Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

6th International Conference on Industry 4.0 and Smart Manufacturing

Enhancing transparency in buyer-driven commodity chains for complex products: a blockchain-based traceability framework demonstrated through an apparel supply chain simulation

Ritwik Takkar[a,b], Ken Birman[a,b], H. Oliver Gao[a,c,*]

[a] *Department of Systems Engineering, Cornell University, Ithaca, NY 14853, USA*
[b] *Department of Computer Science, Cornell University, Ithaca, NY 14853, USA*
[c] *School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA*

Abstract

Buyer-driven commodity chains are characterized by commercial relationships between buyers and sellers that may obscure accountability due to complexity, thereby undermining sustainability efforts. Conventional methods to trace production, including ineffective human-led audits, risk reorienting global corporate governance towards the interests of private business and away from social benefit by limiting the role of objective data in the process. This study examines the relevant features of private, permissioned blockchain towards harnessing the transparency challenge by demonstrating the efficacy of our proposed framework against a simulation of a real-world multi-tier apparel supply chain. The simulation integrates a set of functional and operational requirements achieved through a combination of programmable smart contracts and underlying blockchain architecture. We then evaluate the framework both qualitatively and quantitatively before discussing the limitations of our work.

Keywords: Blockchain; Traceability; Supply Chain Management; Manufacturing

---

1. Introduction

Buyer-driven commodity chains (BDCCs), wherein large retailers delegate production to independent factories under specialized arrangements, underpin global trade [1]. The apparel industry is an example where chronic downward price pressure, high volatility, and varying regional norms tend to relegate these inherent multi-tiered networks to a "morass of commercial relationships" eventually compromising supply chain transparency [2]. The information asymmetry between production stakeholders extends to end-consumers, thereby comprising a "nontransparent" sup-

---

[*] Corresponding author. Tel.: +1 (607) 254-8998.
E-mail address: hg55@cornell.edu

ply chain [3]. Nontransparency has led to catastrophic events and counterfeiting issues, while the absence of robust traceability systems renders many product recalls ineffective, with apparel goods constituting a significant portion of often unaccountable recall notifications [4, 5]. The complexity and global spread of apparel supply chains often hinder buyers' ability to monitor sustainability risks and track all involved suppliers, leading to potential deviations from standard practices [6]. Accidents derived from supply chain opacity are not restricted to the apparel industry, with discernible transgressions present in pharmaceutical [7, 8] and food [9, 10] supply chains, among others.

Calls from regulatory bodies [11, 12] and consumer advocates [13] to share increasingly detailed supply chain provenance ushered retailers to invest in comparatively isolated, centralized data management platforms [14]. Systems like enterprise resource planning (ERP) software, typically controlled by one single organization within the entire chain, possess a unique set of challenges and limitations: most notably, single point failure [15]. Furthermore, participation in ERP software demands unwarranted confidence from potentially unwilling participants, thus reinforcing power imbalances. While measures to hold sellers accountable to forms of sustainability standards date back decades [16], continued infractions suggest that the ever-growing complexity of BDCCs has outpaced the ability of standard means to hold them accountable: human-led audits and inspections [17]. A 2016 report [18] consisting of findings from twenty-five interviews with ethical auditors, business executives, NGOs, and supplier firms in North America, the UK, and China, including factory visits, found that (conventional) audits are ineffective.

A surge in literature related to blockchain-related traceability solutions for the apparel supply chain indicates consensus among academia to address the traceability challenges in BDCCs, especially apparel supply chains. For instance, a 2021 paper [19] notes that the query "(TITLE-ABS-KEY (blockchain* AND (textile* OR garment* OR cloth* OR apparel*)))" outputted 32 search results on the Scopus scholarly database of which 10 were determined relevant, whereas the same query outputted 196 search results in August 2024. Of the 10 relevant works found, the authors noted that most of them described only theoretical frameworks and applications of blockchain in the apparel industry, lacking even simulation results. The same work also presents a blockchain-based traceability framework for an apparel supply chain. However, there is an implicit assumption that extant supply chain stakeholders will be able to program smart contracts and transact without outside intervention. Additionally, instrumentation protocols, e.g., the tools/method(s) through which upstream partners such as yarn manufacturers are expected to transact in the private blockchain network, are ambiguous. Another related work [20] introduces a Bitcoin-reliant solution to monitor and verify the state of information flow within business processes at runtime for documentation, accounting, or compensation. Here, runtime verification evaluates whether a process execution met the functional and nonfunctional objectives defined in a contract between process participants. Ultimately, unpredictable bitcoin transaction settlement times not only render this approach suitable for select instances wherein time is not a critical factor, but the functional reliance on a cryptocurrency adds further complexity.

We leverage prior works to establish a conceptual foundation for WEave: our novel blockchain-based traceability framework, encompassing the simulation herein, to demonstrate the applicability and efficacy of integrating blockchain into the solution set to harness the apparel supply chain traceability challenge. WEave is fine-tuned to capture the requirements of complex product assembly, i.e., products that undergo changes in their modular composition during the manufacturing process, in addition to the nuanced characteristics of BDCCs. The following sections outline suitable features of permissioned blockchain, articulate custom smart contract functionality addressing stated traceability requirements, and provide discussion, including simulation methodology and results, on key aspects of the proposed framework.

2. Key concepts

WEave aims to enable the mapping of complex assembly processes, dynamic adjustments of assets, and efficient auditability for apparel goods through three design objectives. First, to enable traceability per ISO 9000:2015 section 3.6.13 [21], requiring the ability to trace "the origin of materials and parts; the processing history; the distribution and location of the product or service after delivery." Next, to store traceability data in a distributed tamperproof log accessible to all production stakeholders. Finally, to allow real-time information sharing and collaboration among stakeholders, facilitating multi-party transaction verification and approval to enhance accountability and streamline processes across the network.

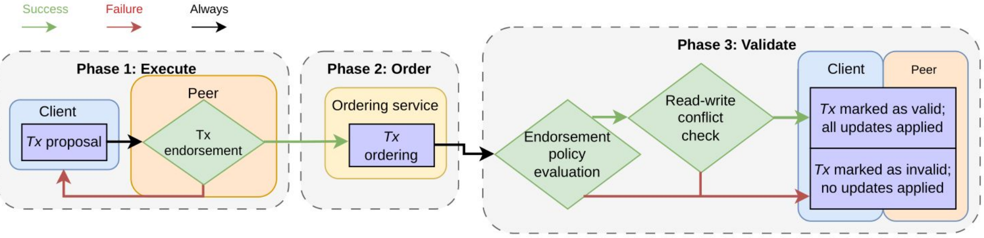


Fig. 1. Fabric's execute-order-validate transaction architecture.

WEave utilizes Hyperledger Fabric [22] (Fabric), an open-source, enterprise-grade, and modular distributed ledger platform, for core customizable permissioned blockchain services as follows. A ledger which consists of two components: the *blockchain*, an immutable append-only data structure containing past transactions, and the *state*, a database reflecting the most current key-value pairs. Smart contract files are packaged into *chaincode* which are executable programs that govern transactions and are deployed on *channels*, which are private subnets of communication between select network participants. The network participants interface as nodes, specifically, *clients*, *peers*, or *orderers*, each with a unique identity provided by a *membership service provider* (MSP). Peers host chaincodes and channel-specific ledgers, while a subset of peers known as *endorsers* validate proposed transactions. The ordering service, composed of orderers, organizes endorsed transactions in *total order* and creates blocks that are atomically broadcasted to peers within the channel.

Fabric offers a unique blend of features that address the limitations of both traditional databases, which are used for ERP software, and permissionless blockchains like Bitcoin [23] and Ethereum [24]. Its permissioned nature ensures precise governance and enhanced data privacy, crucial for sensitive supply chain information, while maintaining the distributed trust model of blockchain technology. Fabric's consensus modularity provides superior performance and energy efficiency for our use case compared to energy-intensive counterparts like proof-of-work, and to a lesser extent, proof-of-stake systems, while addressing scalability concerns inherent to myriad permissionless networks. Although initial investment costs may be higher than traditional databases, Fabric's operational expenses can be lower than those of permissionless blockchains, and its modular architecture allows for customized, scalable solutions tailored to specific supply chain needs potentially even relieving stakeholders from proprietary ERP software vendor lock-in. Unlike typical permissionless blockchain networks like Bitcoin [23] and Ethereum [24] that rely on probabilistic consensus algorithms to achieve ledger consistency among participants, a permissioned Fabric network employs deterministic consensus algorithms thus yielding benefits crucial to fit the needs of enterprise demands. Chief among those being the ability to store records of invalid transactions to facilitate subsequent audits that would otherwise be lost. Additionally, the separation of transaction *execution*, *ordering*, and *validation* phases, as shown in Fig. 1, significantly enhances consensus modularity alongside determinism which benefits predictability, consistency, and reliability. Furthermore, determinism prevents varying interpretations of transactions, thereby nullifying the risk of *forks*, i.e., a divergence in the blockchain's transaction history thus enabling more than one interpretation of the records therein, which can otherwise cause confusion between network participants.

Our traceability framework may be extended to achieve GDPR [25] compliance by storing personal data *off-chain* and using the blockchain to record only the hashes of this data. This approach allows the system to honor the right to deletion and be forgotten, as the off-chain data can be modified or deleted while maintaining the integrity and traceability of the blockchain through cryptographic hashes. For enhanced privacy, organizations within channels can even construct private data collections (PDCs) to exchange secret data to transact upon: this allows them to exchange data they may be apprehensive to share with other stakeholders in a channel while simultaneously recording accessible transactions, e.g., a sign from one entity that the secret data is accurate.

3. Methodology: cotton-to-shirt apparel supply chain simulation

We used Docker v27.2.1 on an 11th Gen Intel Core i7 2.80GHz with 16 GB memory running Ubuntu 24.04.1 LTS to create and run separate containers for each peer node representing different organizations in the supply chain shown

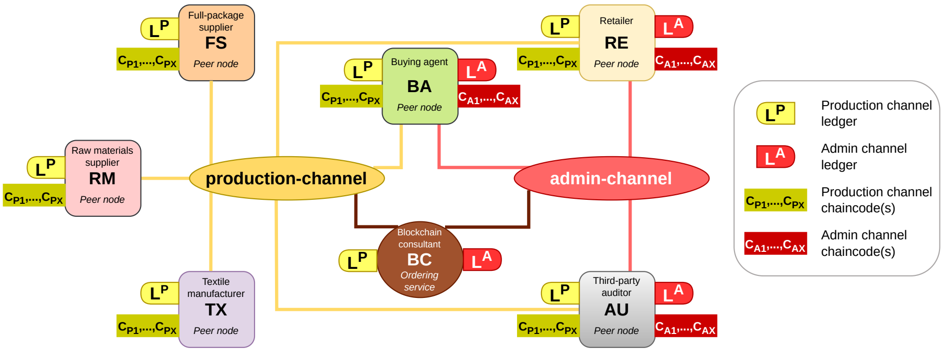


Fig. 2. Network topology for apparel supply chain simulation.

in Figure 2. Each peer hosts the necessary components, such as the chaincode, ledger, and state database, specific to that organization based on the channel(s) they have joined. All simulation functionality, results, and instructions, are available at `https://github.com/ritwiktakkar/ism_WEave`.

3.1. Network

The topology shown in Fig. 2 is influenced by Appelbaum and Gereffi's framework [26] and the experimental setup used by Agrawal et al. [19]. The two channels, "production-channel" (PC) and "admin-channel" (AC), serve complementary functions. PC, consisting of six production stakeholders, functions as a trace for all value-added process(es), like asset transfer transactions. AC, meanwhile, consisting of only RE, BA, and AU, logs indirect manufacturing process transactions, like the purchase order (PO) from RE to BA, that must be kept confidential from upstream suppliers. Realizing channel separation is motivated by Gereffi's description of BDCCs where retailers design and market, but do not "make" the products they sell [27]. Moreover, when it comes to "soft goods," e.g., apparel, owing to an understandable lack of intimate knowledge of necessary supplier networks and product characteristics, retailers often delegate management of the upstream production process to BAs. However, by having access to the PC ledger, RE and AU can always access the same real-time production information that traditionally, only the BA and its upstream partners had access to.

Additionally, we identify the need for an organization to oversee the required technical aspects of this approach. Specifically, it is infeasible to delegate chaincode development, validation, and the overall management of the blockchain network to typical real-world BDCC stakeholders. Therefore, we added a seventh organization, the blockchain consultant (BC), to oversee the technical aspects of this network and participate as a verifiably trusted third-party for all other stakeholders to rely on for blockchain-related services. Smart contracts proposed by the BC alleviate issues of trust that may otherwise arise if a smart contract is instead proposed by a competing organization of one that is also part of the network. To reinforce neutrality, BC node(s) only serve as orderers and are therefore unable to transact directly.

Fabric enforces two checks before chaincode deployment based on information stored in a `configtx.yaml` file that is provided by a network administrator. This file tells Fabric, when creating a channel, which nodes from which organization(s) are allowed to join, and the required endorsement policies to approve chaincode definitions ("*LifecycleEndorsement*") and transactions ("*Endorsement*"). The endorsement policies in this simulation are structured such that all participants must approve the chaincode in each channel before it is available for invocation. Contrastingly, the endorsement policies to invoke chaincode are such that any entity belonging to a given channel may do so. Despite the endorsement policies set in such a way that any channel member may invoke chaincode committed to their channel, certain functions therein programmatically check the membership service provider ID (*MSPID*) of who is invoking chaincode, and consequently (dis)allow its invocation based on the result. For example, to restrict the ability of which stakeholder can add which kind of asset to the ledger, PC chaincode is devised such that only RM can add cotton bale and lots containing it, only TX can add unfinished fabric and lots containing it, etc.

3.2. Requirements

Table 1 contains a set of technical requirements mapped from a realistic collection of business and regulatory needs spread among the simulated network participants. These requirements are split into two types: functional, which focus on the specific business logic and requirements that the supply chain network must satisfy, and operational, which outline expectations regarding the overall system behavior, performance, and security aspects. As discussed earlier, it is expected that BC will consolidate the technical requirements to then generate and propose chaincode which stakeholders can independently verify before endorsing.

Table 1. Simulation technical requirements.

#	Requirement	Type	Fulfilled by
1	Access control	Functional	Chaincode: `SPEC_IsInvokedByAllowedOrg`
2	Asset uniqueness	Functional	Chaincode: `SPEC_{IsNewAsset, NoDuplicateAssetIn{State, ThisLot}}`
3	Chronological ordering	Functional	Chaincode: `SPEC_Chronology`
4	Ownership transfer	Functional	Chaincode: `SPEC_PreviousOwnerListed`
5	Data completeness	Functional	Use of `struct` (typed fields collection) in chaincode
6	Material traceability	Functional	Chaincode: `SPEC_LotConsistency`
7	Mass-balancing	Functional	Chaincode: `Get{PercentageDifference, ContentWeight}`
8	Non-repudiation	Operational	Fabric architecture
9	Data immutability	Operational	Fabric architecture
10	Participant authentication	Operational	Fabric architecture
11	Consensus	Operational	Fabric architecture
12	Scalability	Operational	Fabric architecture
13	Data confidentiality and isolation	Operational	Fabric architecture

3.3. Generating the trace

The production trace is split channel-wise based on Figure 2. The AC trace is as follows. First,, RE generates a purchase order (PO) based on which BA proposes upstream suppliers, namely, RM, TX, and FS, and their specific manufacturing facilities selected to fulfill the order based on the PO. If, and only if, both, RE and AU approve the proposed factories, can BA issue a production plan which contains the approved factories. Again, both RE and AU are required to approve the production plan. Once the production plan has been approved, the PO status and order status can be updated, too. After all this has occurred, production may begin, i.e., the PC ledger can be mutated.

The PC trace is relatively more involved, as one may expect. Figure 3 highlights the various shirt order sizes that were simulated, and the corresponding amounts of raw materials used for each simulation. Stakeholders RM, TX, and FS, all manufacturers, mutate the ledger with chaincode invocations representing the movement of goods in Figure 4. Each asset type is designed using a unique Golang `struct` containing several realistic and important fields. Most fields, especially those that have needs for dynamic adjustments, such as if an auditor decides to flag a certain asset, or if an asset fails some inspection, have the relevant *setter* functions enabling this. Moreover, additional verification checks are programmed into chaincode such that if an asset is flagged or missing an inspection approval, for whatever reason, it cannot be added to a lot. Such mechanisms provide implicit trust based on asset type, e.g., when an asset is in a lot, it means it was not flagged, and it passed relevant inspection. However, to account for retrospective findings or new information, all asset types, including lots and cartons can be flagged if a reason is provided. Once an asset is flagged, however, only BA, RE, or AU can remove the flag. Cotton bale represents the primary input material, serving as the initial entry in the PC ledger. RM is responsible for processing bales of cotton into yarn that are consolidated into lots delivered to TX. Cotton yarn lots are then turned into finished fabric and sent to FS in lots. FS receives lots of finished fabric, cuts them into parts, and creates assembled garments by adding in buttons stored in inventory. Finally, assembled garments are batched together in cartons, which are then loaded for delivery to a port of RE or BA's choice.

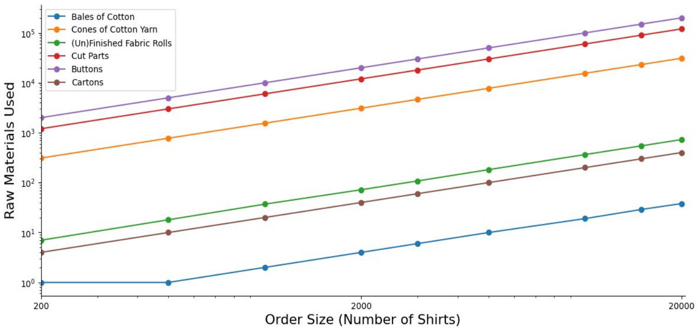
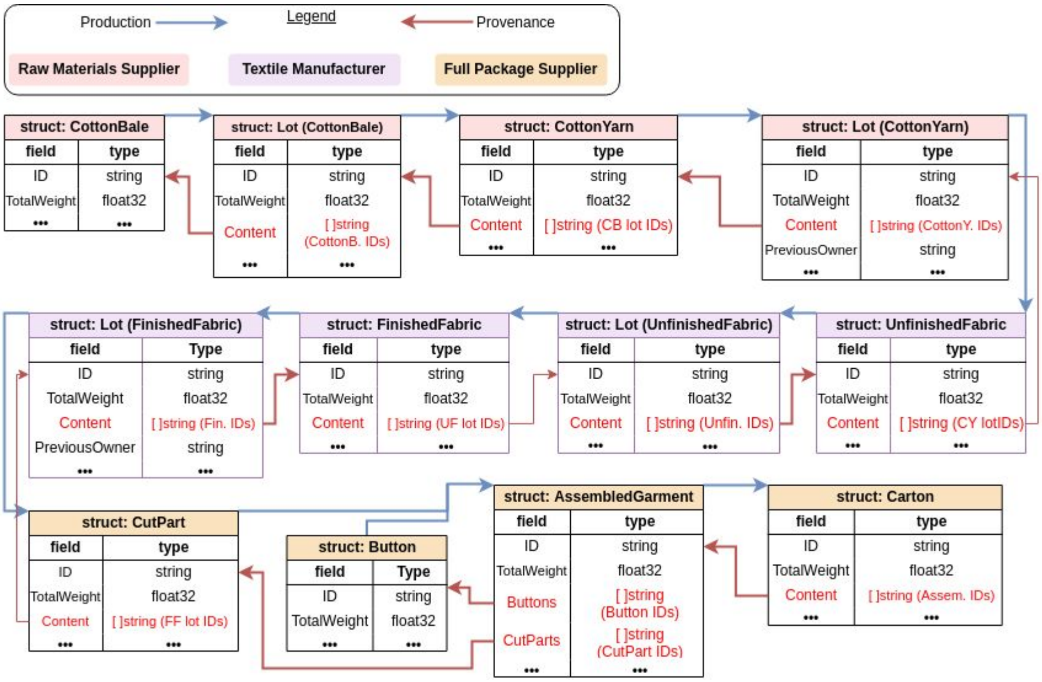


Fig. 3. Raw materials needed for the various order sizes simulated.



struct: CottonBale

field	type
ID	string
TotalWeight	float32
•••	•••

struct: Lot (CottonBale)

field	type
ID	string
TotalWeight	float32
Content	[ ]string (CottonB. IDs)
•••	•••

struct: CottonYarn

field	type
ID	string
TotalWeight	float32
Content	[ ]string (CB lot IDs)
•••	•••

struct: Lot (CottonYarn)

field	type
ID	string
TotalWeight	float32
Content	[ ]string (CottonY. IDs)
PreviousOwner	string
•••	•••

struct: Lot (FinishedFabric)

field	Type
ID	string
TotalWeight	float32
Content	[ ]string (Fin. IDs)
PreviousOwner	string
•••	•••

struct: FinishedFabric

field	type
ID	string
TotalWeight	float32
Content	[ ]string (UF lot IDs)
•••	•••

struct: Lot (UnfinishedFabric)

field	type
ID	string
TotalWeight	float32
Content	[ ]string (Unfin. IDs)
•••	•••

struct: UnfinishedFabric

field	type
ID	string
TotalWeight	float32
Content	[ ]string (CY lotIDs)
•••	•••

struct: CutPart

field	type
ID	string
TotalWeight	float32
Content	[ ]string (FF lot IDs)
•••	•••

struct: Button

field	Type
ID	string
TotalWeight	float32
•••	•••

struct: AssembledGarment

field	type
ID	string
TotalWeight	float32
Buttons	[ ]string (Button IDs)
CutParts	[ ]string (CutPart IDs)
•••	•••

struct: Carton

field	type
ID	string
TotalWeight	float32
Content	[ ]string (Assem. IDs)
•••	•••

Fig. 4. Production channel trace depicting transfers of assets, and attributes therein, as simulated.

4. Results

We measured key aspects of the simulation regarding various order sizes to study scalability and performance. Figures 5-7 depict that as the number of shirts produced increases, so too does the: (a) time it takes to run all relevant transactions, (b) number of blocks in the ledger, (c) peak memory consumption during runtime, and (d) size of the state and ledger. In the average CPU usage trends shown in Figure 5b, comprised of per second readings taken from the output of the Linux `pidstat` [28] command tied to the various simulation scripts, both overall, i.e., includes readings where `%CPU` was 0 indicating another system resource such as I/O was the bottleneck, and active, i.e., excludes readings where `%CPU` was 0, readings reveal a more logarithmic relationship between CPU load and growing order sizes thus indicating efficiencies in compute. We also modified the block sizes to measure their effect on performance, as shown in Figures 8-9. Specifically, we constructed three block sizes by modifying the relevant values in the aforementioned `configtx.yaml` file. The three block sizes we created in addition to the default ("medium") size (`MaxMessageCount: 10, AbsoluteMaxBytes: 99MB, PreferredMaxBytes:`

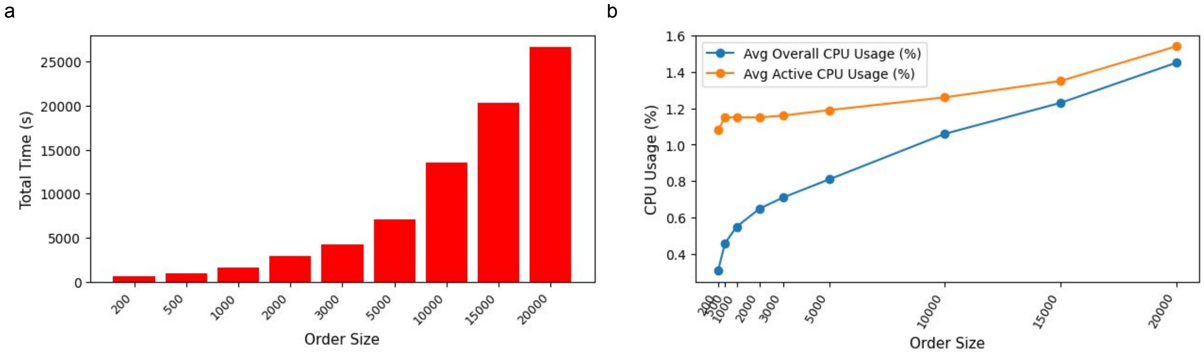


Fig. 5. (a) Simulation runtime versus order size; (b) Average CPU usage versus order size.

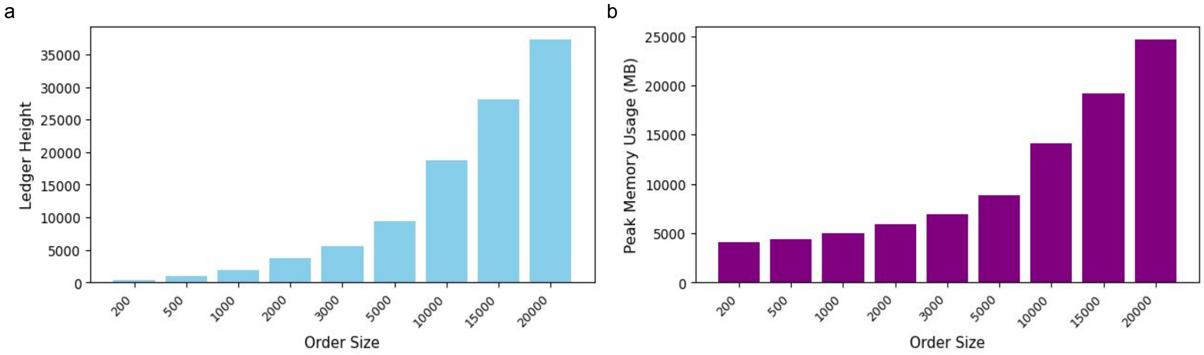


Fig. 6. (a) Ledger height (number of blocks) versus order size; (b) Peak memory usage (MB) versus order size.

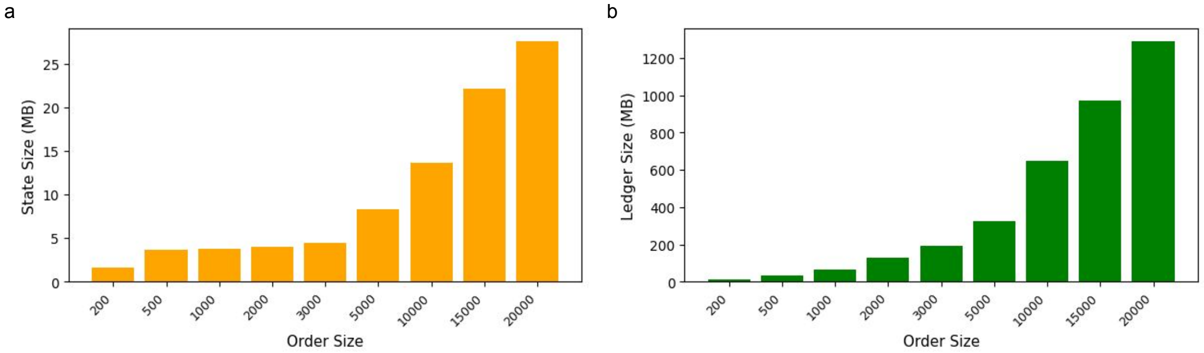


Fig. 7. (a) State size (MB) versus order size; (b) Ledger size (MB) versus order size.

`512KB`) were: “small” (`MaxMessageCount: 5, AbsoluteMaxBytes: 10MB, PreferredMaxBytes: 256KB`), “large” (`MaxMessageCount: 50, AbsoluteMaxBytes: 500MB, PreferredMaxBytes: 2MB`), and “very large” (`MaxMessageCount: 100, AbsoluteMaxBytes: 1000MB, PreferredMaxBytes: 8MB`).

5. Discussion

An earlier study [29] conducted an analysis of blockchain projects in supply chain management and concluded that out of the 43 publications reviewed, “no example exists which has the aim of increasing the transparency of complex manufacturing supply chains, and which enables the mapping of complex assembly processes, an efficient

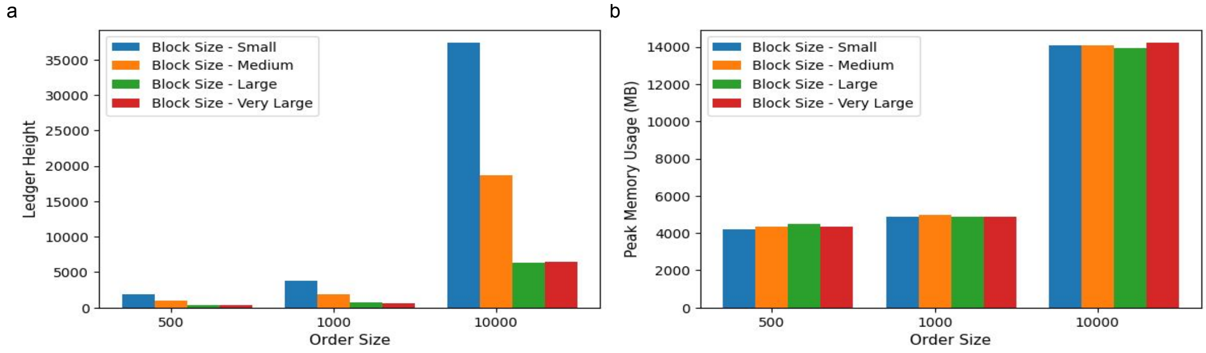


Fig. 8. (a) Ledger height versus order and block size; (b) Peak memory usage versus order and block size.

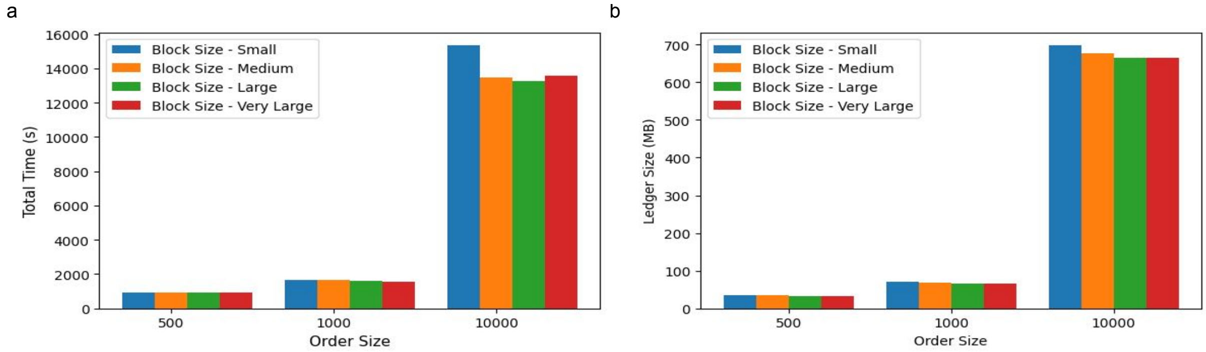


Fig. 9. (a) Total simulation runtime versus order and block size; (b) Ledger size (MB) versus order and block size.

auditability of all assets, and an implementation of dynamic adjustments." The PC trace shown in Figure 4 contains several examples of assets of a certain type being lotted together, e.g., several discrete units, i.e., *cones*, of cotton yarn are consolidated into separate cotton yarn *lots* where each cone of yarn's ID is recorded in the lot `struct`'s `content` field. Certain asset types' lots are then used as input material to create another asset type, e.g., cotton yarn lots generate discrete units, i.e., *rolls*, of unfinished fabrics at TX's facility. Moreover, specification functions, i.e., functions whose name starts with "`SPEC_`" in Table 1, satisfy all functional requirements as invoked for dynamic adjustments, while Fabric's architecture handles the operational requirements. For efficient auditability of assets, WEave empowers not just a third-party auditor to query the ledger, but also all participants therein. Furthermore, each channel's chaincode(s) provides a specific list of audit-specific functions, such as *getter* methods to get any asset's information, including lots and contents therein. Additionally, any time a lot is generated, it must contain a list of unique assets such as cotton bale or cotton yarn, and because each asset has a required `TotalWeight` field, the chaincode automatically calculates and records the percentage difference between what a user manually must enter for the lot weight versus the automatically calculated sum of the weight of assets in content comprising it. This has the potential to spot infractions tied to mass-balancing instantly. However, there remains the possibility that a bad actor may simply calculate the expected total weight for a lot regardless of its actual weight and so on, but this can be prevented using additional verification steps such as only allowing weight entries to be inputted by machines inspected by closed-circuit televisions (CCTVs). It is crucial to acknowledge that blockchain technology cannot itself verify the veracity of input: that requires additional steps and careful consideration.

The quantitative results shown in §4 demonstrate that it is computationally feasible to argue in favor of adopting a framework like WEave for enhanced traceability measures in BDCCs. Neither memory size or peak usage nor CPU usage indicate serious concerns for order sizes ranging from 200 all the way to 20,000 as shown here. In fact, apparel production experts may even argue that the simulations were run without considering optimal scalability in production methods, resulting in more raw materials simulated than necessary. The different block sizes configured and shown in

Figures 8-9 do not seem to have any significant effect on compute or memory apart from the ledger heights, i.e., the number of blocks in the blockchain. This could potentially be due to the values chosen for each of the various block size configurations.

While integrating a robust blockchain-based traceability framework may enhance transparency, technology, including blockchain, only constitutes part(s) of a comprehensive solution that must also address the social, regulatory, and operational complexities of BDCCs. Effective implementation requires collaboration among stakeholders, creation of and adherence to robust international standards, and integration with existing systems. Moreover, we acknowledge that the capabilities of WEave, and likely similar blockchain frameworks/platforms, are capped to the extent a requirement may be expressed in technical terms that can be enforced programmatically. For requirements unable to be mapped this way, such as gauging employee satisfaction, it remains critical to pursue relatively nontechnical avenues. Ultimately, it is impossible to capture the nuance and account for unexpected events that may appear in a real-world experiment when using a simulation to validate a framework such as WEave. We also acknowledge that running a local network simulation mitigates the potential for several network scalability problems or similar issues. A major real-world challenge in implementation, exceeding the scope of this work, is workforce training, especially in upstream supplier facilities, to familiarize workers with necessary technical tools and workflows. Furthermore, realistic adoption of such a framework is limited by the initial implementation costs, especially among networks with chronic downward price pressures approaching slimmer margins, including training of personnel and maintenance for tools, and the need for technological infrastructure across all participating entities. Thus arises the important question of incentive: to what extent is the cost of incorporating robust traceability measures offset by the benefit, and under what circumstances, when so? Additionally, achieving social consensus and interoperability among diverse stakeholders poses significant challenges. Who, in a BDCC, will be responsible for gathering guidelines from multiple stakeholders to subsequently map onto technical requirements to develop smart contracts based on? Perhaps these concerns corroborate our decision to introduce an entirely new entrant among interested supply chain networks: a neutral entity responsible for maintaining and updating the blockchain framework, i.e., a blockchain consultant organization (BC).

6. Conclusion

Supply chain managers can spearhead the transition to Industry 4.0 by increasing transparency to streamline operations and achieve ambitious social goals using tools and methods such as those defined in this study. The inefficiencies of BDCCs reflect labor-intensive methodologies, where record-keeping is often chaotic and inadequate. WEave is a demonstrably effective blockchain-based traceability framework capable of mapping assembly processes for complex parts, efficient auditability of all assets, and implementation of dynamic adjustments in a BDCC as shown through the apparel supply chain simulation. Based on our simulation results on order sizes of shirts ranging from 200 to 20,000, the framework scales well and predictably. While this paper was set in the specific context of apparel supply chains, domain experts from various other industries under the broad BDCC umbrella may tailor WEave to fit their needs. Still, it is only one part of the solution set to address the large-scale sociotechnical traceability challenge, a critical step towards enhanced transparency in supply chains. We urge international non-profit organizations calling for increased transparency in BDCCs and providing voluntary reporting frameworks to large companies to incorporate aspects of blockchain technology, as shown with WEave, to gather more credible and immutable data. We also urge focal firms in BDCCs to draw upon our work and its ilk en route to more transparent supply chains, parts of which may be shared with end-customers as well.

References

[1] G. Gereffi, M. Korzeniewicz, Commodity chains and footwear exports in the semiperiphery, in: W. Martin (Ed.), Semiperipheral States in the World-Economy, GREENWOOD PRESS, pp. 45–68.
URL `https://hdl.handle.net/10161/11454`

[2] D. Miller, Negotiating international framework agreements in the global textile, garment and footwear sector 4 (2) 215–239, publisher: SAGE Publications. `doi:10.1177/1468018104045110`.

[3] R. Guo, H. L. Lee, R. Swinney, Responsible sourcing in supply chains 62 (9) 2722–2744, publisher: INFORMS. `doi:10.1287/mnsc.2015.2256`.

[4] European Commission: Directorate-General for Justice and Consumers, Safety Gate, 2023 results – Safe products, safe consumers, Publications Office of the European Union. `doi:10.2838/407812`.

[5] V. Kumar, C. Hallqvist, D. Ekwall, Developing a framework for traceability implementation in the textile supply chain 5 (2) 33, number: 2 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/systems5020033`.

[6] S. Saberi, M. Kouhizadeh, J. Sarkis, L. Shen, Blockchain technology and its relationships to sustainable supply chain management 57 (7) 2117–2135, publisher: Taylor & Francis. `doi:10.1080/00207543.2018.1533261`.

[7] World Health Organization, A study on the public health and socioeconomic impact of substandard and falsified medical products, World Health Organization, section: vii, 67 p.
URL `https://iris.who.int/handle/10665/331690`

[8] T. K. Mackey, G. Nayyar, Digital danger: a review of the global public health, patient safety and cybersecurity threats posed by illicit online pharmacies 118 (1) 110–126. `doi:10.1093/bmb/ldw016`.

[9] H. Boyacı, H. T. Temiz, R. S. Uysal, H. M. Velioğlu, R. J. Yadegari, M. M. Rishkan, A novel method for discrimination of beef and horsemeat using raman spectroscopy 148 37–41. `doi:10.1016/j.foodchem.2013.10.006`.

[10] R. Cole, M. Stevenson, J. Aitken, Blockchain technology: implications for operations and supply chain management 24 (4) 469–483, publisher: Emerald Publishing Limited. `doi:10.1108/SCM-09-2018-0309`.

[11] The modern slavery act 2015 (transparency in supply chains) regulations 2015, publisher: King's Printer of Acts of Parliament.
URL `https://www.legislation.gov.uk/uksi/2015/1833/contents/made`

[12] The california transparency in supply chains act.
URL `https://oag.ca.gov/SB657`

[13] C. C. Campaign, Who we are.
URL `https://cleanclothes.org/about`

[14] S. Abeyratne, R. Monfared, Blockchain ready manufacturing supply chain using distributed ledger 05 1–10.
URL `https://hdl.handle.net/2134/22625`

[15] F. Dong, P. Zhou, Z. Liu, D. Shen, Z. Xu, J. Luo, Towards a fast and secure design for enterprise-oriented cloud storage systems 29 (19) e4177. `doi:10.1002/cpe.4177`.

[16] T. Norton, J. Beier, L. Shields, A. Househam, E. Bombis, D. Liew, A guide to traceability: A practical approach to advance sustainability in global supply chains | UN global compact.
URL `https://unglobalcompact.org/library/791`

[17] N. Sinkovics, S. F. Hoque, R. R. Sinkovics, Rana plaza collapse aftermath: are CSR compliance and auditing pressures effective? 29 (4) 617–649, publisher: Emerald Group Publishing Limited. `doi:10.1108/AAAJ-07-2015-2141`.

[18] G. LeBaron, J. Lister, Ethical audits and the supply chains of global corporations, backup Publisher: ARRAY(0x55812cc917a0) Series: SPERI Global Political Economy Briefs.
URL `https://eprints.whiterose.ac.uk/96303/`

[19] T. K. Agrawal, V. Kumar, R. Pal, L. Wang, Y. Chen, Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry 154 107130. `doi:10.1016/j.cie.2021.107130`.

[20] C. Prybila, S. Schulte, C. Hochreiner, I. Weber, Runtime verification for business processes utilizing the bitcoin blockchain 107 816–831. `doi:10.1016/j.future.2017.08.024`.

[21] I. O. for Standardization, ISO 9000:2015(en), quality management systems — fundamentals and vocabulary.
URL `https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en`

[22] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, J. Yellick, Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, EuroSys '18, Association for Computing Machinery, pp. 1–15. `doi:10.1145/3190508.3190538`.

[23] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system.
URL `https://bitcoin.org/bitcoin.pdf`

[24] V. Buterin, A next-generation smart contract and decentralized application platform.
URL `https://ethereum.org/en/whitepaper/`

[25] C. of the European Union, Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (general data protection regulation) (text with EEA relevance).
URL `https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04/eng`

[26] R. P. Appelbaum, G. Gereffi, Power and profits in the apparel commodity chain, in: E. Bonacich, L. Cheng, N. Chinchilla, N. Hamilton, P. Ong (Eds.), Global Production, The Apparel Industry in the Pacific Rim, Temple University Press, pp. 42–62.
URL `http://www.jstor.org/stable/j.ctt14bt3hw.6`

[27] G. Gereffi, The organization of buyer-driven global commodity chains: How u.s. retailers shape overseas production networks., in: G. Gereffi, M. Korzeniewicz (Eds.), Commodity Chains and Global Capitalism, Praeger, pp. 95–122.
URL `https://hdl.handle.net/10161/11457`

[28] S. Godard, pidstat(1) - linux manual page.
URL `https://man7.org/linux/man-pages/man1/pidstat.1.html`

[29] F. Dietrich, Y. Ge, A. Turgut, L. Louw, D. Palm, Review and analysis of blockchain projects in supply chain management 180 724–733. `doi:10.1016/j.procs.2021.01.295`.