DECISION SCIENCES
A JOURNAL OF THE DECISION SCIENCES INSTITUTE

Decision Sciences
Volume 0 Number 0
June 2019

Is Blockchain a Silver Bullet for Supply Chain Management? Technical Challenges and Research Opportunities

Akhil Kumar[†]
Department of Supply Chain and Information Systems, Smeal College of Business, Pennsylvania State University, 420 Business Building, University Park, PA, 16802, USA, e-mail: akhilkumar@psu.edu

Rong Liu
School of Business, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken, NJ, 07030, USA, e-mail: rong.liu@stevens.edu

Zhe Shan
Department of Information Systems and Analytics, Farmer School of Business, Miami University, Oxford, OH, 45056, USA, e-mail: jayshan@miamioh.edu

ABSTRACT

Blockchain technology is based on the idea of a distributed, replicated, and immutable digital ledger that enables parties to conduct business in a trustful and transparent way without the need for a central authority or intermediary. Its most popular application thus far is in payment system applications, e.g., bitcoin. This disruptive technology is expected to contribute significant business value to multiple industry sectors, including supply chain management (SCM), where it can provide greater visibility, accountability and trust in interorganizational business collaboration. In this article, we review some fundamental concepts of Hyperledger Fabric, one of the most mature permissioned blockchain implementations. Further, we use the context of a food supply chain to highlight key design and implementation challenges for blockchain, and provide a strategic assessment of its prospects. Our aim is to dispel misguided notions and myths about blockchain as a silver bullet for all businesses. We believe it is important to penetrate the hype to allow a more realistic understanding of this technology. Blockchain is a high-cost, high-overhead storage medium. It is viable only when its higher cost is counterbalanced by the set of benefits that are identified by a careful and thorough analysis of the business. Thus, it will be used mainly for storing important data related to interorganizational transactions among partners where trust is lacking and provenance and visibility are critical. Our paper offers enterprises a systematic way to understand the real costs and risks of blockchain adoption. The insights gained in the SCM context also apply to other areas such as financial services and healthcare that could leverage the full potential of blockchain technology. [Submitted: August 20, 2018. Revised: April 26, 2019. Accepted: May 10, 2019.]

[†]Corresponding author.

Subject Areas: Blockchain, Data Management, Food Supply Chain, Hyperledger Fabric, Smart Contract, Supply Chain Management, and System Configuration.

INTRODUCTION

Blockchain is an emerging technology (Nakamoto, 2008; Swan, 2015; IBM Institute for Business Value, 2016; Pilkington, 2016; de Kruijff & Weigand, 2017) that enables parties to conduct business transparently and maintain a distributed, immutable, and tamper-proof digital ledger of transactions without a central authority. Applications of blockchain technology are rapidly emerging in payment systems (Nakamoto, 2008), supply chains (Lohade, 2017), healthcare (Ekblaw, Azaria, Halamka, & Lippman, 2016; Robert Plant, 2017), and in other areas of business. These are primarily B2B applications at this stage as noted by Lannquist (2018), although B2C and C2C applications are also emerging (see Table 1). Blockchain is believed to be a promising technique for decentralized marketplaces, which match sellers and buyers and ensure transaction integrity and security (Subramanian & Hemang, 2017). Its major innovation is that it allows business partners to transfer digital assets or business information about orders, receipts, payments, etc., across the internet by recording the data on a shared, distributed immutable ledger without the need for a centralized third party. The ledger is a write-once-only log produced after ratification by multiple partners where each successive block contains the hash value or a signature of the previous block in the chain. Thus, a transaction once recorded in a block is protected from tampering because any attempt to change it will invalidate the block signature stored in the successive blocks and thus can be easily discovered. This property is called "immutability." Thus, in a supply chain network, the ledger can secure provenance information to serve as a reliable record of product safety and authenticity.

With a growing interest in leveraging e-business to coordinate decisions by integrating diverse and sometimes conflicting objectives of various partners in supply chains (Vakharia, 2002), the distributed and sustainable aspects of blockchains find applications in the supply chain ecosystems, thus offering a digital transformation technology that can simplify business processes to improve profitability

Table 1: Applications of blockchain technology in different sectors.

	Business	Consumer
Business	B2B	B2C
	Supply chain	Payments
	Finance	Finance
	Healthcare	Proxy voting
Consumer	C2B	C2C
	Job referral	Taxi service
	Home rental	Home rental

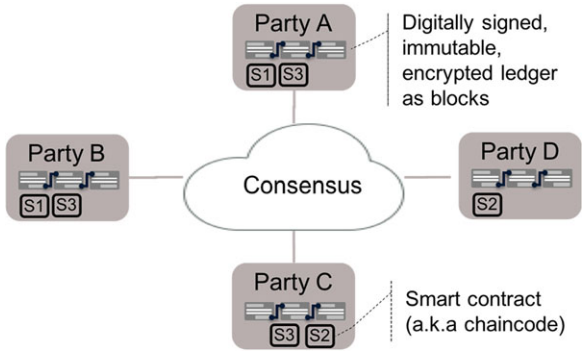
and deliver a better customer experience. The security and immutability features of blockchain help to ensure provenance and safety for shipments of drugs, food items, critical components (say, for an aircraft), and so on. The visibility feature of blockchains facilitates easier and lower-cost audits of financial transactions. Finally, because transactions are performed through consensus reached among multiple parties, the risk of dealing with unknown, far-flung parties is reduced, thus promoting such interactions.

Confusion abounds between blockchain and bitcoin. Bitcoin was introduced by Nakamoto (2008). It was, in fact, the first application of blockchain technology for making secure and anonymous payments. But the underlying technology has numerous other business applications. Many novel blockchain initiatives have been proposed, particularly for supply chains (e.g., in global trade, food supply chains, and high-value goods). Lannquist (2018) reported that about 140 companies had implemented blockchain prototypes by 2017. However, very few were fully operational, and Gartner believed that 90% of blockchain initiatives in supply chains would fail (Gartner, 2017). Thus far, bitcoin remains the most successful real-world deployment of this technology. By contrast, adopting blockchain technology for business applications poses bigger challenges because secure payment is only one aspect of such applications. Several other factors such as trust, visibility, privacy, data security, scalability, and performance must be addressed in a satisfactory manner (White, Daniel, Ward, & Wilson, 2007) for the technology to go mainstream. Hence, this is a fertile area for research for building the next-generation blockchain technology.

Numerous product offerings such as Ethereum (Buterin, 2013), Hyperledger (Cachin, 2016), Ripple (Todd, 2015), Quorum (Wattenhofer & Foerster, 2017), and so on, have mushroomed in just the last two to three years and are being applied in prototype applications. These products extend blockchain technology to private (or permissioned) blockchain solutions capable of executing flexible smart contracts (computerized transaction protocols agreed to by participants) rather than just Bitcoin-style transactions (Christidis & Devetsikiotis, 2016). Such extensions make blockchain well suited for enterprise applications such as supply chains, which handle complicated interorganizational transactions, often in a private environment. Moreover, due to concerns surrounding data and transaction privacy, most B2B blockchain prototypes were implemented in private, not in public networks such as Bitcoin (Lannquist, 2018). Consequently, private or permissioned blockchains warrant more attention.

Blockchain technology has the potential to revolutionize applications and redefine the digital economy (Underwood & Sarah, 2016). It has attracted tremendous interest from both SCM and Information Systems communities. Deloitte professionals have identified four "pain points"—traceability, compliance, flexibility, and stakeholder management—and see promise in blockchain for relief (Pawczuk, 2018). A recent comprehensive study by Staples et al. (2017) provided interesting high-level design alternatives for illustrative blockchain use cases (e.g., agricultural supply chains), and evaluated these options in terms of system scalability, interoperability, and latency. One key finding was that "Supply chains are a highly promising domain for the application of blockchain technology. " Babich and Hilary (2018) identified key strengths (e.g., visibility and aggregation)

Figure 1: A sample blockchain network.



and weaknesses (e.g., lack of privacy and standardization), and research themes (e.g., information and automation) as they relate to blockchain applications in operations management. Our present work is an effort to understand blockchains from a more technical perspective. We describe fundamental technology components of blockchain, including network design, consensus, smart contracts, and data management, and discuss the design of each component and potential technical challenges while applying blockchain to SCM. These can also be viewed as research opportunities.

This paper is organized as follows. In the next section, we present an overview of permissioned blockchain systems to make the fundamental technological concepts understandable. In the following section, a use case for a food supply chain illustrates the major functionalities of a blockchain network. Later, we elaborate on the key challenges that arise from an end-user or designer perspective in blockchain system design, in particular in relation to smart contract design, data management, and channel configuration. Finally, we provide an in-depth assessment and outlook of blockchain technology from a strategic standpoint. The conclusions section suggests some directions for the future of this technology.

FUNDAMENTALS OF BLOCKCHAIN TECHNOLOGY

Architecture and Comparison with Interorganizational System Technologies

In contrast to traditional business models, where some parties (e.g., banks) serve as central repositories of trust, a blockchain system allows each business party to maintain a copy of a digitally signed, encrypted ledger (see Figure 1). A ledger is organized as linked *blocks* with each block containing one or more transactions. *Smart contracts* (also known as *chaincode*) implementing business logic/contracts are installed on the blockchain system and executed by parties to create transactions. Moreover, each transaction is validated and approved by multiple relevant parties based on an agreement that must be predefined before it can be appended to

Table 2: Comparison of blockchain with other IOS technologies.

Features	EDI SOA	Blockchain	Blockchain Pros/Cons
Network Topology	Dyadic (1:1) or Hub-and-spoke (1:N)	Multi-lateral (M:N)	Pro: Fully decentralized and more resilient
Data Governance	Data actions (create, read, update, delete) are enacted unilaterally	Consensus protocols constrain and validate data actions by multiple parties	Pro: Shared control of data, greater data integrity and reliability
Business contract and governance	Not enforced automatically	Automated execution through smart contracts is enforced	Pro: Automation without human intervention
Provenance	Provisioned through a common centralized server despite no assurance of immutability	Comes at no additional cost as a byproduct of the technology	Pro: Inherent and tamperproof
Visibility	Provided through a common centralized server	Can be fully decentralized	Pro: Reliable without a single point of failure
Cost	High setup cost; low to medium transaction cost	High setup cost (due to the learning curve and technique immaturity) High transaction cost (varies by consensus mechanism)	Con: higher overhead because of validation and replication
Power and trust	One party dominates the others and forces adoption; high level trust among parties is needed	Power is disseminated among parties; parties trust the underlying Blockchain infrastructure	Pro: Blockchain reduces the need for trust among parties and for an intermediary

the ledger of each party. Finally, *consensus mechanisms* (Malkhi & Reiter, 1998; Miller, Xia, Croman, Shi, & Song, 2016) are implemented to ensure that each party maintains an exact replica of the ledger after reaching agreement.

Blockchain is essentially another technology for building interorganizational systems (IOS). In that sense, it is similar to other technologies like EDI (electronic data interchange) and SOA (service-oriented architectures). EDI is based on an asynchronous exchange of messages over proprietary value-added networks (VAN), while SOA invokes web services using open standards in a synchronous manner over the internet. However, our comparison along several dimensions in Table 2 shows that blockchain is inherently different. First, blockchain can

support multilateral relationships among parties in a decentralized network topology. For example, in Figure 1, party A conducts transactions with party B through smart contract S1, and parties C and D collaborate through S2, without involving a central party. In comparison, previous studies have found that a centralized design, e.g., hub-and-spoke, is pervasive in both EDI- and SOA-based IOS (Hart & Saunders, 1997; Löhe & Legner, 2010). Second, in blockchain systems, data is stored on the ledger after validation through a consensus mechanism and is virtually immutable, and thus has greater integrity. It also enables higher resilience because the ledger is replicated on several peer nodes. However, EDI and SOA lack technical mechanisms to automatically enforce data validation by multiple parties, creating vulnerabilities in data integrity and the risk of transaction repudiation (Ratnasingham, 1998).

Finally, the notion of smart contracts implies a self-executing code on the blockchain that automatically implements the terms of an agreement among parties or other business logic. It is for the most part an unbreakable agreement with predefined rules. In addition, smart contracts are deterministic that means the same output will be generated from a given initial state/input. Both smart contracts and SOA can be used to fulfill a business contract between parties, but SOA cannot enforce the business contract automatically. With SOA, a web service is provisioned and controlled by a party. Thus, it is not "unbreakable" in the same sense as a smart contract. A smart contract, once deployed to blockchain, is outside the sole control of any single party. For example, smart contract S3 in Figure 1 is controlled by parties A, B, and C, and cannot be easily altered or terminated (Levi & Lipton, 2018).

The technological components discussed above, e.g., decentralized topology, consensus mechanisms, and smart contracts, highlight a number of intrinsic properties of blockchain. Provenance is defined as "the origin or source of something." It also relates to the idea of traceability in a supply chain. This feature comes for little extra cost as a by-product of the technology since the blockchain maintains a reliable record of all the transactions that were performed. Thus, an independent auditor can obtain a complete audit trail with timestamps of every activity that was performed, and can verify whether it was in accordance with the contractual agreements laid out among the various parties involved. Of course, the setup cost for blockchain is higher than for the existing technologies as is the cost of running transactions because of the overhead cost of validation and replicated storage. As with other IOS technologies where power and trust play a key role in the adoption (Hart & Saunders, 1997), blockchain also relies on a certain amount of trust in the underlying infrastructure of protocols, algorithms, ledgers, and their governance mechanisms, which is often referred to as "meta trust" (Babich & Hilary, 2018).

Depending on who can join it, a blockchain system can be *permissioned* or *permissionless*. A comparison is shown in Table 3. A public blockchain system is accessible to everyone, while a permissioned blockchain system allows only identified parties to join with strict membership control. Consequently, these two types of systems differ in many aspects. From the comparison, we can see that permissioned blockchain systems can better address an enterprise's concerns about transaction security, privacy, and scalability. In this paper, we will focus on

Table 3: Public verus Private blockchains.

	Public (permissionless)	Private (permissioned)
Access control	Everyone can join; Identity is hard to control	Access is controlled; All users are identified
Security & Privacy	Transactions are viewed by anyone. Privacy may be compromised due to public access to transaction history (Genkin, Papadopoulos, & Papamanthou, 2018)	Transactions are viewed by authorized users only to ensure privacy
Consensus	Since users cannot be trusted, proof of work or a similar consensus protocol is required	Since users are trusted, selective endorsement by a few other parties is sufficient to verify transactions
Scalability	Low due to large number of nodes and proof of work consensus protocol that is expensive to run	Relatively higher due to fewer nodes and selective endorsement
Example	Bitcoin, Ethereum	Hyperledger

Hyperledger, one of the most mature permissioned blockchain frameworks (Dinh et al., 2017), to illustrate blockchain systems for enterprise applications.

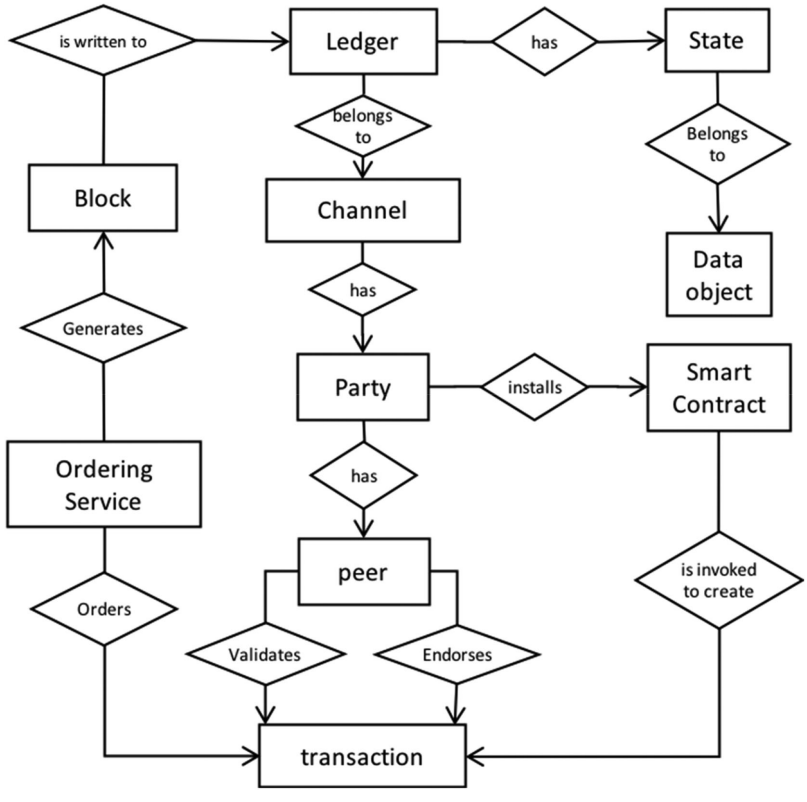
Permissoned Blockchain Metamodel

Hyperledger is an initiative of the Linux Foundation started in December 2015 with several member companies such as IBM, Intel, and JP Morgan. It is an umbrella project of open source blockchain technologies and related tools to support the collaborative development of blockchain-based distributed ledgers.

Figure 2 gives a metamodel for a Hyperledger Blockchain system. The central entity or player in this picture is a party (or a business firm, e.g., a supplier) that interacts with other parties (or business partners, e.g., a customer, shipper). A party is represented on the blockchain network by a node called a *peer*. Transactions performed by any partner such as an order, acknowledgment, shipment notice, invoice, payment, and so on, are all stored in a chronological sequence of successive blocks in the *ledger* kept at every peer. A party associates with one or more *peers* who participate in performing transactions and maintain the ledger on its behalf. A subset of parties can set up a *channel* to conduct business among themselves so that their ledger can be segregated from others.

A *smart contract* (or chaincode) is a protocol or agreement, written in a language like Java or Go, that parties agree to observe on the channel. All read and write operations on the blockchain can be performed *only* by invoking smart contracts. A peer will invoke a smart contract to create a *transaction*, e.g., a purchase order, a payment, and a receipt. The *transaction* is eventually written to a *block* and the block is committed to the ledger in a channel. However, before

Figure 2: A Hyperledger metamodel to describe blockchain entities and relationships.

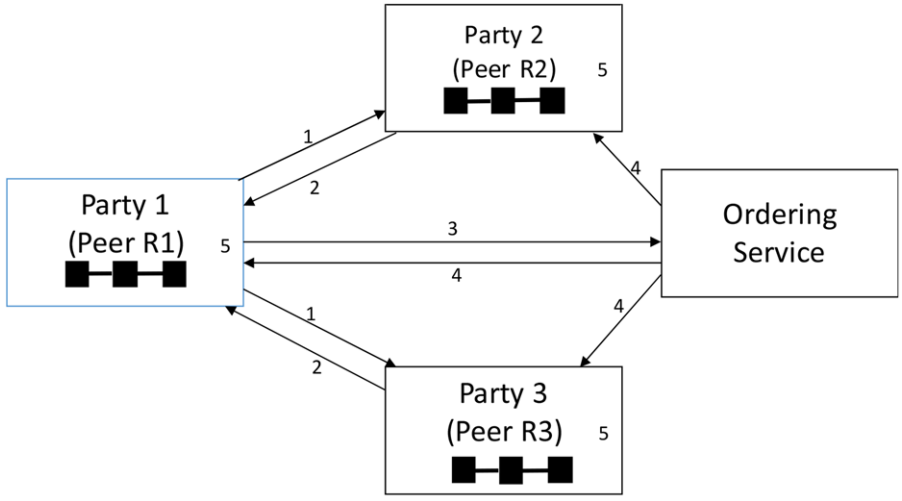


a transaction is written, it must be endorsed by a group of *endorsing* peers and *validated*. Further, it must be ordered by an *ordering service*. Transactions from multiple channels are sorted by channel and time during this ordering process, and blocks are generated from them for each channel. Finally, in addition to transactions, the ledger maintains a state that reflects the values of actual data objects in the real world. The state is stored as attribute-value pairs in a database. A glossary of blockchain terminology can be found in the Appendix (online).

Blockchain Transaction Workflow

Figure 3 shows the process flow diagram of interactions on a Hyperledger Blockchain network with, in general, n business parties. Each party is represented, in general, by one or more *peers* as noted above. For example, Party P1 (as also P2 and P3) has, for simplicity, one peer named R1 (R2 and R3). Further, as shown in the figure, the subgroup of P1, P2, and P3 form a channel to conduct business among themselves. These three parties could assume the roles of a buyer, a supplier, and a producer in a supply chain working together to settle a

Figure 3: A Hyperledger Blockchain process flow diagram.



purchase–sale transaction. In some cases, another peer may join simply as an "observer" of the channel for purposes of monitoring or auditing, without engaging in any transaction.

Within a channel, participating parties can install *smart contracts* that implement certain business logic agreed to by the collaborating parties. A smart contract installed on the blockchain allows a peer to execute a transaction on behalf of its associated business party. A deployed smart contract, say, *S1*, is associated with a predefined *endorsement policy* that specifies the number of endorsements required for valid transactions. A simple endorsement policy may be that a majority of other parties and/or at least two parties within the channel must approve a transaction proposal. Thus, P1 may invoke smart contract *S1* to propose a transaction, and an endorsement policy might specify that this proposal requires endorsements (or approvals) from P2 and P3. Prior to endorsing a proposal, an endorsing party must validate it by simulation as we shall explain shortly.

Another important player in a Hyperledger Blockchain network is the *Ordering Service* consisting of one or more nodes called *Orderers*. An orderer acts as a coordinator that collects transaction proposals from peers, sorts them by channels, timestamps them, organizes them into blocks, and finally sends the blocks to all peers for committing. To balance the workload, a network may be equipped with a cluster of orderers that constitute the ordering service.

Figure 3 also shows the lifecycle of a transaction that consists of five steps as follows:

- Step 1 (*Proposal*): Peer R1 acting on behalf of P1 contacts peers R2 and R3. In this case, P1 is submitting a transaction proposal for endorsement.
- Step 2 (*Endorsement*): R2 and R3 execute the corresponding smart contract *S1* independently and determine the data objects to be read and written by the contract. If these read and write set values match the values they had received in the proposal earlier, they sign the response as proof of endorsement and return it to P1. This is called *validation by simulation*.
- Step 3 (*Assembly*): Peer R1 collects all endorsement responses, checks the results from each response, and determines whether the transaction was properly endorsed. If so, R1 collects all endorsement responses as an uncommitted (provisional) transaction, and forwards it to the ordering service.
- Step 4 (*Ordering*): The orderer collects transactions from peers, sorts them by channel and transaction timestamp, assembles them into blocks, and finally delivers a message containing the newly created block to all peers in the channel for *committing*.
- Step 5 (*Committing*): Each party validates every transaction in a received block to ensure that the versions of the data objects in the transaction match its current version of the local data objects. If a block is valid, it adds the block to its ledger and notifies the initiator (in this case P1) that the transaction was committed successfully. Otherwise, a failure notification is sent.

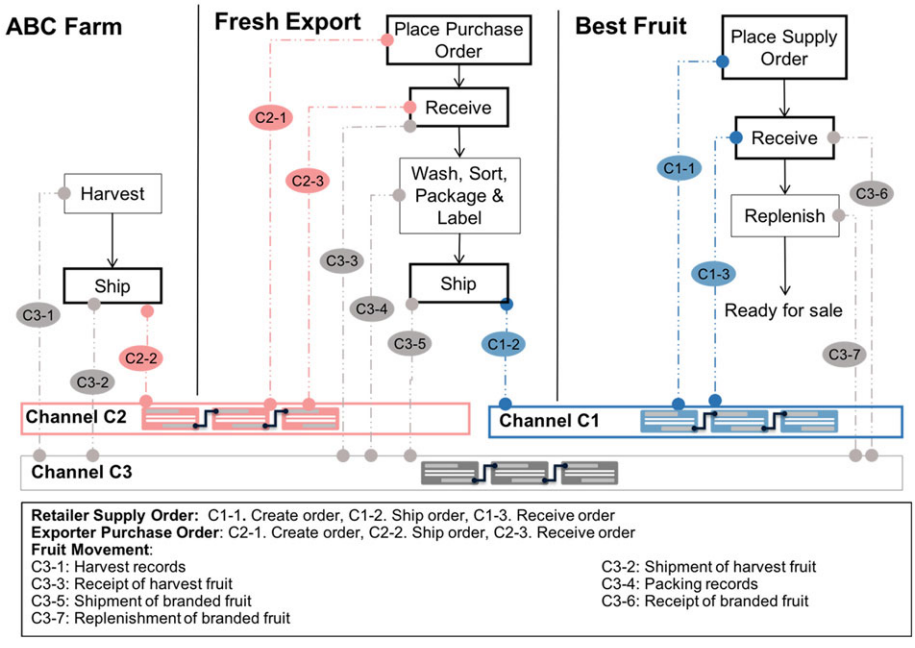
Thus, as we can see, writing new data values on to a blockchain involves a somewhat complex workflow consisting of a series of interactions among nodes in the blockchain network. It is expensive both in terms of processing and storage costs. In designing a blockchain network, several decisions must be made regarding the number of nodes, consensus protocol, on/off chain data, and so on, as we shall discuss at length later.

FOOD SUPPLY CHAIN: A SAMPLE USE CASE

In this section, we present a use case from the food industry to illustrate the concepts outlined above and demonstrate how to design critical components such as channels, smart contracts, and endorsement policies to support a critical supply chain scenario. Food safety has emerged as a major concern within global food supply chains as evidenced by the increasing number of food recalls (Maberry, 2017). Parties in food supply chains have found themselves under greater pressure to provide visibility into their supply chains to allow tracking and tracing of food from "farm to fork." Food supply chains can utilize blockchain technologies to gain visibility and food safety provenance (Rogers, 2017). Figure 4 shows an example of a food supply chain with a fruit grower, an exporter, and a retailer as three parties.

In this realistic scenario, a retailer, say, *Best Fruit* Inc., places a supply order for apples with an exporter named *Fresh Export*. The exporter, who also plays the role of distributor, accordingly places a purchase order with a fruit grower, *ABC Farm*. The grower harvests fruit and transports it in boxes to *Fresh Export*. Upon receiving fruit shipments from the grower, a packing service contracted by Fresh Export washes, sorts, and packs the fruits into small packages, say, 5 lb. bags, each

Figure 4: Blockchain Framework for a Food Supply Chain.



labeled with the Best Fruit brand. The packaged fruit is then delivered to Best Fruit to fulfill the retailer's supply order. For simplicity, the subsequent payment process is not described here. This can be performed using conventional means like bank transfer or credit card payment, or through bitcoin.

As shown in Figure 4, the highlighted activities, such as *Place Supply Order*, *Ship*, and *Receive*, involve multiple parties. For example, Best Fruit can engage with Fresh Export over a private channel *C1*. Once *C1* is created, the participants can use it to synchronize orders. Moreover, in general, a *data object* such as a supply order is called an *asset* in blockchain terminology. This asset contains attribute fields for buyer/seller information, order details, shipments, and receipts that will be filled with actual data as order fulfillment progresses. Each asset has a unique identifier and is realized simply as an attribute-value pair representing attribute names and their corresponding values. Thus, the blockchain ledger maintains the current state of each asset continuously as transactions act upon it. Further, information about the assets can be obtained through a query and retrieval interface.

In addition to using channel *C1* with Best Fruit, Fresh Export can also engage with, say, ABC Farm for purchase of bulk fruit over another private channel *C2* as shown in Figure 4 to protect their business transactions from unauthorized access by other parties such as Best Fruit. Finally, a third channel *C3* is created for all three parties to trace the movement of fruit for supply chain visibility and food safety provenance. For this purpose, an asset type called *Fruit Movement* can

Figure 5: Example of a smart contract written in GO language: Ship supply order.

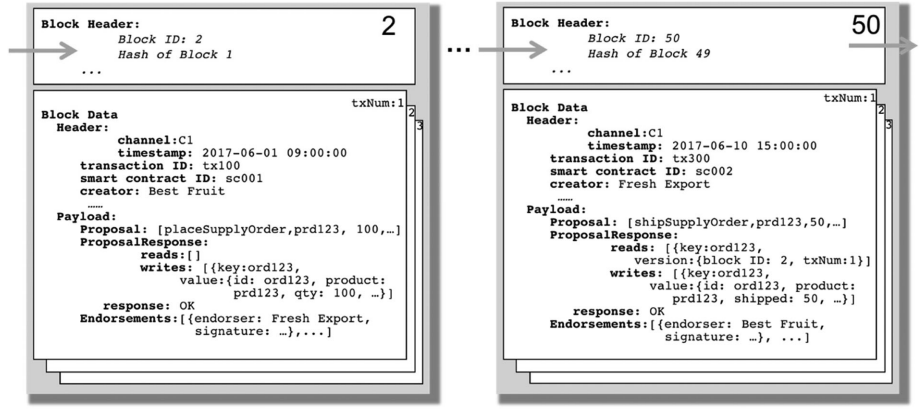
```
func shipSupplyOrder(stub shim.ChaincodeStubInterface, args []string) (string, error) {
        // Get user input arguments
        orderID := args[0]
        shipment := Shipment{"ShipDate":args[1],
                                    "Product":args[2], "Qty":args[3], ……}

        ……
        // Retrieve supply order from ledger as byte array by orderID
        orderAsBytes, err := stub.GetState(orderID)
        // Logic to validate shipment details against supply order
        // e.g., supply order exists, total shipped qty < ordered qty, etc.
        ……
    // If validation is passed, append new shipment to supply order
  // initiate a new object using SupplyOrder structure
        ……
        // Copy retrieved supply order to the new object
        ……
        //Write shipped order into the ledger
        err = stub.PutState(orderID, updatedOrder)
        ……
        // Return success if everything is OK
        return shim.Success([] byte("OK")
                                        }
```

be held on this shared channel and each party can post fruit movement records to the channel by invoking another smart contract. When fruit is harvested by ABC Farm, the farm posts harvested fruit records with a unique asset identifier, along with the harvest date and other information such as the field, production batch, and so on. When the harvested fruit is shipped from ABC Farm to Fresh Export, the farm invokes another smart contract *Ship Harvest Fruit* to update Fresh Export's *Purchase Order* with the shipment information on channel *C2*, and posts the shipment record as a new *Fruit Movement* asset to channel *C3*. This invocation requires endorsement by Fresh Export. As the fruit moves downstream in the supply chain, each movement is recorded in the "public" channel *C3*. Within channel C3, each participant can trace any unit of fruit up or downstream. In addition, an outside "third party" may be added to C3 as an "observer." This party would be dedicated to auditing, food safety tracking, and providing more advanced analytics for the efficiency, waste, energy consumption, and so on of the entire food supply chain.

The various activities such as *Place Supply Order*, *Ship*, and *Receive* can be realized as *smart contracts* that create and update assets such as *Retailer Supply Order*. For example, a smart contract Supply Order Shipment can be implemented as a piece of Go code that appends new shipments to an existing supply order. A snippet of the code is given in Figure 5. This smart contract is implemented

Figure 6: Two blocks in a blockchain showing transactions.



as a function *shipSupplyOrder* with the standard Hyperledger chaincode interface (*stub*) passed as an argument along with the user's input arguments (*args*).

In the first step, this function receives input arguments from the user such as an order ID (*orderID*) and shipment details (*shipment*) in the right format. Then the supply order with the order ID is retrieved from the ledger through the chaincode interface (*stub.GetState*). More business logic can also be implemented here, e.g., to ensure that the supply order is valid, the shipped products are correct and the shipped quantity matches the ordered quantity. Once this validation passes, the content of the *shipment* variable is appended to a copy of the supply order (s*hippedOrder*) and this copy is written to the ledger through the chaincode interface (*stub.PutState*).

This smart contract is deployed at both the retailer and the exporter peers with an endorsement policy. A sample endorsement policy can be "Endorse(Retailer AND Exporter)", which means invocation of this smart contract *shipSupplyOrder* requires valid signatures from both parties. Hyperledger provides support for peers to invoke deployed smart contracts. Thus, the exporter would invoke this smart contract by sending a proposal to the retailer. In turn, the retailer would execute the smart contract (i.e., validate it by simulation) and produce an endorsement signature if successful. Once the invocation is complete, a transaction composed of the proposal and the endorsement is written into a block.

An example of a simplified block is shown in Figure 6. Each block has a block header section and a block data section. The header section contains the block ID, the hash of the previous block, and other block information called metadata. The block data section itself contains a list of transactions. Each transaction also has header and payload sections. In the sample transaction shown in Block 50, the header includes attributes for channel, transaction timestamp, transaction ID, smart contract ID, creator identity and signature, and so on. In the payload, the proposal that Fresh Export sent to Best Fruit is recorded as a set of parameters to invoke

the smart contract *shipSupplyOrder*, including product ID (prd123), quantity (50), and so on. The proposal response from Best Fruit to Fresh Export is also recorded. Here, the proposal is approved as indicated by the "OK" response. Finally, the payload also contains endorsements including endorser names and signatures.

In summary, this use case illustrates how the blockchain technology infrastructure can realize an interorganizational process using smart contracts. First, the interorganizational process needs to be implemented as smart contracts (see Figure 5) installed on the participating parties with appropriate endorsement policies. Second, the architectural design of the blockchain system (see Figure 4), particularly the channel configuration, is a key determinant of transaction privacy and visibility. Finally, transaction data are stored as blocks (Figure 6) on every participating party or its peer node. The cost of network traffic, processing, and storage naturally grow steeply as the number of participating peers and on-chain content increase. With this background, we now turn to understand several design issues that are germane to actually building a blockchain-based system.

BLOCKCHAIN SYSTEM DESIGN: CHALLENGES AND RESEARCH OPPORTUNITIES

A company wishing to adopt blockchain technology for, say, the food supply chain discussed above must address several key issues and challenges. These relate to the design of smart contracts and endorsement policies, data management, and channel configuration. Table 4 gives an overview of these issues and each one is discussed next.

Smart Contracts: Design and Verification

A smart contract was defined as "a computerized transaction protocol that executes the terms of a contract," and it was suggested that smart contracts can be created by translating contractual clauses (Christidis & Devetsikiotis, 2016). A smart contract is simply computer code running on peers in a network, and is similar to stored procedures in a database system. In Hyperledger Blockchain (Parzygnat & Thibeau, 2018), business transactions are executed by invoking smart contracts that are encoded using programming languages such as Java or Go. The actual code of the smart contracts encapsulates business logic that includes process models, rules for internal operational control, rules governing transactions between organizations, and regulatory policies and industry standards.

One area where smart contracts are likely to play a key role is the legal domain (Clack, Bakshi, & Braine, 2016). Many contracts in supply chain or in financial trading applications are quite standard. Hence, an efficient way to facilitate smart contracts is through standardized templates that can be easily customized for various situations by providing specific parameter values. Smart legal templates can facilitate smart contracts, connect legal agreements to automated business logic, simplify processes, drive standards adoption via reusable templates, and reduce costs through the use of common components (Clack et al., 2016; Idelberger, Governatori, Riveret, & Sartor, 2016). Design of industry-specific generic templates is

Table 4: Blockchain system configuration.

Design Decision	Description	Example for the food supply chain use case	SCM Implication
Smart contracts and endorsement policies	A smart contract encapsulates the business logic of the internal operational policies and rules, and any regulatory policies and standards.	See Figure 5 for an example of a smart contract	Interorganizational process transformation and integration
	Smart contracts are enacted by endorsement policies, which specify a minimum number of endorsing peers, a percentage of endorsing peers, or all the endorsing peers in a channel.	Endorsement: "Fresh Export and Best Fruits"	Integrity of transaction versus cost tradeoff
Data management and design	Transactions contain states of assets (or data objects). Thus, the amount of information in assets determines the data volume transmitted through the network and stored at peers.	A supply order contains ordered items (ID, quantity, etc.), delivered items, and other attributes.	Storage and processing cost is higher
Network design and channel configuration	Channels are used to ensure transaction privacy. A channel configuration directly determines data flows in the network.	Bilateral channels *C1* and *C2;* and multi-lateral channel *C3*	Privacy versus visibility tradeoff

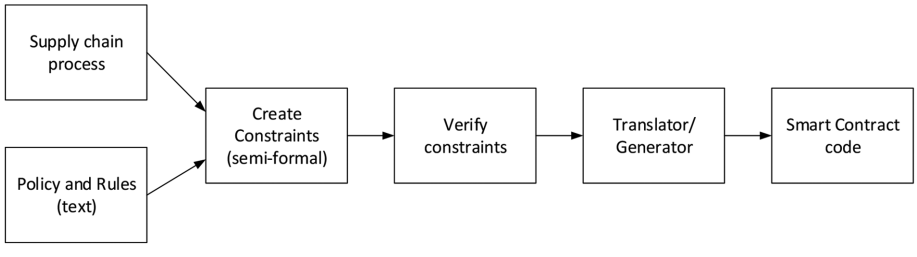
a topic for further research. Next, we illustrate one approach for producing smart contracts directly from the business logic.

A contract on paper consists of various clauses related to pricing, quantity flexibility, minimum purchase commitment, payment terms, lead time, quality, periodicity of ordering, contract duration, information sharing, etc. Some clauses that commonly appear in a contract are listed in Table 5. These contractual clauses can be specified by a textual language to describe constraints for a data model (Cabot, Clarisó, & Riera, 2007). Constraints can be expressed in a semi-formal English-like language to model them more precisely while remaining implementation-independent. Later, these constraints can be translated into a more formal language

Table 5: Representations of constraints in the smart contract.

#	Contractual Clause	Representation of the clause as business rules
1.	The delivered quantity must be within a 3% margin of the ordered quantity.	Shipment.Delivery.Delivery_qty $\leq$ 1.03*Self.Quantity and Shipment.Delivery.Delivery_qty $\geq$ 0.97*Self.Quantity
2.	Actual price must not exceed the quoted price by 2%.	Invoice.Price $\leq$ 1.02$*$Self.Price
3.	Total order must be delivered within two shipments.	Shipment.Delivery.Size $\leq$ 2
4.	For payment within 15 days of the invoice date, a 5% discount applies.	if (Payment.Payment_date – Invoice_date) $\leq$ 15 then Early_pay_discount = 0.05* Self.Invoice.Invoice_amount

Figure 7: Translating a process model into a smart contract.



like SQL for implementation, or into other code such as Java or Python. They can also be converted into code for a smart contract.

Table 5 shows some sample contractual clauses and constraint conditions written semi-formally. Automated techniques for converting from natural language to a formal constraint syntax called OCL (Object Constraint Language) are discussed in (Bajwa, Bordbar, & Lee, 2010). Once we have the constraints in a formal syntax, it is also possible to check them for consistency, completeness, and correctness using software tools (Ziemann & Gogolla, 2003).

A blockchain ledger is essentially a record of transactions in chronological order. Implementing a process on a blockchain system requires mapping, say, a supply chain process into a series of successive transactions, which are executed in an ordered sequence. Our proposal is to perform this translation following the steps shown in Figure 7. Based on a process description and the accompanying rules and policies, a set of constraints is created in a semiformal language. These constraints are verified for correctness and completeness, and then fed into a Translator/Generator that creates the smart contract code for them. In addition to specifying contracts, it is necessary to ensure a smooth connection between a

party's internal business processes and smart contracts that execute on a blockchain system. This can be done through events. When a transaction occurs, events are emitted from the blockchain system and relevant parties listen to these events to keep their internal systems synchronized with the data on the blockchain ledger.

Although intuitive, operationalizing this approach is fraught with challenges. In particular, the wording of regulatory policies in plain text can be rather convoluted and such policies do not lend themselves easily to mapping into an executable language with the correct semantics. To illustrate, consider a sample clause in the purchase agreement between Fresh Export and Best Fruit: "in case of an unforeseen situation, such as drought, excessive rain, fire, war, or for any other reasonable cause that is beyond the parties' control, neither party shall be considered liable for losses and damages." First, this clause contains "desired ambiguities" (Mik, 2017), such as "reasonable cause," which leave parties the flexibility to work out a resolution when an unanticipated event occurs. However, such a clause can hardly be represented formally or measured objectively to translate accurately into smart contract code. Second, executing contracts often requires information from third parties. For example, only an independent authority may be able to evaluate the condition for "excessive rain." However, to ensure determinism, that is, for every party to reach the same conclusion after execution, a smart contract is unable to retrieve off-chain resources but can request a trusted third party called an "oracle" to push necessary resources onto the blockchain system. The dependency on oracles for off-chain data raises risks of erroneous data and weakens the benefits of decentralization. Another major concern lies in the need to confirm that smart contract code reflects the intent of parties and contains no inadvertent coding errors. However, it is practically impossible to ensure that the software code is bug-free. Hence, risk is not entirely eliminated (Mik, 2017). This is clearly a limitation of blockchain technology.

Blockchain has the potential to create decentralized global platforms to support global supply chains. Therefore, within a global supply chain, a smart contract may be executed by globally distributed parties. Within the current US legal framework, smart contracts are enforceable under contract law (Mik, 2017). However, if a smart contract is executed by parties across jurisdictions, an overarching governing law is essential to determine what specific law will apply for the interpretation of the smart contract and which jurisdiction will adjudicate disputes (Levi & Lipton, 2018). Finally, there are even more complicated issues related to the interpretation of legal prose in smart contracts as discussed by Mik (2017). Given the early stage of smart contract adoption, most smart contracts are used mainly to execute simple transactions, such as payment instructions. Legal experts have also suggested that smart contracts can be used as ancillary to traditional text-based contracts (Levi & Lipton, 2018).

In addition to the challenge of designing smart contracts for complicated business logic, endorsement policies that define how smart contracts are enacted also warrant careful consideration. An endorsement policy applies to each channel and determines how many peers are required for transaction endorsement. It could be specified as a minimum number of endorsing peers, a percentage of endorsing peers, or all the peers in a channel. In general, more endorsements are a sign of stronger agreement and reflect more confidence in the integrity of the transaction.

Further, if a transaction requires endorsements from M parties, an endorsement policy can either state M specific parties or it can ask for any M out of N parties to endorse the transaction. In the latter case, once M endorsements are received for a transaction, it can continue without waiting for further endorsements. While offering some flexibility, this policy also increases the data flow because the transaction request is sent to *all* N parties for endorsement in the first place. This leads to a tradeoff between the number of messages exchanged and the strength of the consensus. Clearly, stronger consensus requires exchange of more messages. Thus, appropriate methods are needed to evaluate endorsement policies to ensure transaction integrity and blockchain system performance.

Data Management and Design

In addition to smart contracts, another key component of a blockchain system is asset and transaction data. A blockchain system is primarily a reliable, immutable distributed data storage mechanism. Thus, data management is a critical design consideration because it has considerable impact on the performance and cost of the blockchain operation. In this section, we briefly highlight the key issues related to data management.

The assets in blockchain applications can vary from business documents (e.g., supply orders, invoice) to images of products and objects, and even smart objects (or virtual avatars) representing physical entities in the real world. The images are often very large. Since blockchain data are replicated across all the nodes in a network, data requirements can become enormous. A typical 10 MB image is small, yet when replicated on 100 nodes it would consume 1 GB of storage space. A 1 GB high-resolution image replicated on 1000 nodes would require 1 TB of space. The actual cost of storage consists not only of the physical storage cost but also the networking and processing costs. Say, there are n nodes in a blockchain network, and an endorsement policy requires that e out of n nodes must endorse this transaction. Hence, the total number of messages exchanged before a data object is committed to the blockchain is:

- $2e$ messages for endorsement requests to and replies from the endorsing parties;
- 2 messages (one to the ordering service and one for its reply);
- n messages from the ordering service to all n nodes with the new block to be committed.

This results in a total of $n + 2e + 2$ messages in the normal case. If there is a problem at commitment time, further messages will be exchanged to invalidate the block. In addition to the network traffic, computing capacity is also consumed at each stage of the process. These costs must all be included in the imputed storage cost. Precise calculations of the cost of blockchain storage are not available. However, some informal estimates place the cost of 1KB of Ethereum storage at $1.58 (Ethereum, 2018). Even though current estimates are rough, they nevertheless suggest that the cost of storage on a public blockchain platform can be staggering, a few thousand times higher than on a distributed database system or in the cloud. On a permissioned blockchain system, the cost is likely to be less but still one or two orders of magnitude higher. In addition, response times, both

for writing and reading data, on a blockchain system are again much slower than for a conventional system. This goes to show that blockchain storage is a precious commodity.

Hence, an important challenge lies in deciding what data should be stored on the blockchain system (on-chain data) and what should be kept off-chain. One way to analyze this problem in the context of a supply chains is to break down the data in terms of "material facts." Material facts pertain to critical data such as product number, quantity, order date, delivery date, quoted price, shipping notice, and so on. Such information should be on-chain. In other words, material facts are information that would be essential for dispute resolution among supply chain partners. Additional information such as the image of the product and so on is less important and may be stored off-chain. Of course, these decisions must be made on a case-by-case basis but a few initial solutions have been proposed to manage the huge amount of data and minimize what is stored on-chain.

One solution is to decompose assets into components, and store only the changes (or deltas) in data value in the transactions. However, a disadvantage of this approach is that most blocks do not contain a complete snapshot of digital assets at a specific time and it can become very hard to reconstruct it from history. Another option is to offload large images and data objects to an "off-chain document repository" and store only a hash value of each such document on the ledger so that its integrity may be verified (Ben-Ari, 2017). Although the hash of off-chain data can detect any alteration to the data object, it is more susceptible to deletion or loss, and thus less durable.

Channel Configuration and its Impact on Privacy and Visibility

We discussed the significance of channels above. If there is only one channel in a blockchain network, then all transactions would be posted to that channel and all parties would normally have access to all transactions unless their data payload is encrypted, and the cryptographic keys are shared only among select partners. An alternative mechanism is to separate the interactions among subsets of parties by creating multiple channels. Of course, additional channels increase complexity of the network design.

If multiple parties collaborate on a business process, there are many ways to configure blockchain channels. In the fruit supply chain example, *C1* and *C2* are bilateral channels. If a party, say Fresh Export, partners with many retailers or farms, a channel would be created for each partner, leading to a proliferation of channels. Managing these channels can be an arduous task as each channel has its own database and cryptographic keys. It is also possible to reconfigure these channels in other ways. In general, channel configuration can become complex, especially when secrecy and privacy considerations arise.

Channel configuration has a substantial impact on the data volume over the network, workload distribution, network resilience, and, most importantly, visibility. In the context of supply chains, there are growing calls for higher upstream and downstream visibility by using new technologies to provide provenance data to the marketplace enabling customers to have easy access to such information (New, 2010). Visibility can reinforce the customer–supplier bond by erasing the

costs and delays implicit in traditional arrangements (Lamming, Caldwell, Harrison, & Phillips, 2001). Sharing data regarding current order and production status as well as plans and forecasts with the various supply chain partners involved enables better decision making in SCM (Gavirneni, Kapuscinski, & Tayur, 1999; Lee & Whang, 1999). Most importantly, visibility helps establish trust, which has been well studied in the organizational management literature (Parkhe, 1993; Korsgaard, Schweiger, & Sapienza, 1995; Young-Ybarra & Wiersema, 1999). While collaboration in a supply chain network promotes a higher level of visibility, it also further intensifies competition among partners (Wilhelm, 2011). This in turn can deter parties from joining a blockchain network, and thus constrain the growth of the network and diminish the network effect. Hence, channel configuration must strike the right balance between visibility and competitive edge.

ASSESSMENT AND IMPLICATIONS FOR BLOCKCHAIN TECHNOLOGY

While blockchain is a very promising technology with considerable potential, it is also a high overhead technology because it is based on the principle of replication and redundancy of data storage and processing. Numerous implications in terms of storage, processing and networking costs must be carefully considered, not to speak of its environmental impact since it dramatically increases the consumption of energy and hence natural resources.

Therefore, it is evident to us that blockchain belongs only in applications where the additional cost of running a transaction can be justified by the aggregate benefits that accrue after a careful and thorough analysis. It is important to recognize that enterprise resource planning (ERP) systems such as SAP and Oracle store and manage large databases on servers (or in the cloud) that are internal to an organization and protected within its firewalls. Such data is accessible only to the employees of the organization with proper authorization, and, for the most part, does not belong in a blockchain ledger. An exception can be made when there are regulatory requirements for maintaining provenance information about, say, a manufacturing process in the pharmaceutical or high technology industry. Then, a firm may consider blockchain technology to document the actual process that leads to the production of a batch.

Interorganizational transactions as in a supply chain make blockchain attractive because data are exchanged with outside parties. However, here again consider a manufacturer such as GM that has a well-established relationship going back 50 years with a supplier like Johnson Controls. Naturally, the two parties perceive each other as trustworthy, and are willing to forgo the risk of opportunistic behavior (Wang & Wei, 2007). Consequently, there is little incentive for either party to move to a blockchain platform.

Previous work has summarized a number of strengths of blockchain, for example, visibility, traceability, compliance, and resiliency (Staples et al., 2017; Subramanian & Hemang, 2017; Babich & Hilary, 2018; Pawczuk, 2018). We also studied several use cases to understand value propositions of blockchain in practice. The main insights are distilled in Table 6 as the strategic considerations for

Table 6: An illustrative approach to assess value of blockchain technology in different industries.

	Strategic considerations					
Use Cases	Trust Level	Need for Traceability	Need for Visibility	Need for Privacy	Cost focus	Use Case Summary
Food (e.g., Fresh produce)	Low	High	High	–	High	A large number of parties that may lack trust; visibility and traceability required to ensure food safety; pressure to keep implementation costs low (Nunes, 2018)
Luxury (e.g., jewelry)	Low	High	High	–	–	Provenance is needed for authentication and regulatory compliance at all stages of the global supply chain involving many parties who may not trust each other (Bitcoin Exchange Guide, 2018)
Manufacturing (e.g., aircraft parts)	–	High	–	High	High	Tens of thousands of aircraft parts need to be tracked reliably to ensure aircraft safety, and to meet regulatory demands at a reduced operational cost (Bryan, 2018)
Pharma (e.g., drugs)	–	High	High	High	–	Provenance is needed to prevent counterfeit drugs and medical devices; real-time tracking of temperature, humidity and other factors during drug handling, transport and storage can ensure the quality and efficacy of drugs (Ram, 2018)
Third-party Logistics (e.g., Maersk)	Low	–	High	High	High	The global shipping ecosystem calls for an efficient and secure platform to manage paperwork and streamline operations (Moise & Chopping, 2018)

assessing the value of blockchain technology along several aspects relevant to this technology. We illustrate these considerations using notable use cases from supply chains in different industries. Each use case emphasizes a few key aspects that are especially pertinent to its underlying supply chain or its business characteristics. The entries in the table are either "low" or "high," while a blank indicates that the aspect is not addressed for this case. A more granular scale may also be employed.

Table 6 suggests that one important determinant of whether blockchain technology is appropriate is the level of trust in a supply chain. The general literature on buyer–supplier relationships regards the establishment of trust as a key objective when traceability and visibility are difficult to achieve (Johnston, McCutcheon, Stuart, & Kerwood, 2004). Therefore, in supply chains where the parties have a low level of trust, the main contribution of blockchain technology is to break the physical and temporal barriers to provide effective and efficient visibility. It could potentially enhance visibility across the supply chain by eliminating information distortions, and increase information velocity by reducing delays. As a result, blockchain can be considered as an equitable relationship-specific investment to grow mutual trust (Klein, Rai, & Straub, 2007). For example, Everledger (Nofer, Gomber, Hinz, & Schiereck, 2017), a blockchain system for tracking high-value goods (e.g., diamonds), can provide trading partners complete visibility of transaction history, even though they may not know each other well. The existence of counterfeit products is another issue related to trust and the suitability of blockchain in this context is analyzed in (Pun, Swaminathan, & Hou, 2018).

A second key determinant of the value of blockchain technology is the need for provenance or traceability (see proposals for adapting blockchain designs for provenance in Lu & Xu, 2017; Kim & Laskowski, 2018). As noted earlier, traceability is a unique and inherent feature of blockchain technology. Since the distributed ledger is a shared medium among all partners, all information that relates to provenance can be stored on the ledger. Then an independent third party would be able to check it to verify compliance. This is particularly relevant in the context of regulatory requirements as in the case of the 2013 Drug Supply Chain Security Act (DSCSA) being implemented in a phased manner. This act places onerous requirements on the partners in the drug supply chain such as manufacturers, distributors, and retailers to implement, first lot-level, and, eventually item-level, serializability in the drug supply chain. This would eliminate counterfeit products from the supply chain through an immediate detection and removal capability. Similarly, such provenance information is also required in the context of sustainability initiatives, environmental regulations, and laws that prohibit use of child labor in the textile supply chain (New, 2010).

A third factor is visibility. This is slightly different from provenance in that it refers to information about the operational aspects of the supply chain for the smooth flow of goods. By knowing the exact status of an order or shipment in the supply chain, a partner can better cope with unexpected situations such as breakdowns, delays, bottlenecks, and so on, proactively. A fourth factor relates to privacy that conflicts with visibility. Hence, there is a tradeoff between these two factors since higher visibility usually leads to a sacrifice of privacy. Resilience of information is yet another factor not shown in Table 6. High resilience would imply that all information about the supply chain is always available with few or no outages.

In fact, this feature follows naturally from the redundancy built into the blockchain architecture. A number of blockchain initiatives were motivated by the need for visibility and resilience. For example, IBM Global Finance tested blockchain to manage business documents with its corporate clients to increase transaction visibility and thus reduce disputes (IBM Institute for Business Value, 2016). The London and Australian Stock Exchanges have also explored blockchain-based platforms for security settlement with their banking clients (IBM Institute for Business Value, 2016) for more transparent, efficient, and resilient settlement processes than traditional centralized platforms can offer.

Finally, cost is of course a major concern in blockchain adoption decisions. In some industries such as groceries or commodities, cost may be the main competitive lever as compared with other industries like high-tech where product features are more important, margins are high and there is less focus on cost. An excessive cost focus would deter the use of blockchain unless all the rival firms follow suit and drive up the industry-wide cost structure. One might also argue that the cost of implementing blockchain is outweighed by its benefits such as streamlined operations, reduced risks and compliance cost, and so on. For example, blockchain is an attractive technology for airlines to improve parts management and reduce operational cost.

Similarly, Maersk (Moise & Chopping, 2018) also believes blockchain has the potential to help manage paperwork efficiently and streamline its global trade operations. Fresh produce supply chains are often under pressure to keep prices low. Blockchain can provide supply chain traceability and visibility so that food safety issues can be quickly identified. Such a benefit can offset the cost of adopting blockchain. Moreover, parties may need to be properly incentivized to adopt blockchain. As also observed by Babich and Hilary (2018), in a supply chain, information sharing among partners is highly desirable. Such sharing can reduce over-ordering from retailers and improve the efficiency of the entire supply chain. Moreover, for provenance or traceability, retailers may require suppliers to share product quality information, such as data regarding fertilizer use, pesticide use, and storage control of crops, further increasing the burden on suppliers. Thus, unless suppliers can be appropriately compensated, they have little incentive to join a blockchain network with retailers.

A firm should carefully consider these factors in concert while deciding whether blockchain is the right technology for them. Hence, we conjecture that in some cases conventional technologies such as EDI or SOA would still prevail, given high trust among parties, and low traceability and visibility needs for products or services, while in others a hybrid use of conventional and blockchain technologies would win. Various types of hybrid architectures are likely to emerge. In some scenarios, partners will only keep on the blockchain information that ensures smooth operation of the supply chain and satisfies any regulations. At the same time, they will keep most of their internal information private. Of course, individual firms within each industry will have to use this framework in their own context and in the light of the technological, competitive, and regulatory landscape in which they operate.

We expect the development of blockchain technology to be driven largely at the industry level where industry consortia will create blueprints for the adoption

of this technology by the member firms. IT firms will then develop the infrastructure, platforms and products for rolling out the technology within target industry segments. At this stage, it is difficult to predict timelines for how use of this technology will unfold.

Another key implementation challenge relates to "linking the chains." Different partners in a supply chain may join different blockchain platforms or consortia. Consortia that use different shared ledger technologies must be integrated across a single value chain using common standards and protocols similar to the ones for the Internet (Piscini, Dalal, Mapgaonkar, & Santhana, 2017). Such a multichain integration capability is necessary not only within a country, but more so at a global level where standards are even more likely to differ. Progress in this direction is still in infancy and much more work is necessary. In a recent initiative, the Monetary Authority of Singapore (MAS) and the Hong Kong Monetary Authority (HKMA) are collaborating to test distributed ledger technology in the multitrillion dollar global trade finance industry to develop an infrastructure to facilitate cross-border transactions (RBC Investor & Treasury Services, 2018). Their goal is to reduce the risk of human error and fraud, and speed up integration across markets in Asia.

CONCLUSIONS

Blockchain is a nascent technology that allows business transactions among multiple parties to be recorded in a reliable, immutable and secure manner. Currently, the largest applications of this technology lie in payment systems such as bitcoin. However, it potentially represents a major paradigm shift in terms of how business transactions are conducted, and thus has far-reaching implications. Blockchains serve as shared, indisputable records of all transactions that take place among multiple partners in accordance with their established contracts. They create trustworthy networks by enabling nontrusting parties to transact with each other. In this sense, they are a further step toward disintermediation. The roots of blockchain technology lie in other fundamental underlying technologies such as database systems, public key cryptography, smart contracts, and consensus protocols. To date, many application prototypes and products have been developed to commercialize this technology. However, it is still far from going mainstream.

A blockchain serves as an audit trail to verify provenance, say, in the food or the drug supply chains. Hence, it will discourage litigation because all the business partners have equal access to the facts. Of course, in international supply chains, blockchain technology will help only if all the countries have a robust functioning legal system and are willing to enforce the laws. Our focus in this article is mainly on B2B applications, but we expect that B2C and C2C applications will also emerge in a significant way. Thus, in a C2C scenario, a taxi or lodging services provider will be able to connect directly with customers without the need for an intermediary, like Uber or Airbnb. This is another step toward disintermediation that will fully unleash the power of network effects (Tapscott & Tapscott, 2016).

Our analysis in this article suggests that blockchain technology should be deployed selectively, mainly for interorganizational transactions among untrusted parties, and in applications that need high levels of provenance and visibility. It is not a silver bullet for all applications. It also incurs a very high overhead

in terms of storage, networking, and processing costs that can be justified only after a thorough case-by-case analysis. Neither these costs nor their environmental implications have been explored at any length. Hence, the economics of blockchain technology must be studied in more depth (Davidson, De Filippi, & Potts, 2016; Pun et al., 2018). Moreover, the impact of different information sharing arrangements on blockchain system design needs to be investigated further along the lines of (Chang, Katehakis, Melamed, & Shi, 2018).

The idea of cryptocurrency tokens as a "programmable currency unit that is bolted to a blockchain" has given rise to an area called Tokenomics (Mougayar, 2017). Apart from use as cyptocurrencies, tokens can be generalized to represent assets, utility, or a claim, and used for various other purposes as equity tokens, funding tokens, consensus tokens, work tokens, voting tokens, asset tokens, and so on. (Oliveira, Zavolokina, Bauer, & Schwabe, 2018). Since tokens will serve as a basic means of value exchange, developments in blockchain technology will be influenced by progress in frameworks and tools for token design.

We posed a number of research questions particularly related to smart contracts, blockchain data, and network design issues, and proposed possible directions to address these challenges. We see this as a call for information systems and operations management researchers to take a lead role from a variety of perspectives in realizing the full potential of blockchains for transforming supply chain management.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Online Appendix.

REFERENCES

Babich, V. R., & Hilary, G. (2018). Distributed ledgers and operations: What operations management researchers should know about blockchain technology. *Manufacturing & Service Operations Management*, Forthcoming.

Bajwa, I. S., Bordbar, B., & Lee, M. G. (2010). OCL constraints generation from natural language specification. *Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference*. Vitoria, Brazil: IEEE, 204–213.

Ben-Ari, A. (2017). Outstanding challenges in blockchain technology in 2017, available at https://appliedblockchain.com/outstanding-challenges-in-blockchain-2017/

Bitcoin Exchange Guide (2018). How the jewelry industry can apply blockchain distributed ledger technology, available at https://bitcoinexchangeguide.com/how-the-jewelry-industry-can-apply-blockchain-distributed-ledger-technology/

Bryan, V. (2018). Aerospace suppliers look to blockchain for parts tracking, available at https://www.reuters.com/article/us-aerospace-blockchain/aerospace-suppliers-look-to-blockchain-for-parts-tracking-idUSKBN1I32AW

Buterin, V. (2013). Ethereum: A next-generation smart contract and decentralized application platform, available at https://github.com/ethereum/wiki/wiki/White-Paper

Cabot, J., Clarisó, R., & Riera, D. (2007). UMLtoCSP: A tool for the formal verification of UML/OCL models using constraint programming. *Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering*. New York, NY: ACM Press, 547–548.

Cachin, C. (2016). Architecture of the Hyperledger Blockchain Fabric. *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 1–4.

Chang, J., Katehakis, M. N., Melamed, B., & Shi, J. (2018). Blockchain design for supply chain management. *SSRN Electronic Journal*, *3295440*, 1–35.

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. *IEEE Access*, *4*, 2292–2303.

Clack, C. D., Bakshi, V. A., & Braine, L. (2016). Smart contract templates: Essential requirements and design options. *arXiv preprint, 1612.04496*, 1–15.

Davidson, S., De Filippi, P., & Potts, J. (2016). Economics of blockchain. *SSRN Electronic Journal*, *2744751*, 1–23.

de Kruijff, J., & Weigand, H. (2017). Understanding the blockchain using enterprise ontology. *Proceedings of the 29th International Conference on Advanced Information Systems Engineering*. Berlin, Germany: Springer, 29–43.

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K.-L. (2017). BLOCKBENCH: A framework for analyzing private blockchains. *Proceedings of the 2017 ACM International Conference on Management of Data*. New York, NY: ACM Press, 1085–1100.

Ekblaw, A., Azaria, A., Halamka, J. D., & Lippman, A. (2016). A case study for blockchain in healthcare: "MedRec" prototype for electronic health records and medical research data. *Proceedings of the 2nd International Conference on Open and Big Data*. New York, NY: ACM Press, 1–13.

Ethereum (2018). How can I estimate price of data storage?, available at https://ethereum.stackexchange.com/questions/40944/how-can-i-estimate-price-of-data-storage

Gartner (2017). Seven things that supply chain leaders need to know about blockchain, available at https://www.gartner.com/doc/3620517/seven-things-supply-chain-leaders

Gavirneni, S., Kapuscinski, R., & Tayur, S. (1999). Value of information in capacitated supply chains. *Management Science*, *45*(1), 16–24.

Genkin, D., Papadopoulos, D., & Papamanthou, C. (2018). Privacy in decentralized cryptocurrencies. *Communications of the ACM*, *61*(6), 78–88.

Hart, P., & Saunders, C. (1997). Power and trust: Critical factors in the adoption and use of electronic data interchange. *Organization Science*, *8*(1), 23–42.

IBM Institute for Business Value (2016). Fast forward: Rethinking enterprises, ecosystems and economies with blockchains, available at https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=GBE03757USEN

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016). Evaluation of logic-based smart contracts for blockchain systems. *Proceedings of the 10th International Symposium on Rules and Rule Markup Languages for the Semantic Web*. Berlin, Germany: Springer, 167–183.

Johnston, D. A., McCutcheon, D. M., Stuart, F. I., & Kerwood, H. (2004). Effects of supplier trust on performance of cooperative supplier relationships. *Journal of Operations Management*, *22*(1), 23–38.

Kim, H. M., & Laskowski, M. (2018). Toward an ontology-driven blockchain design for supply-chain provenance. *Intelligent Systems in Accounting, Finance and Management*, *25*(1), 18–27.

Klein, R., Rai, A., & Straub, D. W. (2007). Competitive and cooperative positioning in supply chain logistics relationships. *Decision Sciences*, *38*(4), 611–646.

Korsgaard, M. A., Schweiger, D. M., & Sapienza, H. J. (1995). Building commitment, attachment, and trust in strategic decision-making teams: The role of procedural justice. *Academy of Management Journal*, *38*(1), 60–84.

Lamming, R. C., Caldwell, N. D., Harrison, D. A., & Phillips, W. (2001). Transparency in supply relationships: Concept and practice. *The Journal of Supply Chain Management*, *37*(4), 4–10.

Lannquist, A. (2018). Blockchain in enterprise: How companies are using blockchain today, available at https://blockchainatberkeley.blog/a-snapshot-of-blockchain-in-enterprise-d140a511e5fd.

Lee, H., & Whang, S. (1999). Decentralized multi-echelon supply chains: Incentives and information. *Management Science*, *45*(5), 633–640.

Levi, S. D., & Lipton, A. B. (2018). An introduction to smart contracts and their potential and inherent limitations, available at https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/.

Lohade, N. (2017). Dubai aims to be a city built on blockchain. *The Wall Street Journal*, available at https://www.wsj.com/articles/dubai-aims-to-be-a-city-built-on-blockchain-1493086080. April 24.

Löhe, J., & Legner, C. (2010). SOA adoption in business networks: Do service-oriented architectures really advance interorganizational integration? *Electronic Markets*, *20*(3–4), 181–196.

Lu, Q., & Xu, X. (2017). Adaptable blockchain-based systems: A case study for product traceability. *IEEE Software*, *34*(6), 21–27.

Maberry, T. (2017). A look back at 2016 food recalls, available at https://www.foodsafetymagazine.com/enewsletter/a-look-back-at-2016-food-recalls/

Malkhi, D., & Reiter, M. (1998). Byzantine quorum systems. *Distributed Computing*, *11*(4), 203–213.

Mik, E. (2017). Smart contracts: Terminology, technical limitations and real world complexity. *Law, Innovation and Technology*, *9*(2), 269–300.

Miller, A., Xia, Y., Croman, K., Shi, E., & Song, D. (2016). The honey badger of BFT protocols. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY: ACM Press, 31–42.

Moise, I., & Chopping, D. (2018). Maersk and IBM partner on blockchain for global trade. *Wall Street Journal*, available at https://www.wsj.com/articles/maersk-and-ibm-partner-on-blockchain-for-global-trade-1516111543. January 16.

Mougayar, W. (2017). Tokenomics - A business guide to token usage, utility and value, available at https://medium.com/@wmougayar/tokenomics-a-business-guide-to-token-usage-utility-and-value-b19242053416

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system, available at https://bitcoin.org/bitcoin.pdf

New, S. (2010). The transparent supply chain. *Harvard Business Review*, *88*, 1–5.

Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, *59*(3), 183–187.

Nunes, K. (2018). Walmart bringing blockchain to fresh produce, available at https://www.foodbusinessnews.net/articles/12569-walmart-bringing-blockchain-to-fresh-produce

Oliveira, L., Zavolokina, L., Bauer, I., & Schwabe, G. (2018). To token or not to token: Tools for understanding blockchain tokens. *Proceedings of the 2018 International Conference on Information Systems*. San Francisco, CA: ICIS, 1–17.

Parkhe, A. (1993). Strategic alliance structuring: A game theoretic and transaction cost examination of interfirm cooperation. *Academy of Management Journal*, *36*(4), 794–829.

Parzygnat, M., & Thibeau, D. (2018). IBM blockchain 101: Quick-start guide for developers, available at https://developer.ibm.com/tutorials/cl-ibm-blockchain-101-quick-start-guide-for-developers-bluemix-trs/

Pawczuk, L. (2018). When two chains combine: Supply chain meets blockchain, available at https://www2.deloitte.com/tr/en/pages/technology/articles/when-two-chains-combine.html

Pilkington, M. (2016). Blockchain technology: Principles and applications. In F. Xavier Olleros & Majlinda Zhegu (Eds.), *Research Handbook on Digital Transformations*. Cheltenham, UK: Edward Elgar, pp. 225–253.

Piscini, E., Dalal, D., Mapgaonkar, D., & Santhana, P. (2017). Blockchain to blockchains: Broad adoption and integration enter the realm of the possible, available at https://www2.deloitte.com/insights/us/en/focus/tech-trends/2018/blockchain-integration-smart-contracts.html

Pun, H., Swaminathan, J. M., & Hou, P. (2018). Blockchain adoption for combating deceptive counterfeits. *Kenan Institute of Private Enterprise Research Paper*, No. 18-18.

Ram, P. (2018). Top 5 blockchain use cases in pharma and healthcare - that you should know about!, available at https://hackernoon.com/top-5-use-cases-of-blockchain-in-pharma-and-healthcare-that-you-should-know-about-77ccdd76369b

Ratnasingham, P. (1998). Internet-based EDI trust and security. *Information Management & Computer Security*, *6*(1), 33–39.

RBC Investor & Treasury Services (2018). Singapore and Hong Kong's blockchain innovations, available at https://www.rbcits.com/en/insights/2018/03/singapore_and_hong_kongs_blockchain_innovations

Plant, Robert (2017). Can blockchain fix what ails electronic medical records? *The Wall Street Journal*, available at https://blogs.wsj.com/experts/2017/04/27/can-blockchain-fix-what-ails-electronic-medical-records/. April 27.

Rogers, C. (2017). Why marketers need to get to grips with Blockchain, available at https://www.marketingweek.com/2017/04/05/marketers-need-know-blockchain/

Staples, M., Chen, S., Falamaki, S., Ponomarev, A., Rimba, P., Tran, A. B., & Zhu, L. (2017). Risks and opportunities for systems using blockchain and smart contracts. *Data61 (CSIRO)*, Sydney, Australia, available at https://research.csiro.au/data61/wp-content/uploads/sites/85/2016/08/Blockchain-RisksandOpps-PDF.pdf.

Subramanian, H. (2017). Decentralized blockchain-based electronic marketplaces. *Communications of the ACM*, *61*(1), 78–84.

Swan, M. (2015). *Blockchain: Blueprint for a new economy*. Sebastopol, CA: O'Reilly Media, Inc.

Tapscott, D., & Tapscott, A. (2016). *Blockchain revolution: How the technology behind Bitcoin is changing money, business, and the world*. New York: Penguin Random House.

Todd, P. (2015). Ripple protocol consensus algorithm review, available at http://docshare04.docshare.tips/files/27118/271185330.pdf

Underwood, S. (2016). Blockchain beyond Bitcoin. *Communications of the ACM*, *59*(11), 15–17.

Vakharia, A. J. (2002). E-Business and supply chain management. *Decision Sciences*, *33*(4), 495–504.

Wang, E. T. G., & Wei, H.-L. (2007). Interorganizational governance value creation: Coordinating for information visibility and flexibility in supply chains. *Decision Sciences*, *38*(4), 647–674.

Wattenhofer, R., & Foerster, K.-T. (2017). Quorum systems. In Roger Wattenhofer (Ed.), *Distributed Ledger Technology: The Science of the Blockchain* (2nd ed., pp. 87–104). CreateSpace Independent Publishing Platform.

White, A., Daniel, E., Ward, J., & Wilson, H. (2007). The adoption of consortium B2B e-marketplaces: An exploratory study. *The Journal of Strategic Information Systems*, *16*(1), 71–103.

Wilhelm, M. M. (2011). Managing coopetition through horizontal supply chain relations: Linking dyadic and network levels of analysis. *Journal of Operations Management*, *29*(7–8), 663–676.

Young-Ybarra, C., & Wiersema, M. (1999). Strategic flexibility in information technology alliances: The influence of transaction cost economics and social exchange theory. *Organization Science*, *10*(4), 439–459.

Ziemann, P., & Gogolla, M. (2003). Validating OCL specifications with the USE tool: An example based on the BART case study. *Electronic Notes in Theoretical Computer Science*, *80*, 157–169.

Akhil Kumar is a professor of information systems at the Smeal College of Business at Penn State University. He received his Ph.D. from the University of California, Berkeley, and has previously been on the faculties at Cornell University and the University of Colorado, and spent one year as a researcher at Bell Labs. His research interests are in Blockchain, business process management, business analytics, healthcare IT and process mining. He has published more than 100 papers in academic journals and international conference proceedings. His past research has been supported by NSF, IBM, and HP. He has also served on several editorial boards and program committees. He is a senior member of IEEE and a member of ACM.

Rong Liu is an associate professor of information systems in the school of business at Stevens Institute of Technology. Before joining Stevens, she was a research staff member at IBM T. J. Watson Research Center. She received her Ph.D. in business administration from The Pennsylvania State University. Her research interests include blockchain, business process management, natural language processing, and machine learning.

Zhe Shan is currently an assistant professor in the Department of Information Systems and Analytics in the Farmer School of Business at the Miami University. He earned his Ph.D. degree in business administration and operations research from the Smeal College of Business at Penn State University in 2011. His research interests include FinTech Innovation & Blockchain, information security management, patient-centered healthcare, and business process analytics. His work has been published in *MIS Quarterly*, *Journal of Management Information Systems*, and *Journal of Operations Management*, among others.