmers to get to know where their meat is being sold, some fair trade organisations to view records and ensure fair trade policies are properly applied, organisations to review statistics to ensure sustainable agriculture, etc.

Item Recall

In supply chain applications, it is necessary to quickly trace unsafe products back to their source and where they have been distributed.

In this implementation, the REST endpoint, *http://localhost:3000/api/Packet/p3001001* provides details of the packet p3001001.

```
{
“$class”:” org.jlp.trading.Package”,
“packetID”: “p3001001”,
“mainExchange”: “GBP”,
“quantity”: 300,
“unitprice”: 2,
“commodity”: “resource.org.jlp.trading.Commodity#grassfed10001”,
“owner”: “resource.jlp.trading.Trade#tesco”
}
```

Finding all the packets made from the commodity “grassfed10001” can be extracted using the following REST endpoint and the filter ***http://localhost:3000/api/Packet***

```
where “:{”commodity”:”resource:org.jlp.trading.Commodity#grassfed10001”}}.
[
{
 “$class”: “org.jlp.trading.Packet”,
 “packetID”: “p3001001”,
 “mainExchange”: “GBP”,
 “quantity”: 300,
 “unitprice”: 2,
 “commodity”: “resource:org.jlp.trading.Commodity#grassfed10001”,
 “owner”: “resource:org.jlp.trading.Trade#tesco”
},
{
“$class”: “org.jlp.trading.Packet”,
 “packetID”: “p3001002”,
 “mainExchange”: “GBP”,
 “quantity”: 300,
 “unitprice”: 2,
 “commodity”: “resource:org.jlp.trading.Commodity#grassfed10001”,
 “owner”: “resource:org.jlp.trading.Trader#tesco”
},
{
 “$class”: “org.jlp.trading.Packet”,
 “packetID”: “p3001003”,
 “mainExchange”: “GBP”,
 "quantity": 300,
 "unitprice": 2,
 "commodity": "resource:org.jlp.trading.Commodity#grassfed10001",
 "owner": "resource:org.jlp.trading.Trade#jlp"
},
{
 "$class": "org.jlp.trading.Packet",
 "packetID": "p3001005",
 "mainExchange": "GBP",
 "quantity": 300,
 "unitprice": 2,
 "commodity": "resource:org.jlp.trading.Commodity#grassfed10001",
 "owner": "resource:org.jlp.trading.Trader#asda"
}
]
```

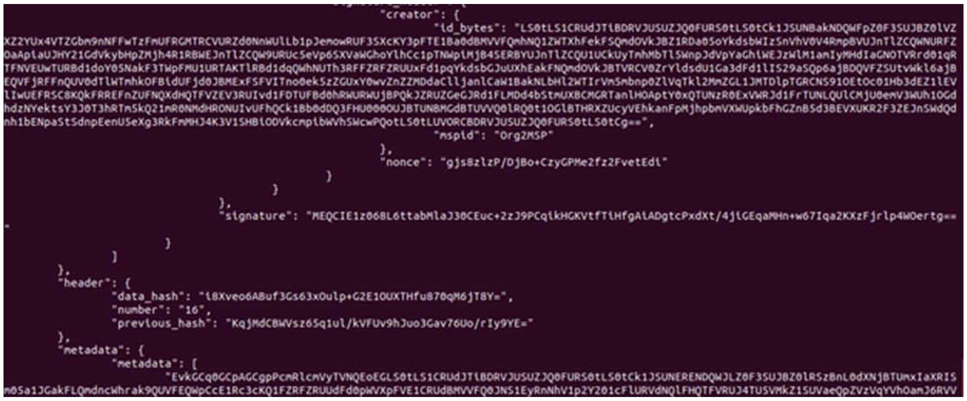
Similarly, farmer and transport details of the commodity can also be found. The recall of items can be processed accordingly for the selected commodities. It is not necessary to recall all the packets provided by JLP or the farmer. This approach helps faster recall of all the items from clients and stores stopping further damage in terms of health and finance. The necessary workflows can be developed to expand as needed.

Blocks in Channels

Peer channel commands[8] implemented in Hyperledger Fabric were used to explore blocks and channels in the blockchain network. In this use case implementation, CLI command *"docker exec -it cli bash"* and *"peer channel getinfo -c mychannel"* were used to connect to a peer and extract blockchain information.

Blockchain info: {"height":37,"currentBlockHash":"avhlBTkndEjbj7GixWkvB9kC7RawTzhKmHhCy7gXHuQ=","previousBlockHash":"+d+lYUBQzoWOXIn/wCRkI7uSAAcCd1Evloyzyau4jGM="}

Figure 15. Details of Block 16



```
                                                  "creator": {
                                                          "id_bytes": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUNBakNDQWFpZ0F3SUJBZ0lVZ
XZ2YUx4VTZGbm9nNFFwTzFmUFRGMTRCVURZd0NnWUlLb1pJemowRUF3SXcKY3pFTE1Ba0dBMVVFQmhNQ1ZWTXhFekFSQmdOVkJBZ1RDa05oYkdsbWIzSnVhV0V4RmpBVUJnTlZCQWNURFZ
OaApiaUJHY21GdVkybHpZMjh4R1RBWEJnTlZCQW9URUc5eVp6SXVaWGhoYlhCc1pTNWpiMjB4SERBYUJnTlZCQU1UCkUyTmhMbTl5WnpJdVpYaGhiWEJzWlM1amIyMHdIaGNOTVRrd01qR
TFNVEUwTURBd1doY05NakF3TWpFMU1URTAKTlRBd1dqQWhNUTh3RFFZRFZRUUxFd1pqYkdsbGJuUXhEakFNQmdOVkJBTVRCV0ZrYldsdU1Ga3dFd1lIS29aSQp6ajBDQVFZSUtvWkl6ajB
EQVFjRFFnQUV0dTlWTmhkOFBldUFjd0JBMExFSFVITno0ek5zZGUxY0wvZnZZMDdaClljanlCaW1BakNLbHl2WTIrVmSmbnp0ZlVqTkl2MmZGL1JMTDlpTGRCNS91OEtOc01Hb3dEZ1lEV
lIwUEFRSC8KQkFRREFnZUFNQXdHQTFVZEV3RUIvd1FDTUFBd0hRWURWUjBPQkJZRUZGeGJRd1FLMDd4bStmUXBCMGRTanlHOAptY0xQTUNzR0ExVWRJd1FrTUNLQUlCMjU0emV3WUh1OGd
hdzNYektsY3J0T3hRTm5kQ21mR0NMdHRONUIvUFhQCk1Bb0dDQ3FHU000OUJBTUNBMGdBTUVVQ0lRQ0t1OGlBTHRXZUcyVEhkanFpMjhpbmVXWUpkbFhGZnBSd3BEVXUKR2F3ZEJnSWdQd
nh1bENpaStSdnpEenUSeXg3RkFmMHJ4K3V1SHBlODVkcmpibWVhSWcwPQotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==",
                                                          "mspid": "Org2MSP"
                                                  },
                                                  "nonce": "gjs8zlzP/DjBo+CzyGPMe2fz2FvetEdl"
                                          }
                                  }
                          },
                          "signature": "MEQCIE1z068L6ttabMlaJ30CEuc+2zJ9PCqlkHGKVtfTlHfgAlADgtcPxdXt/4jlGEqaMHn+w67Iqa2KXzFjrlp4WOertg==
"
                  }
          ]
  },
  "header": {
          "data_hash": "l8Xveo6ABuf3Gs63xOulp+G2E1OUXTHfu870qM6jT8Y=",
          "number": "16",
          "previous_hash": "KqjMdCBWVsz6Sq1ul/kVFUv9hJuo3Gav76Uo/rIy9YE="
  },
  "metadata": {
          "metadata": [
                  "EvkGCq0GCpAGCgpPcmRlcmVyTVNQEoEGLS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUNERENDQWJLZ0F3SUJBZ0lRSzBnL0dXNjBTUmxIaXRIS
m05a1JGakFLQmdncWhrak9QUVFEQWpCcE1Rc3cKQ1FZRFZRUUdFd0pWVXpFVE1CRUdBMVVFQ0JNS1EyRnNhV1p2Y201cFlURVdNQlFHQTFVRUJ4TUSVMkZ1SUVaeQpZVzVqYVhOamJ6RVV
```

[8]

The CLI command *"peer channel fetch 16 -c mychannel"* and *"configtxlator proto_decode --input mychannel_16.block --type common.Block"* enables fetching block 16, decoding, and extracting block information. All inside details of a block (data, payload, actions, creator, nonce, mspid, endorsements, signature, endorser, chain code details, channel details, data hash, previous hash, metadata, etc.) were extracted in this approach as in Figure. 15.

Congestion Analysis

Figure. 16 illustrates transactions in an 8-member blockchain network for 6 imports from the farmer. The authors computed a congestion analysis for a blockchain environment of 8 participants (p1 to p8 – p1: farmer, p2: logistic company, p3: logistic company, p4: JLP meats, p5 to p6 are clients) and 6 imports as illustrated in 6 series. Here, p4 is JLP Meats that creates thousands of products from commodities.

There are more than one P4 type participants in most of the supply chains and that leads to creating transaction congestion on blockchains and delays transaction verification. That is, future blockchain-based supply chain systems should have methods and standards to manage transaction congestion. For example, keeping redundant information locally while recording the hash of this information on the blockchain to assure trust. However, the private channels, local data stores, and plug-and-play modular architecture of Hyperledger Fabric can be improved to develop necessary solutions.

Figure 16. Congestion analysis for 6 imports in the 8-member blockchain network

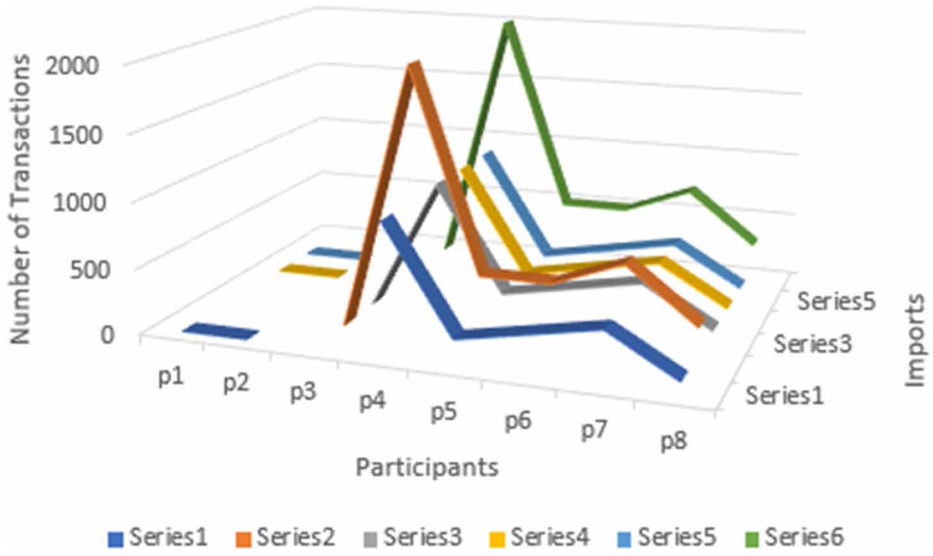
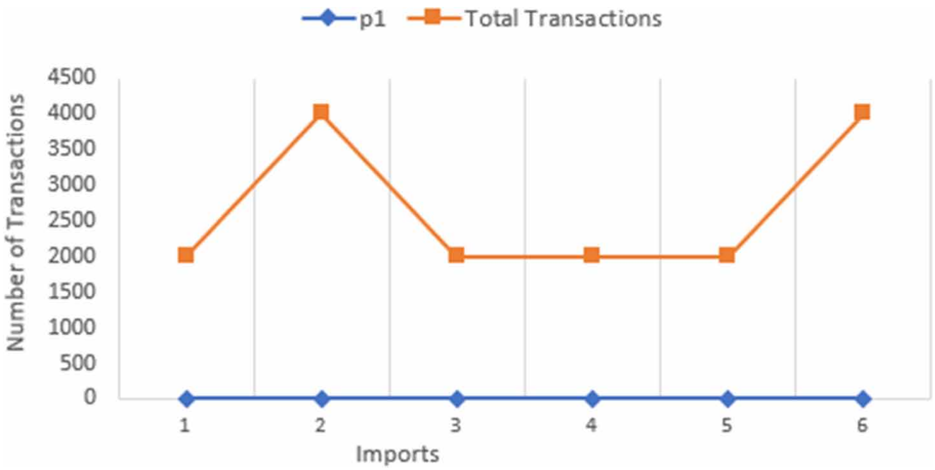


Figure. 17 illustrates a set of transactions for each import. The ordering service in Hyperledger Fabric is managing adding blocks to the blockchain network. P4 is an intermediate participant who creates thousands of products from imports. Having several such participants in a single blockchain network generates many records per import and increases the congestion in the blockchain.

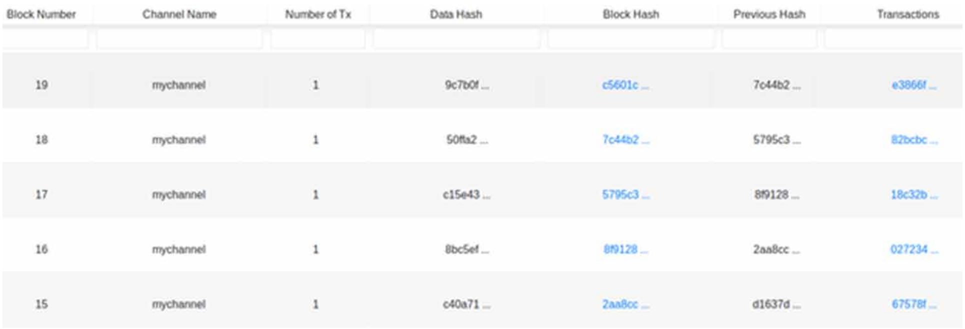
Figure 17. Number of Transactions for Imports



In supply chain applications, the regulator plays the role of checking regulations regarding various types of imports/exports, approving imports/exports, and keeping only the necessary records of them. The regulator cannot maintain ledgers of all the transactions and this proposed approach helps regulators to stay transparent to all other necessary blockchain networks. Moreover, the authors used blockchain explorer to visualize block statistics such as number, block hash, and previous hash as in Figure. 18, Figure. 19, and Figure. 20.

The source code of our POC can be found on https://github.com/kosalayb/JLPMeats.

Figure 18. Blockchain Details

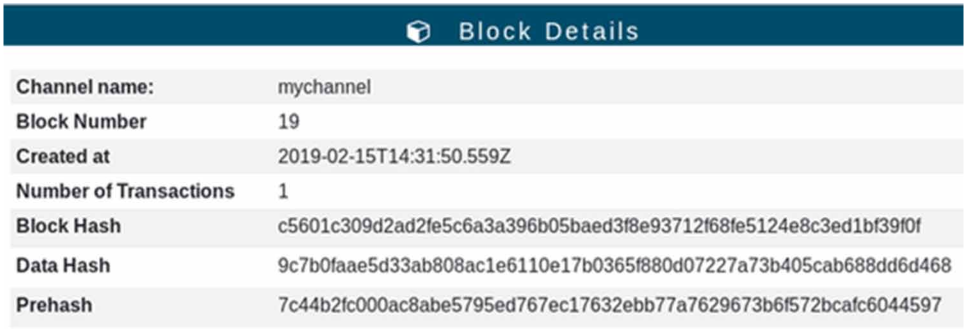


Block Number	Channel Name	Number of Tx	Data Hash	Block Hash	Previous Hash	Transactions
19	mychannel	1	9c7b0f ...	c5601c ...	7c44b2 ...	e3866f ...
18	mychannel	1	50ffa2 ...	7c44b2 ...	5795c3 ...	82bcbc ...
17	mychannel	1	c15e43 ...	5795c3 ...	8f9128 ...	18c32b ...
16	mychannel	1	8bc5ef ...	8f9128 ...	2aa8cc ...	027234 ...
15	mychannel	1	c40a71 ...	2aa8cc ...	d1637d ...	67578f ...

DISCUSSION AND CONCLUSION

Blockchain Technology is a potential technology to realize complex supply chain applications because of its nature of immutable and distributed ledgers which help to ease traceability and ensure the trust of heterogeneous transactions. Traceability and trust of transactions are extremely important in supply chain applications. Most of the pilot studies presented in the literature discuss traceability and they do not discuss in detail transaction verifications, congestion, regulatory compliance, and necessary standards.

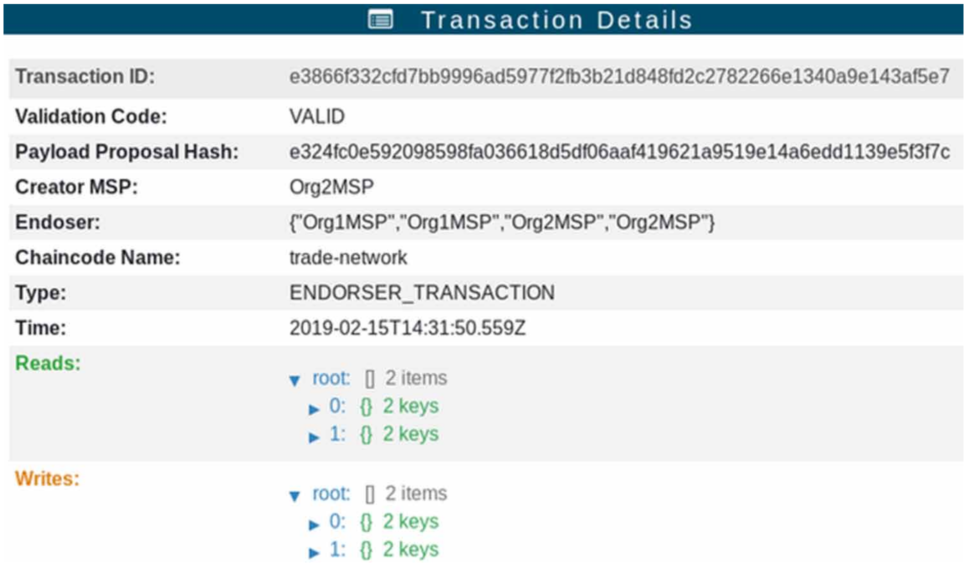
Figure 19. Block Details



Block Details

Channel name:	mychannel
Block Number	19
Created at	2019-02-15T14:31:50.559Z
Number of Transactions	1
Block Hash	c5601c309d2ad2fe5c6a3a396b05baed3f8e93712f68fe5124e8c3ed1bf39f0f
Data Hash	9c7b0faae5d33ab808ac1e6110e17b0365f880d07227a73b405cab688dd6d468
Prehash	7c44b2fc000ac8abe5795ed767ec17632ebb77a7629673b6f572bcafc6044597

Figure 20. Transaction Details



Transaction Details

Transaction ID:	e3866f332cfd7bb9996ad5977f2fb3b21d848fd2c2782266e1340a9e143af5e7
Validation Code:	VALID
Payload Proposal Hash:	e324fc0e592098598fa036618d5df06aaf419621a9519e14a6edd1139e5f3f7c
Creator MSP:	Org2MSP
Endoser:	{"Org1MSP","Org1MSP","Org2MSP","Org2MSP"}
Chaincode Name:	trade-network
Type:	ENDORSER_TRANSACTION
Time:	2019-02-15T14:31:50.559Z
Reads:	▼ root: [] 2 items ▶ 0: {} 2 keys ▶ 1: {} 2 keys
Writes:	▼ root: [] 2 items ▶ 0: {} 2 keys ▶ 1: {} 2 keys

This chapter implements a supply chain use case using Hyperledger Fabric and illustrates the farm-to-fork process relating to the food supply industry, traceability, item recall, transaction congestion, regulatory compliance, necessary standards, etc. Moreover, authors are highlighting necessary improvements to realise complex supply chains with connecting regulators placed in various countries, integrating with IoT systems, managing redundant data, and managing a large volume of transactions.

It is noticeable that current blockchain architectures and key-value data stores do not directly support end-to-end traceability of assets, item recall, and lacking necessary standards. Developers should, therefore, involve and develop necessary workflows. The authors propose a private channel concept to position a regulator in a blockchain network, minimizing data redundancy and ensuring auditing. However, continuous improvements are needed to manage block congestion, transaction verification, and compensation management in real-time complex blockchain applications.

ACKNOWLEDGMENT

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) and the Department of Agriculture, Food and the Marine on behalf of the Government of Ireland under Grant Number SFI 16/RC/3835 (VistaMilk), and also by a research grant from SFI under Grant Number SFI 12/RC/2289_P2 (Insight), with both grants co-funded by the European Regional Development Fund.

REFERENCES

Benhamouda, F., Halevi, S., & Halevi, T. (2018). Supporting Private Data on Hyperledger Fabric with Secure Multiparty Computation. In *2018 IEEE International Conference on Cloud Engineering (IC2E).* Orlando, FL: IEEE. 10.1109/IC2E.2018.00069

Bitcoin. (2020). *Bitcoin developer*. Retrieved from Learn Bitcoin and start building Bitcoin-based applications: https://developer.bitcoin.org/

Composer. (2020, June 23). *Typical Hyperledger Composer Solution Architecture*. Retrieved from Hyperledger Composer: https://hyperledger.github.io/composer/v0.19/introduction/solution-architecture

Consensys. (2020). *Blockchain Use Cases-Blockchain in Supply Chain Management*. Retrieved from https://consensys.net/blockchain-use-cases/supply-chain-management/

Cooper, M. C., Lambert, D. M., & Pagh, J. D. (1997). Supply Chain Management: More Than a New Name for Logistics. *International Journal of Logistics Management*, *8*(1), 1–14. doi:10.1108/09574099710805556

Davor, D., & Domagoj, S. (2018). Blockchain Applications in Supply Chain. *SMART Supply Network*, 21-46.

Esteban Koberg, A. L. (2018). A systematic review of sustainable supply chain management in global supply chains. *Journal of Cleaner Production*, 1084–1098.

Ethereum. (2020, June 23). *Ethereum is a global, open-source platform for decentralized applications.* Retrieved from ethereum.org: https://ethereum.org/

Gao, Z., Xu, L., Chen, L., Zhao, X., Lu, Y., & Shi, W. (2018). CoC: A Unified Distributed Ledger Based Supply Chain Management System. *Journal of Computer Science and Technology*, 237–248.

Hyperledger. (2020a, June 23). *Hyperledger Fabric*. Retrieved from Hyperledger: https://www.hyperledger.org/use/fabric

Hyperledger. (2020b, June 23). *Private data*. Retrieved from Hyperledger Fabric: https://hyperledger-fabric.readthedocs.io/en/release-2.0/private-data/private-data.html

Kenton, W. (2020, June 24). *Hyperledger Fabric*. Retrieved from Investopedia: https://www.investopedia.com/terms/h/hyperledger-fabric.asp

Ko, K., Lee, C., Jeong, T., & Hong, J. W.-K. (2018). Design of RPC-based Blockchain Monitoring Agent. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. Jeju, South Korea: IEEE. 10.1109/ICTC.2018.8539456

Kshetri, N. (2018). Blockchain's roles in meeting key supply chain management objectives. *International Journal of Information Management*, *39*, 80–89. doi:10.1016/j.ijinfomgt.2017.12.005

Li, W., Sforzin, A., Fedorov, S., & Karame, G. (2017). Towards Scalable and Private Industrial Blockchains. In *BCC '17: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* (pp. 9-14). ACM. 10.1145/3055518.3055531

Litecoin. (2020, June 23). *The Cryptocurrency For Payments: Based on Blockchain Technology*. Retrieved from https://litecoin.org/

Martin, S. (2017). *Blockchain use cases for food traceability and control. Swedish county councils and regions: Kairos Future*. Retrieved from Kairos Future: https://tinyurl.com/y7hh6nup

Meyvaert, E. (2020, June 22). *Siemens Mindsphere and SettleMint's distributed middleware: the perfect match!* Retrieved from Medium: https://updates.settlemint.io/siemens-mindsphere-and-settlemints-distributed-middleware-the-perfect-match-b1ce77335c09

Modi, R. (2018). *Solidity Programming Essentials*. Packt Publishing.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* www.bitcoin.org

Perboli, G., Musso, S., & Rosano, M. (2018). Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 62018–62028. doi:10.1109/ACCESS.2018.2875782

Quorum. (2020, June 23). *The proven blockchain solution for business*. Retrieved from Evolve with Quorum: https://www.goquorum.com/

Siemens. (2019). *Trusted Traceability - Blockchain and the Internet of Things.* uremberg, Germany: Siemens AG 2019. Retrieved from https://assets.new.siemens.com/siemens/assets/api/uuid:de496ba4-0081-48f5-965b-4963879b2d43/version:1557493248/vrfb-b10033-00-7600sbblockchainfb-144.pdf

Thummavet, P. (2020, June 22). *Demystifying Hyperledger Fabric: Fabric Architecture*. Retrieved from Medium: https://tinyurl.com/y9yn33k5

Westerkamp, M., Victor, F., & Küpper, A. (2018). Blockchain-Based Supply Chain Traceability: Token Recipes Model Manufacturing Processes. In *IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom).* Halifax, NS, Canada: IEEE Xplore. 10.1109/Cybermatics_2018.2018.00267

Yiannas, F. (2018). A New Era of Food Transparency Powered by Blockchain. *Innovations: Technology, Governance, Globalization*, *12*(1-2), 46–56. doi:10.1162/inov_a_00266

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *2017 IEEE International Congress on Big Data (BigData Congress).* Honolulu, HI: IEEE. 10.1109/BigDataCongress.2017.85

ADDITIONAL READING

Ahamed, N. N. K. P, S. P. Anandaraj and V. R. Sea Food Supply Chain Management Using Blockchain, *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 473-476.

Du, M., Chen, Q., Xiao, J., Yang, H., & Ma, X. (2020). Supply Chain Finance Innovation Using Blockchain, in IEEE Transactions on Engineering Management, doi:10.1109/TEM.2020.2971858

Hastig, G. M., & Sodhi, M. S. (2019). Blockchain for Supply Chain Traceability: Business Requirements and Critical Success Factors. *Production and Operations Management*, 935–954.

Jabbar, A., & Dani, S. (2020). Investigating the link between transaction and computational costs in a blockchain environment. *International Journal of Production Research*, *58*(11), 3423–3436. doi:10.1080/00207543.2020.1754487

Li, Z., Guo, H., & Ali Vatankhah Barenji, W. M. (2020). A sustainable production capability evaluation mechanism based on blockchain, LSTM, analytic hierarchy process for supply chain network. *International Journal of Production Research*, *58*(24), 1–21. doi:10.1080/00207543.2020.1740342

Subramanian, N, Chaudhuri, A, & Kayikci, Y. (2020). Blockchain Applications in Food Supply Chain. *Blockchain and Supply Chain Logistics*, 21-29. Springer.

Subramanian, N, Chaudhuri, A, & Kayikci, Y. (2020). Blockchain Applications in Health Care Supply Chain. *Blockchain and Supply Chain Logistics*, 31-38. Springer.

KEY TERMS AND DEFINITIONS

Digital Twin: Digital twin refers to a digital replica of potential and actual physical assets, processes, people, places, systems, and devices that can be used for various purposes.

Farm-to-Fork: The stage involves the growing, processing, and consumption of food – the entire food cycle, from supplier to the customer table.

IoT (Internet of Things): The Internet of things is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Miner: Miners validate new blockchain transactions and record them on the blockchain. Miners compete to solve a difficult mathematical problem based on a cryptographic hash algorithm.

Peer-to-Peer (P2P): P2P computing is a distributed application architecture that partitions tasks between peers. Peers are equally privileged in the application.

REST API: Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. The application programming interface (API) defines interface functions.

RFID: Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects.

Smart Contract: A smart contract is a vital component of a Blockchain; it is a self-enforcing agreement embedded in computer code managed by a blockchain. Agreement comes in force, automatically, when certain pre-agreed conditions are met.

ENDNOTES

[1] https://www.cdc.gov/salmonella/2009/peanut-butter-2008-2009.html
[2] https://www.cdc.gov/ecoli/2016/o121-06-16/index.html
[3] https://www.cdc.gov/ecoli/2020/o103h2-02-20/index.html
[4] https://www.arcweb.com/blog/supply-chain-transparency-farm-table
[5] https://www.hyperledger.org/projects/fabric
[6] https://github.com/hyperledger/composer/wiki/Composer-REST-Server
[7] https://swagger.io/
[8] https://hyperledger-fabric.readthedocs.io/en/release/commands/peerchannel.htm