Towards a Better Blockchainification of Supply Chain Applications

Mazumdar, Somnath

Document Version
Final published version

Published in:
Systems and Soft Computing

DOI:
10.1016/j.sasc.2022.200043

Publication date:
2022

4 (2022) 200043

Contents lists available at ScienceDirect

Systems and Soft Computing

journal homepage: www.elsevier.com/locate/soft-computing-letters

Towards a better blockchainification of supply chain applications

Somnath Mazumdar

Department of Digitalization, Copenhagen Business School, Solbjerg Plads 3, Frederiksberg, 2000, Denmark

ARTICLE INFO

Keywords:
Blockchain
Data
IoT
Framework
Machine learning
Software
Supply chain

ABSTRACT

A supply chain ecosystem is a collection of complex asynchronous events. Blockchain has already found commercial applications in the SC domain, particularly in product tracing and verification. However, there is a lack of uniformity in these approaches. Application-generated data cannot be accessed across the supply chain ecosystem, resulting in data silos. Data silos reduce the opportunity for supply chain process optimizations. This paper does not propose any supply chain solution but a generic framework primarily aimed at reducing the communication gaps among the stakeholders and application developer(s) to build quality solutions. The ideal readers are who want to blockchanify their existing supply chain applications. The proposed framework can add real value to the organization by developing effective SC solutions satisfying application requirements. The framework consists of four stages. In the first stage, it extracts the application requirements and then maps on blockchain following an asynchronous mode of communication among the stakeholders and application developer(s). Next, it discusses how it can combine technologies to achieve the requirements stated in the first stage. Later, it discusses how to perform effective data management. Finally, it proposes a four-stage software build method that can lead to an efficient SC solution. The primary aim of this framework is to reduce communication gaps during solution development and ensure smooth operational data movement across the SC ecosystem, thanks to blockchain. The software development process also embeds eight essential features for a quality solution. The paper is concluded by discussing the technical challenges.

1. Introduction

Supply chain (SC) applications are complex and consist of multiple asynchronous processes and components [1]. Human stakeholders control the core activities of the SC ecosystem. Traditional SC applications aim to help human stakeholders to trace the SC processes and components efficiently and smoothly. However, traditional SC applications (also known as legacy applications) are centralized and very restrictive while delegating data to the outside world. Cloud computing has distributed these centralized SC applications in recent years. However, such restrictions are still in place. Restrictions reduce the possibility of harnessing the insights from the SC-generated data. Gartner, in its report,[1] mentioned that more than 60% industry respondents think that technology is a source of competitive advantage in SC. It is also projected that the blockchain-based SC market size is expected to grow at a compound annual growth rate of more than 50% during the years 2022 and 2031 [2]. In particular, blockchain use cases specific to SC, i.e., product trace and verification, cover around 31% and 28% of total blockchain-based solutions, respectively [3]. Blockchain started to become famous for its cryptocurrencies. However, it became popular in other commercial applications due to its data-focused solid security features. Blockchain-based SC solutions are becoming popular among the end-users. Industries are also started to promote transparency in the SC domain to achieve sustainable manufacturing habits. Existing SC-focused blockchain applications can primarily be categorized into product traceability and product verification [4].

Moving from a traditional SC application to a blockchain-based application is not trivial. Blockchain is a distributed append-only data store (or ledger) that combines hashing and digital signature-like cryptography to resist data tampering. Blockchain is not attack-proof but attack-resilient. Blockchain-based SC applications store immutable product history related to multiple important events. Such immutable traces can increase the transparency of SC-related activities. Some blockchain-based SC applications also integrate internet-of-things (IoTs) to collect real-time data to support specific events. For instance, a GPS device can track all covered locations for a product delivery. It is worth noting that the practical application of IoT devices can also facilitate sustainable logistical operations [5]. Successful blockchain-IoT integration can manage and securely store relevant event data for process optimization.

E-mail address: sma.digi@cbs.dk.

[1] https://www.gartner.com/smarterwithgartner/gartner-predicts-the-future-of-supply-chain-technology.

[2] Press Releases, Jan-23-2020, Gartner.

https://doi.org/10.1016/j.sasc.2022.200043
Received 1 August 2022; Received in revised form 8 September 2022; Accepted 9 September 2022
Available online 17 September 2022

Blockchain can create value for the SC application, but the network structure of the SC should be mapped onto the blockchain platform. Successful blockchain use cases for SC require a different approach.[2] Selecting a blockchain platform for SC application is complex and can be translated as an instance of a multi-criteria decision-making problem [6]. Besides, blockchain is not mature enough and requires extensive feasibility studies before implementation [7]. After selecting a blockchain platform, important events related to SC processes and components need to be selected. For instance, for manufacturing wooden furniture one event can be *logging*. For logging events, the relevant data can include tree name, logging time, logging location, logged by, sustainability standards related metadata and others. Later, the collected data is stored in the cloud, while the important product metadata (hashed value of cloud data) is stored in the blockchain. Deciding which events are important and how much data in blockchain should be stored is difficult. A low level of technical understanding can reduce the efficiency of the whole SC application.

One generic framework have been proposed in Section 3, which consists of four stages. The first stage decomposes important communication among stakeholders and application developer(s) for a better understanding of the problem while reducing the communication gap (refer to Section 3.1). Next, it recommends how other technologies (such IoT, and machine learning (ML)) can be integrated with blockchain (mentioned in Section 3.2). Later, it also shows how data should be managed for the application (refer to Section 3.3). The fourth stage proposes a four-step software build method that can help to execute the application development process (mentioned in Section 3.4) while focusing on eight primary software development features (mentioned in Section 2.1). Finally, successful implementation of this framework will offer three advantages. They are (i) a quality SC application, (ii) reduction of communication gap among stakeholders during application development, and (iii) the reduction of data silos[3] in the SC ecosystem thanks to the blockchain.

2. Application modelling

A successful "blockchainification" of an SC application is the outcome of a good match between the application's demands and the blockchain's features. Although blockchain is not mature enough, but it is developed on a mature technology stack, primarily peer-to-peer infrastructure, cryptography protocols, and distributed storage with associated access control protocols. Communication gaps can be created while transferring knowledge among the stakeholders and application developer(s) during the application development phase. The situation can further be complicated during data collection of relevant events.

The proposed framework can be applied to *reduce the communication gap* between stakeholders and the application developer(s). Here, the framework can cover all important SC events ranging from the initial stage of the product manufacturing to the end of the product life cycle including the support of the circular economy [8]. This framework is generic and can support any SC application. It is worth noting that the framework can support any SC application feature by listing related events. Data selection is essential for application performance. During development, data selection can be categorized into *must-have* and *nice-to-have* features. Must-have features aim to support the functional suitability of the application, while nice-to-have features aim to support usability. SC stakeholders and application developer(s) are both primarily responsible for the high acceptance of the SC application.

Current SC applications must offer faster delivery of quality products while adhering to social compliance (such as sustainability). Sustainability can be implemented by adequately collecting related event data. The circular economy is also a collection of multiple events. Relevant events must be identified during the planning stage, and the stakeholders must decide how these event data can be collected. Later, the application developer(s) should decide how to process and store such data. The selected events should support the proper implementation of functional suitability and usability.

[3] Data silo stops the free communication with other SC processes and components.

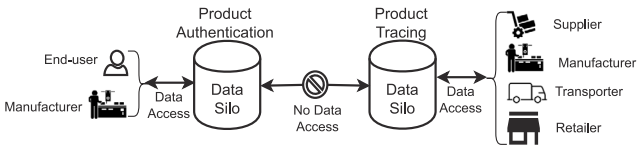

Fig. 1. A representation of data silos generated during product verification and product tracing.

2.1. Application mapping

Eight software qualities, i.e., *compatibility*, *functional suitability*, *maintainability*, *performance efficiency*, *portability*, *reliability*, *security*, and *usability* must be maintained for a successful application mapping on the blockchain [9]. Compatibility is one of the most important features for better integration with existing legacy SC applications and other technology. Functional suitability includes functional completeness, appropriateness, and correctness of the SC application. Software should be easy to maintain post-deployment and should be easy to port as well. Better performance should include higher network throughput. It is hard to achieve a lower energy cost of running a blockchain application. The application should be reliable and ideally should recover from code failures. Application security needs to be maintained with proper access rights. Finally, usability refers to the high acceptance of the build application among all users. Usability also represents appropriateness, user interface quality, and feature accessibility.

2.2. Communication paradigm

Most SC applications can be categorized into product verification and product tracing. Different solutions for different processes keep related SC event data in silos (refer to Fig. 1). From Fig. 1, we can see that data silos always support vertical communication, making it hard for other SC processes to work with. It results in no use of such important data across the SC ecosystem. For seamless data transfer, the data movement should be unified. Multiple SC processes communicate in asynchronous mode among multiple stakeholders. To counter such silos, blockchain can delegate data access based on push and pull communication protocol. The blockchain platform primarily acts as secure data storage and allows data access based on the credentials. If IoTs are embedded into the system, they will work as data collectors. Some users may have read-write access, while the rest will have read-only access to the blockchain. Blockchain data delegation can be done based on user access rights. It should be decide during implementation by stakeholders. The append-only ledger of the blockchain holds all event traces. Such traces are essential to implement all required features, including sustainability and circular economy. It is worth noting that only users who enter data into the blockchain are primarily responsible for data authenticity. In a private blockchain network, an admin conducts a user verification process before allowing users to use the blockchain network. Enterprise-grade blockchain platform should be selected which uses a traditional crash fault-tolerant consensus protocol such as Hyperledger Fabric (HLF) that does not need mining [10]. Non-mining private blockchain not only employs a less power-hungry consensus protocol but also offers secure communication channels. Such channels can be implemented based on the access rights of the stakeholders. Overall, for implementing SC applications, HLF can achieve a higher success rate than public blockchains such as Ethereum [11].

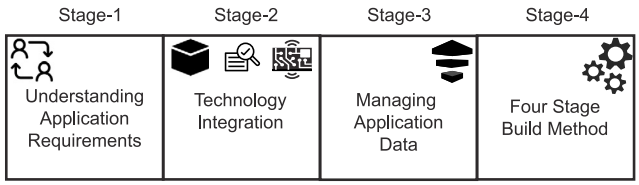

Fig. 2. Proposed generic framework with four stages.

2.3. Network planning

Setting up a blockchain network is complex and consists of four sequential stages. They are (i) blockchain *network type selection* decide whether the network will be a public, private, or consortium-based network; (ii) *applied consensus protocol* is crucial to the network throughput and energy cost. A traditional and popular consensus protocol for public blockchain platforms is proof-of-work which is very power-hungry and must be avoided. However, there exist a few less power-hungry consensus protocols which can reduce blockchain running cost [12]; (iii) *setting access rights* based on the user types, and finally, (iv) *node creation* refers to the users (including the admin) using the blockchain-based SC application.

3. Proposed framework and its four stages

Primarily, the framework consists of two components *application* and *technology*. The framework has four stages (refer to Fig. 2). The first stage starts by understanding the important application requirements following a feedback-oriented communication mode among the stakeholders and the application developer(s). The next stage discusses how other technologies (such IoT, and ML) can be integrated with blockchain for seamless data flow. In these cases, IoTs will collect data. Blockchain will secure on-chain product metadata and delegate the data per access rights among the stakeholders. Next, ML can work on collected data (off-chain storage) to give insights for optimizing the SC processes. Third stage concerns efficient data management. Finally, it applies a four-stage method to build a quality blockchain-based SC application. The four-stage build method embeds eight important software features during development.

3.1. Stage-1: Requirement understanding

Fig. 3 presents the first stage of the proposed framework. It further divides the application into stakeholders and core application parts. The stakeholders of SC represent the primary decision-makers. They primarily focus on the application's functional stability and usability features. The core application part represents the important SC processes and components which should be digitized or need further enhancements. Only stakeholders can decide whether to accept the new modifications made to the application or not. Stakeholders can decide which features of the SC application should fall into must-have and nice-to-have category. It is also very much required to know by stakeholders how the relevant data can be collected to support these two categories. The technology domain can be further divided into application developer(s) who map the core technology (such as blockchain) to the required application features. During the initial stage of a technology life cycle, it does not offer stable and complete support to any application. Thus many application integrations fail. However, technology understanding is another reason for failure during these initial times. The number of matches between the required and offered features increases as time passes. Some frequent HLF updates are user interface improvement, bug fixes, securing data flow, extending access control, transaction ordering, complex network creation support, and more smart contract features. However, the development stage can be long. Developer(s) can either extend the current blockchain platform by adding other technologies or add third-party libraries (or packages) to support the required features.

We can see that three communication paradigm such as 'publish–subscribe', 'require-satisfy', 'understood by' and 'translated to' exists in six sequential stages (refer to right-side of Fig. 3). Some of these stages can also run in parallel. First, the primary requirements of the application (must-have features) should be understood by stakeholders. Next, these features are conveyed to the application developer(s). Communication gaps can be created when the requirements are translated to the application developer(s). Communication gaps can also creep in regarding how the stakeholders have accurately understood the complex SC processes (including the necessary components). It is mainly related to the knowledge gap among stakeholders. Apart from that, a gap can also be created on the application developer(s) side. It means the gap can depend on the technical understanding of the application developer(s) and the technology's maturity level. Technology (such as blockchain) publishes its features while the application subscribes to them. Application developer(s) can satisfy stakeholders by accurately mapping the requirements with the technical offerings. There will be no perfect match between the offered features and the required application support. Overall, there are two primary stages where the gap can occur: *How successfully application requirements are translated to application developer(s) by stakeholders?* and *how much technology and developer's knowledge base is matured?*

3.2. Stage-2: Technology integration

It is very hard for large enterprises to move away from legacy systems as they have already invested many resources. Thus, the blockchain-based SC application should not aim to replace legacy systems but to complement them. In other words, blockchain-based applications should be developed as a layer that can work on top of legacy systems. From Fig. 4, we can see that data from the legacy systems can be passed to the blockchain (via read-only mode). Such integration can be done vertically. Data can be passed in one direction for vertical integration, if there is a strict data access rule exist. Existing blockchain-based SC applications also use IoT. From the bottom part of Fig. 4, we can see that IoT and blockchain are integrated vertically (similar to legacy and blockchain integration). At the same time, IoTs are connected to legacy systems horizontally. Such hybrid integration lets the data flow per requirements and reduces unnecessary bottlenecks. More complex hybrid integration can be where a legacy system is integrated with the blockchain vertically, while IoTs are integrated horizontally into the legacy system and vertically into the blockchain. Similarly, ML can be combined with blockchain horizontally while following vertical integration with the rest. ML can bring two advantages: (i) it can extract meaningful insights from the SC generated data and (ii) it also can add security to the generated data sets. One of the most popular ML models in SC is artificial neural networks (ANNs) which have widely been used in the SC domain to extract insights and patterns. ANNs mainly used for sales and production forecasting, marketing, pricing and customer segmentation, and supplier selection. Apart from that, natural language processing is another promising model for SC applications. ML models can optimize distribution and transportation activities as well [13]. Apart from ML, meta-heuristics-based optimization methods can also be applied to optimize recycling [14] and logistical operations [15]. ML is not new to the application security domain and has already been employed for malware, anomaly, and intrusion detection [16].

3.3. Stage-3: Managing application data

The quality and amount of data stored in the blockchain are crucial for overall solution quality. Keeping and placing data in the cloud is not a trivial task. It becomes more challenging when it gets bigger and distributed across locations. Unfortunately, no well-accepted

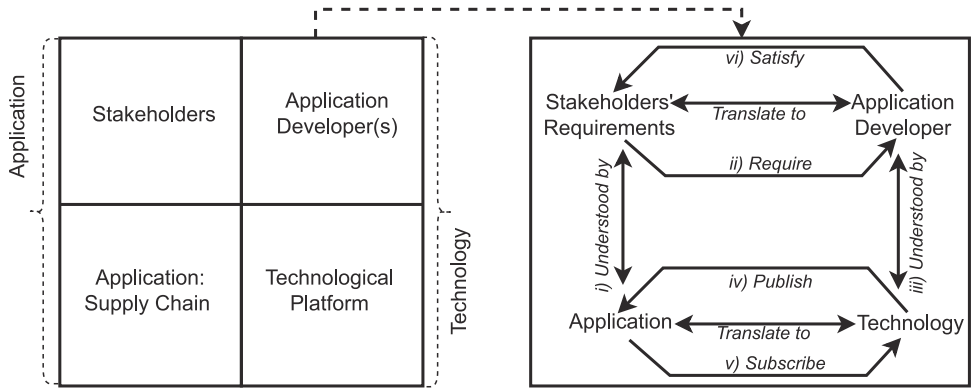

Fig. 3. Two primary components (left-side) and three communication paradigm in six stages (right-side).

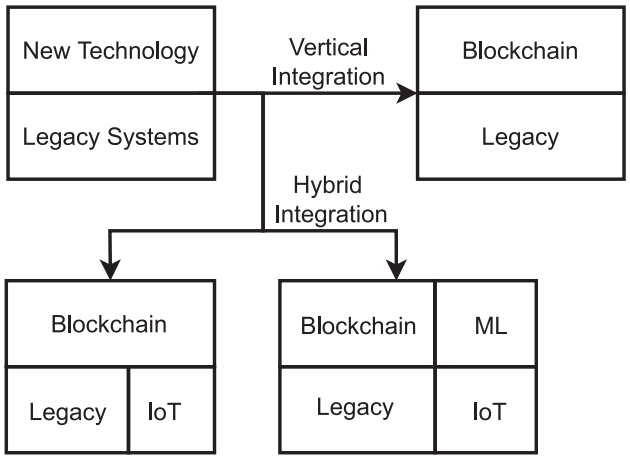

Fig. 4. Stacking other technologies with blockchain.

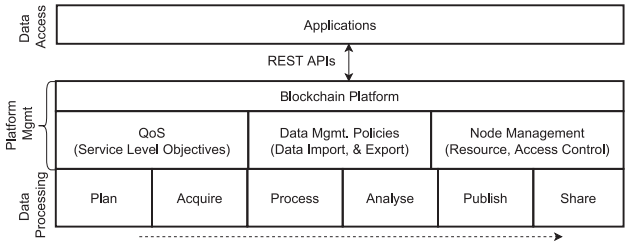

Fig. 5. Managing application data in three stages.

data access optimization methods exist for cloud [17]. Having feature-rich data stored in the cloud helps to harness meaningful insights by ML models. As already mentioned, stakeholders must be aware of how to collect event data that supports must-have features. Thus, the stakeholders and the application developer(s) should consider data collection and processing seriously. Fig. 5 presents three stages to manage data and delegate it to users or ML models. The first stage is divided into six sequential sub-stages for data processing. They are (i) *plan:* how to collect data and also related to whether IoT devices (and QR codes, barcodes) are there or not; (ii) *acquiring* phase collects relevant data, the frequency of data collection is important as it can impact the data volume; (iii) collected data need to be *processed* applying the data policies and requirements set by stakeholders; (iv) only required part should be kept for further *analysis*, (v) on-chain and off-chain data storage strategies should be decided (among the stakeholders and application developer(s)) during data *publication* into the blockchain. Finally, (vi) the data is stored, it can only be *shared* among network users based on access rights. Platform management consists of data management policies (ideally storage management), computing resource management, and Quality of Service (QoS). QoS can also include service-level objectives to improve the application experience and network productivity. Finally, the data can be shared using the REpresentational State Transfer APIs among the users.

3.3.1. Further considerations

The framework primarily relies on the existing components that come with a standard version of the blockchain solution stack. Primarily for blockchain, four features can further be looked at. They are briefly discussed below:

1. **Storage.** Blockchain mostly handles semi-structured data via JavaScript Object Notation. It comes with a NoSQL database (such as CouchDB) or key–value store (such as LevelDB). However, such storage can be replaced by other storage applications (such as ledger databases), and data management policies can be updated accordingly.
2. **Computing technologies** such as multi-party computation and serverless computing can further be added to make blockchain more scalable and secure. Using newer computing technology can reduce the energy consumption of running a blockchain network. The node management policies (including the QoS) should also be updated.
3. **Security features** such as zero-knowledge proof protocol and enhanced user wallet security can be added to node management to improve network security.
4. **Privacy** is very much important in today's world. Privacy of user data and SC process-related data should be preserved following privacy reserving rules. Blockchain-based solutions offer hashing and digital signatures to secure data over industry-grade cloud security features.

3.4. Stage-4: Four stage build method

Finally, a four-stage software development method can be followed to build a production-grade solution from a prototype. A prototype with must-have features should be built first. After the successful prototyping, it can move to the pilot stage (focusing on nice-to-have features) and later to the production (including all eight software features) following four stages (described below).

1. In **plan stage**, the stakeholders, with the help of application developer(s), must define the use cases focusing on must-have features to complete the prototype. A good understanding of SC processes and components is required at this stage to select a suitable blockchain platform. It also should be considered how much blockchain *alone* can support these critical features. Application compatibility should be considered here.

2. In **analyze stage**, the application developer(s) must understand the SC processes via multiple communications or feedback sessions. Here, good technical know-how (may also include existing literature study) plays a vital role in knowing the application domain and selected technology's ability. After proper understanding, a system model should be built for further implementation. It is worth noting that blockchain can either be used as a stand-alone application or can work on top of existing legacy systems. Other supporting technologies (such as IoT and ML) can also be used while building the system model. *How much extra security can the blockchain offer?* and *how much the application will be reliable?* can be considered here. As can be seen, there are many situations where communication gaps can be created. Some relevant questions can be: *Which features are essential? How can we collect the required data? How will the collected data be stored? What are the access rights?*
3. In **build and execute stage**, the system model is translated into the application and tested with relevant use cases. The available software quality and third-party software (library, packages) support are crucial during the build process. If it is being built on open-source tools, then pre-build and post-build support from the community is also very much essential for the success of the prototype. After successful execution, the performance should be benchmarked, and the relevant QoS can be verified. If required, the service level objectives can also be checked.
4. In **verify and monitor stage**, the must-have features must be checked and monitored whether they satisfy functional suitability and usability. Stakeholders should also review the application's maintenance, including its portability.

3.5. Technical challenges

Four primary challenges exist to "blockchainify" the SC application. These include standardization, network scaling, energy cost, and network type selection. Blockchain is a composite platform, and developing a production-grade application on a non-mature technology is complex [7]. Apart from that, blockchain has not yet been standardized [18]. Scalability is another problem dependent on the data storage strategies (on-and-off-chain). If the data block sizes and numbers are getting large, soon, the network will reduce its throughput. An informed decision should be made while deciding which data should be stored in and outside the blockchain. However, multiple solutions are also being proposed to improve the application quality [19]. Apart from that, interfacing two popular platforms (such as HLF and Ethereum) is also not standardized. The use of non-standardized interfaces or protocols can lead to data leaks. Thus, precautions should be taken while integrating multiple blockchains (including legacy systems). Traditionally, public blockchain networks consist of many nodes and employ a power-hungry consensus protocol. A good part of energy cost can be saved while using simpler consensus protocols [20]. Finally, selecting the network type and answering *how the blockchain network will be managed?* and *who will manage?* is tricky. Primarily, the commercial setup prefers a private blockchain. Here, a primary stakeholder hosts the network and becomes the admin of the whole network. In the future, such a private network in the SC domain may turn into a federated network with proper interface development. Overall, for implementing SC applications, it has been seen that private blockchain networks can achieve a higher success rate than public blockchain [11].

Apart from blockchain, IoTs also suffer from security challenges [21]. It has been seen that a large number of ML models (such as clustering [22], Naive Bayes decision tree, multilayer perceptron, support vector machine [23]) are vulnerable to security attacks due to the existence of malicious data in the training data sets, which lead to decreased model performance. It is also interesting to note that popular ML models are not attack-proof.

4. Conclusion

There is a general trend of manufacturers investing in advertisement and pricing strategies, which can influence customers' purchase behaviours. There are also some exciting models to handle such scenarios [24]. In today's competitive market, it is challenging for a product manufacturer to document their products, including compliance with standards. However, product manufacturers are addressing it for documentation and marketing purposes with the help of blockchain. With other technologies (such as IoT and ML), blockchain can optimize the SC processes and improve overall SC applications.

This paper proposes a generic framework to help readers to develop a quality SC application by dividing the problem into four stages. The first stage aims to extract the application requirements, while the next stage selects technology based on requirements. The third stage focuses on managing the application data for better access. Finally, a four-stage software build method is proposed to improve overall solution quality. Blockchain helps to extend data security and smooth data flow across SC processes so that data silos can be reduced. Although the framework is generic, obtaining a high-level abstraction of an SC application is not easy. The framework is not complete and lacks the discussion on (i) the data access rights and (ii) sustainability support. For future work, it is aimed to implement it in the apparel industry, covering both the limitations.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] M.C. Cooper, D.M. Lambert, J.D. Pagh, Supply chain management: more than a new name for logistics, Int. J. Logist. Manage. 8 (1) (1997) 1–14.
[2] Kenneth Research, Blockchain in supply chain market, Tech. rep., Kenneth Research, 2022.
[3] D. Insights, Deloitte's 2021 global blockchain survey: A new age of digital assets, 2021.
[4] P. Gonczol, P. Katsikouli, L. Herskind, N. Dragoni, Blockchain implementations and use cases for supply chains-a survey, IEEE Access 8 (2020) 11856–11871.
[5] P. Singh, Z. Elmi, V.K. Meriga, J. Pasha, M.A. Dulebenets, Internet of things for sustainable railway transportation: Past, present, and future, Cleaner Logist. Supply Chain 4 (2022) 100065.
[6] S. Farshidi, S. Jansen, S. España, J. Verkleij, Decision support for blockchain platform selection: Three industry case studies, IEEE Trans. Eng. Manage. 67 (4) (2020) 1109–1128.
[7] H. Wang, K. Chen, D. Xu, A maturity model for blockchain adoption, Financial Innov. 2 (1) (2016) 1–5.
[8] A. Murray, K. Skene, K. Haynes, The circular economy: an interdisciplinary exploration of the concept and application in a global context, J. Bus. Ethics (2017) 369–380.
[9] ISO, Systems and software engineering: Systems and software quality requirements and evaluation (SQuaRE): System and software quality models, Int. Organ. Stand. (2011) 2910.
[10] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, et al., Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, 2018, pp. 1–15.
[11] N. Vadgama, P. Tasca, An analysis of blockchain adoption in supply chains between 2010 and 2020, Front. Blockchain (2021) 1–8.
[12] L.M. Bach, B. Mihaljevic, M. Zagar, Comparative analysis of blockchain consensus algorithms, in: 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, 2018, pp. 1545–1550.
[13] R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, M. Fischl, Artificial intelligence in supply chain management: A systematic literature review, J. Bus. Res. 122 (2021) 502–517.

[14] A.M. Fathollahi-Fard, M.A. Dulebenets, M. Hajiaghaei-Keshteli, R. Tavakkoli-Moghaddam, M. Safaeian, H. Mirzahosseinian, Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty, Adv. Eng. Inform. 50 (2021) 101418.
[15] M.A. Dulebenets, A comprehensive evaluation of weak and strong mutation mechanisms in evolutionary algorithms for truck scheduling at cross-docking terminals, IEEE Access 6 (2018) 65635–65650.
[16] W. Wang, M. Zhu, X. Zeng, X. Ye, Y. Sheng, Malware traffic classification using convolutional neural network for representation learning, in: International Conference on Information Networking, 2017, pp. 712–717.
[17] S. Mazumdar, D. Seybold, K. Kritikos, Y. Verginadis, A survey on data storage and placement methodologies for cloud-big data ecosystem, J. Big Data (2019) 1–37.
[18] N. Drljevic, D.A. Aranda, V. Stantchev, Perspectives on risks and standards that affect the requirements engineering of blockchain technology, Comput. Stand. Interfaces 69 (2020) 1–7.
[19] J. Xie, F.R. Yu, T. Huang, R. Xie, J. Liu, Y. Liu, A survey on the scalability of blockchain systems, IEEE Netw. (2019) 166–173.
[20] E. Delliere, C. Grange, Understanding and measuring the ecological sustainability of the blockchain technology, in: Proceedings of the International Conference on Information Systems, 2018, pp. 1–8.
[21] Y. Yang, L. Wu, G. Yin, L. Li, H. Zhao, A survey on security and privacy issues in Internet-of-Things, IEEE Internet Things J. 4 (5) (2017) 1250–1258.
[22] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, F. Roli, Poisoning behavioral malware clustering, in: Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop, 2014, pp. 27–36.
[23] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, N.K. Jha, Systematic poisoning attacks on and defenses for machine learning in healthcare, IEEE J. Biomed. Health Inf. 19 (6) (2014) 1893–1905.
[24] M. Asghari, H. Afshari, S. Mirzapour Al-e hashem, A.M. Fathollahi-Fard, M.A. Dulebenets, Pricing and advertising decisions in a direct-sales closed-loop supply chain, Comput. Ind. Eng. 171 (2022) 108439.