



Intelligent Smart Contracts for Innovative Supply Chain Management

Paolo Bottoni¹, Nicola Gessa², Gilda Massa², Remo Pareschi^{3*}, Hesham Selim⁴ and Enrico Arcuri²

¹ Department of Computer Science, Sapienza University of Rome, Rome, Italy, ² ENEA, Rome, Italy, ³ Department of Bioscience and Territory, University of Molise, Isernia, Italy, ⁴ Food and Agriculture Organization, Rome, Italy

OPEN ACCESS

Edited by:

Michael Henke,
Fraunhofer Institute for Material Flow
and Logistics, Germany

Reviewed by:

Meghana Kshirsagar,
University of Limerick, Ireland
Francesco Santini,
University of Perugia, Italy

*Correspondence:

Remo Pareschi
remo.pareschi@unimol.it

Specialty section:

This article was submitted to
Non-Financial Blockchain,
a section of the journal
Frontiers in Blockchain

Received: 17 February 2020

Accepted: 02 November 2020

Published: 26 November 2020

Citation:

Bottoni P, Gessa N, Massa G,
Pareschi R, Selim H and Arcuri E
(2020) Intelligent Smart Contracts for
Innovative Supply Chain Management.
Front. Blockchain 3:535787.
doi: 10.3389/fbloc.2020.535787

We propose blockchains and smart contracts as enabling technologies for an innovative type of supply chain management, with the goal of achieving higher levels of collaboration between the companies participating in the chain, which in turn pays in the form of higher levels of profitability and economic health for the participating enterprises. Our proposal goes far beyond simply using blockchains as decentralized systems to track the origin and delivery of goods, which is what most of the current blockchain projects on supply chains are focused on. In fact, we introduce a type of smart contract aimed to solve two of the main problems that hinder the efficiency and effectiveness of supply chains, namely *trust* and *coordination*. Solving the problem of *trust* amounts to the capability of establishing quickly and cheaply contractual relationships based on convergent business needs among parties that may not know each other, and therefore need to protect themselves from opportunistic or incorrect behavior. Solving the problem of *coordination* consists in creating, at convenient management costs, a control system capable of directing the objectives of the supply chain as a whole, so as to achieve a greater common good in the medium term, as an alternative to the state of affairs in which each participant pursues, on its own behalf, lower but immediate returns. Our smart contracts for innovative supply chain management replace human coordinators in tackling the problems above, thus eliminating one major obstacle to their effective solution, namely the need to trust the coordinator itself. Furthermore, in this way, by automating the process of coordination, they unburden the supply chain of a considerable management cost. Contracts of this kind not only automate contract execution as in standard smart contracts, but also adjust costs and compensations of the members of a supply chain, effectively taking up the role that was of human coordinators. Thus, we refer to them as “intelligent smart contracts.” In the course of the paper, we will illustrate an innovative supply chain architecture based on intelligent smart contracts running on blockchain, we will detail the algorithmic methodologies underlying the decision-making process of these contracts and we will outline the wider socio-economic perspectives opened by our approach.

Keywords: revenue sharing, supply chain, blockchain, smart contract, digital trust, algorithmic coordination

INTRODUCTION

Blockchain technology initially established itself through cryptocurrencies, but is currently seen as carrying huge potential for many further applications, due to its effectiveness at managing trust among stakeholders and at enforcing high standards for data protection, transparency, and authentication, without resorting to third parties or central authorities. Blockchains are deployed on wide-area networks such as the Internet and, in a broad perspective, it is conceivable that they will extend the scope of the Internet from Information and People to Value, by making value transfer viable on a global scale. In this way, through the trust algorithmically ensured by the blockchain and the consequent data protection, values of products and services can be exchanged among parties, who do not need to be acquainted with each other, to their best advantage without having to go through the encumbrance of costly intermediaries and laborious bureaucracy. This implies, however, re-thinking and redesigning business processes in a variety of sectors, from food to manufacturing, from health to government, from copyright to large-scale retail. Supply chain management is one such process that, within this outlook, has recently gained the limelight.

Extant use cases for the blockchain in the context of supply chains have been implemented by distribution giants such as Walmart¹, which has focused on blockchain-based monitoring of the safety of its food supplies. It is expected, however, that smaller players can reap even larger benefits from applying blockchain technology to their supply chain management. Indeed, a promising arena for the application of blockchain-based supply chains is given by business ecosystems made up of multiple corporate actors, more or less equivalent in size, which, through the blockchain, attain the possibility of intertwining supply relationships on equal footings without undergoing the hegemony of larger chain leaders, apt to impose economic terms and conditions to their advantage. It is well-known that one of the main challenges in the effective construction of these balanced supply chains is to establish trust relationships quickly and effectively among the various participants. This is a non-issue in unbalanced chains, but just because chain dominators impose themselves and act as managers of trust. They do this in the strictly technical meaning of providing directly or indirectly the legal (contractual documentation, legal advice etc.) and financial (validation of payments, credit management etc.) apparatus pertaining to the execution and validation of transactions. But they also guarantee the overall reliability and solvency of the chain, and ensure and, if necessary, impose effective levels of internal collaboration. Algorithmic trust, as made possible by the blockchain, is well-suited to cope fairly with this aspect, doing away with the need of power asymmetries to get things to work.

Yet, for this to become reality, another obstacle needs to be overcome. Indeed, for balanced supply chains to compete on the market on a par with those dominated by corporate giants, their participants must strive for a greater good that lets them create innovative products at competitive prices. This is at

odds with the traditional way of operating, whereby participants charge steeped-up prices of the components supplied to other participants, all to the detriment of the correct pricing of the final product. It is therefore a question of finding an attractive and sustainable alternative to the approach that favors small but immediate gains.

As a matter of fact, there does exist a methodology that applies exactly to our case and has been widely explored within the supply chain community during the past two decades [in research at least since the seminal paper (Cachon and Lariviere, 2000), while the practice of the approach dates back to the 1990's]. It is called Revenue Sharing (RS) and involves sharing the proceeds of the sale of the finished product among all the participants, starting from the retailer and moving backwards up until reaching every member of the chain. In this way, suppliers participate in the final gain of the retailer, in exchange for initially supplying at a discount, while protecting downstream the retailer, who can therefore afford to order as many items of the product as warranted by market forecast, without risk of paying penalty for excess of unsold items in case demand drops. Why then, despite the evidence of its benefits and the existence of some relevant success stories, has Revenue Sharing not yet found the large-scale adoption it seems to deserve? The answer is summed up in two words: trust and coordination. Participants must indeed be bound by a pre-existing strong mutual trust if they have to take for granted the timely and precise communication of sales made and the consequent fair distribution of proceeds. In reality, in the vast majority of cases, nobody goes that way and expensive forms of legal protection as well as of administrative control are hence made necessary to prevent misconducts from unreliable participants. Furthermore, the proceeds must be redistributed along an arbitrary length chain (i.e., two or more participants), which requires coordination involving additional administrative costs and dedicated human resources.

Blockchain technology can act as a crucial factor of change also in this respect, by making large-scale Revenue Sharing affordable so as to pave the way for balanced supply chains and, in general, for balanced business ecosystems. This is because it can automate the management of trust and coordination between the participants in the chain through the application of smart contracts, i.e., software programs that embody contractual relations between the parties and are written and executed on the blockchain. Thus, smart contracts can enact transactions corresponding to the sale-purchase of components within the supply chain according to internal prices optimally defined with respect to the pricing of the final product and accepted and verified by all participants. Redistribution of proceeds can be similarly handled.

Here automation is key, in that it radically lightens the burden of coordination which, if implemented through human managers, hinders the adoption of Revenue Sharing with computational and economic barriers bound to ramp up as the chain lengthens. However, Revenue Sharing is not limited to calculating and distributing sale proceeds, but also optimizes various parameters and indicators in order to plan the best price for the final product. By incorporating this logic too, smart contracts enter the wider arena where computers are propelled

¹<https://www.ibm.com/annualreport/2018/walmart.html>

beyond ordinary deterministic computations, and enabled to make intelligent choices along the lines and vision of trends like artificial intelligence. Hence, we refer here to such smart contracts as “intelligent smart contracts,” which not only automate the various steps of production and distribution (thus being smart, as in the common sense of the term in a blockchain context) but also plan for optimal returns (thus being intelligent).

The main contributions of this article are therefore as follows:

1. We characterize a class of “intelligent smart contracts” containing Revenue Sharing in first instance;
2. We provide a description of the algorithm of Revenue Sharing and general architecture for its deployment as an intelligent smart contract that has Ethereum as a reference point but can also be adapted to other blockchains;
3. We show how RS-based intelligent smart contracts totally automate the aspects of trust and coordination in supply chain management, thus effectively pushing forward the state of the art in this sector;
4. We highlight the substantial boost that this solution could channel for the purposes of a more balanced market, freed from oligopolistic yokes, where small and medium-sized companies with high value-add can live and thrive, with benefits that spread to the socio-economic fabric at large.

The remainder of the article is structured as follows. Section Blockchains, (Intelligent) Smart Contracts and Digital Trust recalls the main aspects and concepts underlying blockchains and smart contracts. Section Algorithmic Supply Chain Coordination illustrates the Revenue Sharing methodology of cost optimization and distribution of returns within supply chains. Section Intelligent Smart Contracts for Supply Chain Coordination describes the deployment of this methodology as an intelligent smart contract. Section Discussion and Future Work discusses the contribution of this article in the context of the perspectives opened by the blockchain in the direction of innovative economic systems. Finally, section Conclusions concludes the article.

BLOCKCHAINS, (INTELLIGENT) SMART CONTRACTS AND DIGITAL TRUST

Distributed ledgers are databases where transactions are validated through a decentralized mechanism involving a network of peer nodes, and are therefore managed in the absence of a centralized validator. The most popular implementation of distributed ledgers is given by the blockchain. The validation mechanism in blockchains amounts to a form of consensus among peer nodes and a number of consensus algorithms have been devised for this purpose, the most relevant and popular being Proof-of-Work [pioneered by the mother of all blockchains, the Bitcoin (Nakamoto, 2008)], Proof-of-Stake, and Byzantine Agreement.

Digital Trust

Since we are primarily interested in how blockchains provide a new version of trust, suitable for the flourishing of innovative economic models, let us see through a simple example how it

basically works. Suppose that company *A* records an invoice issued to customer *B* and containing data of both *A* and *B*, such as address, VAT number, invoiced amount, invoice number, bank details. Bank *C* views the invoice and agrees to advance 80% of its amount, based on an invoice advance contract between *A* and *C*. This operation is then in turn duly recorded. All three, *A*, *B*, and *C*, want to be certain that nobody tampers with these records, so as to prevent, for example, fraudster *D* from replicating *A*'s invoice data and then redirecting the paid or anticipated amounts to *D*'s bank account. Since digital files can be easily copied and modified, how to make sure that this does not happen? In general, how can binding agreements be protected from tampering, in order to demonstrate that each of them represents a specific, non-replicable state, inserted at a specific time? A traditional solution consists in using trusted third parties (TTPs) as guarantors, but this has considerable drawbacks, in particular the high prices imposed by TTPs for their services, often accompanied by laborious bureaucratic procedures. Indeed, such costs, both economic and procedural, fall under the category of “transaction costs” as defined in Transaction Cost Economics (Coase, 1937; Williamson, 1985) and are considered a hindrance to the establishment of large-scale business relationships. But it is here that blockchain technology can turn things over, with its ability to create tamper-proof ledgers and overcome the need of costly TTPs.

The word “Blockchain” in turn derives from two words, their combination determining both its morphology and its meaning. “Block” refers to a block of fixed size into which new transactions are entered and recorded up to its filling. Each new block is then added to the end of a pre-existing chain of blocks (“Chain”) that contains the digital history of past transactions. Since the nodes involved in validating transactions all keep a copy of the blockchain, each participating node has the complete and transparent history of what happened, thus making transactions secure. In addition, the blocks are encrypted and difficult to break in with the purpose of modifying the information they contain. A blockchain can therefore be seen as an incorruptible digital ledger of records, replicated on different computers in a peer-to-peer network, which guarantees the genuineness of the recording of data. Records can be anything, from money transfers to information of all kinds. Network members are anonymous individuals called nodes. All communications within the network exploit cryptography to securely identify senders and recipients for each transaction, by using the hash (unique identifier) contained within each individual block in addition to the information and hash of the previous block. The hash of the last previous block binds with the new block and ensures that neither one can be altered, hence preventing the insertion of new blocks between the two. In this way, each new block that gets into the chain strengthens the verification of the previous block and, consequently, of the entire blockchain. When a node wants to add information to the ledger, a consensus is formed in the network to determine where it will appear in the chain.

Summing up, a blockchain is a ledger of transactions. It is shared by multiple actors and, thanks to its underlying technology and processes, cannot be changed. Similarly, but with much higher execution times and costs, notaries (a typical

example of TTPs) record the actual and irreversible transfers of properties between parties. Therefore, blockchains provide the digital basis for the implementation of five key concepts, namely *decentralization*, *transparency*, *security*, *immutability* and *consensus*, which together lead to an even wider result: a digital (algorithmic) implementation of trust. Those same authentications of fiduciary relationships that had hitherto been entrusted to human intermediaries with high costs and sub-optimal timing, as well as with the burden of a cumbersome and intrusive bureaucracy, can now be dealt with by an algorithm: fast, simple and cheap. One of the consequences is that trust between businesses is made much more affordable, hence leveling out the barriers that have shackled so far cooperation among enterprises on a global scale.

Smart Contracts

Smart contracts have nowadays found a practical and effective implementation on blockchain technology. They have, however, an independent history that is worth reconstructing to understand both the way they work and their relationship with the contracts of the legal tradition.

At its most general, the concept of smart contract is rooted in various disciplinary areas such as information technology, economics and jurisprudence. To begin with, smart contracts were first introduced by Nick Szabo in the 90's of the last century (Szabo, 1997), therefore well before of the coming of age, more than 10 years later, of blockchains, which nowadays provide the background technology for their implementation. Consequently, at the time Szabo characterized smart contracts at an abstract general level, by describing them as methods to formalize and automate contracts from the legal tradition through the combination of computer protocols with user interfaces. Szabo talked about the potential use of smart contracts in various fields that involve contractual agreements—such as credit systems, payment processing and content rights management.

On the other hand, smart contracts are not legal contracts, if by the term “legal contract” we mean a real legal institution between two or more parties, which binds them to each other before the law. From this point view, smart contracts that run on blockchain are a partial departure from the concept as originated by Szabo, which nevertheless provided the initial inspiration. In fact, the blockchains that put the concept of Smart Contract into practice introduced a computer object not yet present on early blockchains, such as the Bitcoin, namely a type of code that can be recorded in an immutable way on the blockchain so as to run it automatically on a virtual machine. In other words, it is software that does not get executed on a single machine, but is instead executed on the virtual machine of a decentralized network, which makes it tamper-proof. Since a blockchain is a register for validating transactions, smart contracts are best viewed as automated transaction managers, and this is indeed what smart contracts retain from legal contracts. In fact, many legal contracts stipulate that transactions must be carried out if certain conditions are met. Thus, we can view a smart contract as the automation of the transactional part of a legal contract, with the guarantee that transactions will be effectively carried out

and will not be tampered with, something which fits neatly with the need to cope efficiently with the requirements of trust and coordination so as to effectively deploy Revenue Sharing.

Therefore, under this view, a smart contract is a program that runs on a blockchain and implements a digital agreement controlled by a set of rules. The software coding such rules is replicated and executed by all the nodes of the network. In essence, smart contracts enable the quick and cheap deployment of computer protocols embodying complex contractual obligations realized through the execution of transactions triggered by specific events (e.g., the date of distribution of a dividend, or the payment date of an installment of a lease) involving entities that may not have a history of former trust relationships. In this way, they raise the paradigm of digital trust, implemented for simple transactions through blockchains such as the Bitcoin, to a higher level, suitable for relations between companies. Smart contracts were first introduced in Ethereum (<https://ethereum.org>), whose community also developed Solidity (<https://solidity.readthedocs.io>), a contract-oriented programming language, but are now supported by several other blockchains like EOS (<https://eos.io>), Cardano (<https://www.cardano.org>) and Tron (<https://tron.network>), as well as by blockchain platforms for private and consortium blockchains, like HyperLedger Fabric (<https://www.hyperledger.org/projects/fabric>).

That said, the “smartness” attributable to this type of software must be appropriately circumscribed, since, in its basic usage, it lies entirely in automating the execution of causally linked transactions, rather than in the ability to intelligently make autonomous decisions about alternatives that may be available. Indeed, as commonly understood and implemented, a smart contract behaves like a deterministic program, by performing a certain operation *if* and *when* certain conditions are met. Hence, the code of a smart contract is often and largely made up of “*if ... then...*” constructs. We can summarize the main features of smart contracts as follows:

- **Automation:** smart contracts can automate all sorts of operations, within the limits of expressiveness of the programming context from which they originate. For example, Solidity, the programming language for coding contracts on the Ethereum blockchain, is Turing-complete, i.e., it can encode any problem that can be solved by a Turing machine, thus granting as much flexibility as there can be for the implementation of smart contracts.
- **Determinism:** smart contracts perform only the actions for which they are designed, whenever the relevant pre-conditions occur, with results wholly determined by the inputs they are given.
- **Distribution:** smart contracts are replicated and distributed over all of the nodes of a blockchain network, as opposed to software maintained on central servers as in the client-server paradigm.
- **Immutability:** smart contracts cannot be modified once implemented; such inalterability is a constraint, as well as a guarantee for users, that directly derives from their implementation on blockchain. Therefore, smart contract

versioning is decidedly different compared to that of traditional software, since new versions of a pre-existing smart contract are new smart contracts altogether, to which the inputs of previous versions must be redirected.

- *Transparency*: since smart contracts are based on blockchains, their source code is not only immutable but also visible to anyone on public blockchains, as well as to all authorized users on permissioned and private blockchains.
- *Trust*: smart contracts transfer the digital trust that is intrinsic to single transactions on the blockchain to the level of complex contractual relationships. As a result, two or more parties who do not know each other can now interact through contractual agreements that do not require expensive management and supervision by human third parties.

Intelligent Smart Contracts

The next stage in the evolution of smart contracts is in making them effectively intelligent, by going beyond the simple automation of the contractually enforced chaining of transactions. This evolution appears indeed desirable and justified every time it comes to planning, and therefore a variety of criteria must be used (for example, in the case of supply chains, profitability, delivery times, etc.) to choose among different possible plans that correspond to transaction chains implementable as standard smart contracts. Planning is at the heart of supply chain management and, in the case of Revenue Sharing, is practiced so as to define an optimal price to bring the product to market, taking into account criteria such as the degree of willingness of participants to adapt to optimized internal costs as well as possible variations in product demand. But it can also be applied to other contexts of the supply chain, like inventory control and operation control, and of industrial production in general.

Consequently, we can conceive the evolution of smart contracts toward “intelligent smart contracts,” capable both of autonomously deciding and tuning the inputs of transactions, as well as of executing them. This perspective does not upset the original foundations of smart contracts, but rather derives from the composition of two kinds of automation: (1) the automated execution of contractual conditions which, starting from Szabo’s initial contribution, has evolved into the current implementations of smart contracts on blockchain; and (2) the automated identification of optimized parameters for the execution of plans and actions that stems from artificial intelligence and operation research. An interesting analogy can be drawn with the components for, respectively, planning [analogous to (2) above] and actuating [analogous to (1) above] a complex action in a robot: on the one hand, the planner identifies the best options in the form of input parameters for the actuators (e.g., the motor components), thus exploring the non-deterministic space of possible plans; on the other hand, once the best plan is fully defined and decided upon, it is then executed by the actuators deterministically.

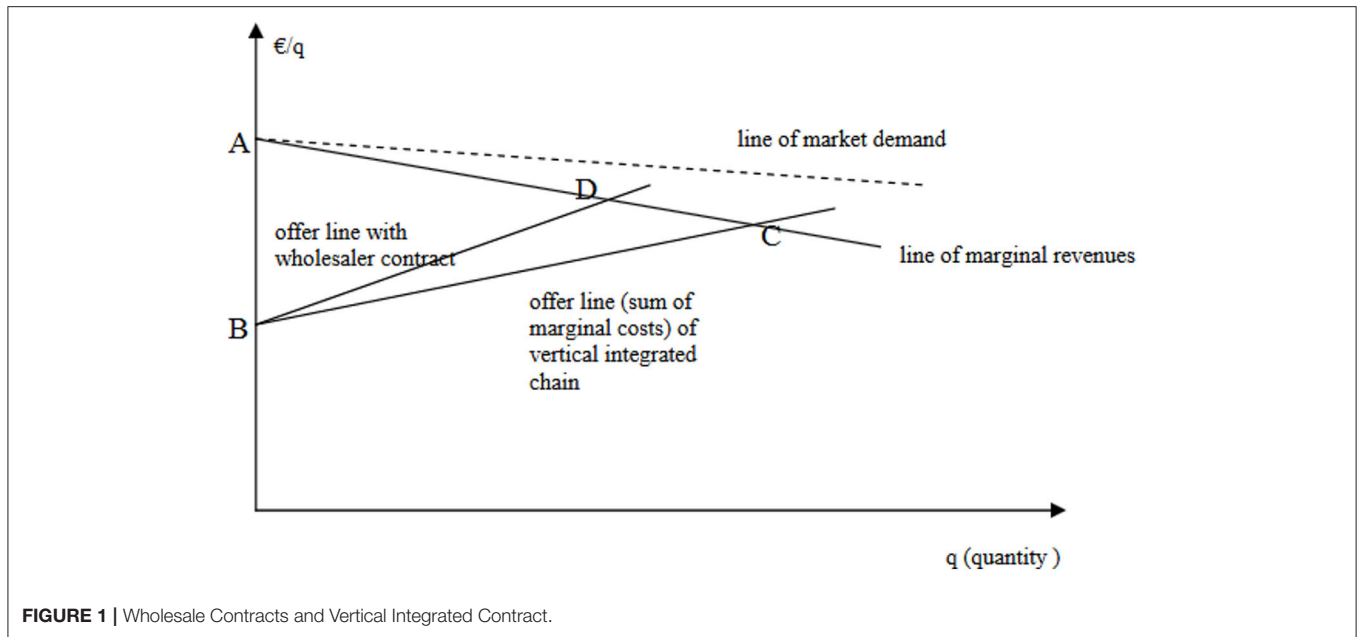
Our contribution here is in the construction of the formal and computational apparatus for a relevant class of such intelligent smart contracts, aimed at the optimized automation of supply and production chains in collaborative business ecosystems.

ALGORITHMIC SUPPLY CHAIN COORDINATION

Supply Chains and, generally, production chains composed of Small and Medium Enterprises (SMEs) need to be profitable, but also to rely on strong trust relationships to function effectively. On the other hand, those chain participants who enjoy a position of dimensional advantage, both organizational and economic, often impose themselves as chain dominators (Albrecht, 2009) and generally end up tightening their control over the other participants by managing the problem of trust to their advantage (Tononi and Amorosi, 2002). Munson et al. (1999) provide several examples of how in a supply chain the performance and returns for each of its participants can vary significantly based on choices related, among other things, to the internal prices of the goods traded, and how much the position of power (economic, financial, commercial, relational) of one participant compared to others may weigh in imposing one choice over another. On the other hand, it is the subtle interplay between trust and power that achieves the best results in order to implement the strategies and policies pursued by chain dominators (Sridharan and Simatupang, 2013). As the management of trust by predominant subjects translates not only into an advantage to them, but also is a way of making the chain work as a whole, Bitran et al. (2007) propose the use of TTPs precisely to ensure the functioning of balanced supply chains. In this way, however management costs ramp up, a problem that is instead overcome through the digital trust guaranteed by the blockchain, hence getting rid of trust management through either predominant subjects or costly TTPs.

In this section, we detail how profitability can be guaranteed through an algorithmic methodology of calculating fair sharing of revenues among peer participants. In the following sections we will show how trust can be achieved by implementing this method in the form of “intelligent smart contracts” supported by blockchain technology, thus relying on algorithmic trust rather than on dominating trust managers or TTPs. The algorithm described below is aimed at implementing the distribution of the proceeds to supply chain partners in a balanced and decentralized way, so as to maximize the profit of each partner in chains that operate in the absence of dominators. The integrated approach to supply chains described in Massa et al. (2007) shows how the growing need for competitiveness and high levels of quality has shifted the focus of logistic research toward approaches that make extensive use of information technology to optimize coordination in supply chains. Starting from these premises, an innovative management model for supply chains was developed, with the aim of combining competition and cooperation among participating companies and with specific focus on the needs of SMEs. Thus, the foundations were laid for the approach presented here.

Decentralized supply chains are usually organized according to *wholesale contracts*, whereby resources are exchanged through purchase and sale along the chain. As a result of the accumulation of mark-ups at each transaction, and of the consequent price increases, chain retailers tend to minimize the risk of unsold inventory by ordering and placing on the market smaller



quantities of product units compared to what happens within *vertically integrated chains*, which are managed centrally by companies acting as chain dominators that impose the exchange of resources at cost value (Pappas et al., 1983). As a consequence, chain dominators tend also to dominate the market.

In **Figure 1**, the wholesale contract and the vertically integrated contract are compared, showing that, with the wholesale contract, the offer level is the quantity corresponding to D, hence lower than the offer of the vertically integrated chain corresponding to C. The revenues will thus be lower and so will be profits, as measured by the area ABD, for the wholesale contract, and by the area ABC, for the vertical integrated chain. These considerations show that there is an intrinsic incentive to the adoption of the behavior of a vertically integrated chain, where the most outstanding and best performing feature is the exchange of resources, along the chain, at cost value. The quantity Q of resources that the vertically integrated chain is able to bring to the market is greater and, although the exchange of resources occurs at cost, the profit of each partner will be higher, as will be clarified in the following paragraphs.

Clearly, replicating this feature of integrated vertical chains within decentralized chains means that the proceeds collected by the retailer from the sale to the market amount to the revenues of the whole chain and not just of the retailer. Hence the need for an adequate process for the distribution of such revenues among all the companies participating into the chain.

As is well-known, this behavior is provided by so-called “Revenue Sharing Agreements,” represented in **Figure 1** with the line BC. Much has been written about this type of contract and its proposed variants for implementation in the real world (Cachon and Lariviere, 2000), such as the “repurchase agreement” and the “franchise contract.” The Revenue Sharing approach proposed in Massa et al. (2007) gives the best results if the following two conditions are met:

- Disincentives are imposed on chain members that exhibit higher production costs than real ones;
- The supply chain is kept balanced so that each member has a clear relationship with suppliers and no dominator is present.

With regard to the first criterion, it is clear that every company would tend, without appropriate countermeasures, to exhibit a higher production cost than actually sustained, so as to guarantee a (latent) surplus already in the recovery phase of the costs incurred [and opportunistic adjustments on mark-ups, as proposed by other researchers including Cachon and Lariviere (2000), do not appear effective at curbing this attitude]. The following approach has been designed to prevent this behavior and, at the same time, satisfy the second criterion (Massa et al., 2010):

To apply the Revenue Sharing algorithm, regardless of the size of the supply chain and the number of supplier-customer relationships, the only constraint is the product. This means that the supply chain is product-oriented and only partners with an effective role in the production process of the specific product must be involved: consequently, the data collected are all regulated on the specific product and its market demand.

It is also assumed that the involved companies all agree to adopt a Revenue Sharing approach and, therefore, to exchange resources at cost, clarified in the Request of Tender conditions before the implementation of Revenue Sharing. The power of the approach is that the revenues of the entire supply chain can be simultaneously managed and maximized.

In the past, the Revenue Sharing contract has had a slow diffusion, and some authors have attributed this situation to its complex implementation (De Kok and Graves, 2003). Within the Intelligent Smart Contract framework, these complexities in implementing the revenue sharing contract are taken care of by an IT Platform based on blockchain technology that

can guarantee automation, trust and decentralization of the management of the Revenue Sharing contract.

As we have already pointed out, complex implementation is not the only problem of the Revenue Sharing contract, since it is also exposed to opportunistic behaviors of member enterprises, which may choose to declare operational costs higher than those really incurred, so as to cash in some profit even at the time of the initial distribution aimed at cost recovery. This behavior damages the whole chain because it changes the offer line by affecting negatively the overall profit of the chain (by moving the line BC in **Figure 1** toward line BD). To avoid this scenario, specific constraints are agreed upon and settled in the Request for Tender phase, before Revenue Sharing takes place, including, but not limited to, the following ones:

- For each SME the sub-supplies are identified, and thus the chain is fully defined and designed from start to end.
- Refueling periods related to the dispatch of all subcontracts are also identified.
- The production flows assigned to each SME (q_{sme}), relative to its output resource, are calculated.
- The market demand of the specific product is identified and defined before the Revenue Sharing running phase.

Steps of the Revenue Sharing Algorithm

In the following paragraphs, we provide a high-level outline of our Revenue Sharing algorithm, and summarize the various steps with some simplifications with respect to a fully detailed version of the process. The aim here is to give an idea of the procedure to better understand the whole vision and the implementation of the methodology using blockchain technology and intelligent smart contracts.

Step 1

For the process of managing the revenue sharing contract, as a first step, the average unit cost of each member (that we identify with c_{sme}) must be calculated, as follows:

$$c_{sme} = \frac{C_f}{T_p \times q} + C_v \tag{1}$$

The cost is calculated based on: C_f , the fixed production cost of each company supplying a specific resource; C_v , the average variable production cost of each company supplying the resource; q , the production quantity of each company supplying the resource defined in the Request of Tender; and T_p , the period of recurrence (in days) of the fixed cost of each company providing the resource.

Step 2

For each resource k , it is necessary to identify the minimum of competing suppliers' costs for that same resource.

We will call c_{min}^k this minimum cost.

Step 3 [Distribution of Proceeds of Law (PL)]

We now calculate the amount of revenue from sales to be assigned to the group of suppliers of the same resource (k) as follows: $PL_k = q_v c_{min}^k b$ [where q_v is the market demand, b is

the bill-of-material (BOM) factor which identifies the conversion factor between the input-output resource in a node of the chain from the final product back to the resource k .]

$$PL = \sum_k PL_k \tag{2}$$

The Equation (2) represents then the part of the proceeds from the sales that must be assigned to the resource provider group as operating a cost recovery. The individual member will receive a fee PL_{sme} proportional to its supply quota f_{sme} of the specific resource in the chain.

f is calculated as follows:

$$f_{sme} = \frac{q_{sme}}{\sum_k q_k} < 1 \tag{3}$$

where q_{sme} is the supply of the *specific* resource provided by each *sme* and $\sum_k q_k$ is the entire supply of that resource in the chain [e.g., three tons of potatoes (q) provided by a member on nine ton of potatoes ($\sum_k q_k$) requested by the supply chain].

In these first three steps the revenues, collected by the retailers from selling the goods to the consumer market, are first distributed among member enterprises to cover the operational costs. In the “nodes” of the chain/network where more enterprises provide the same type of resource, the revenues are distributed at the rate of the minimum average cost in the node (c_{min}); i.e., the revenue collected by an enterprise of the node is given by the quantity it has provided times the average cost of the most efficient enterprise in that same node. Through this procedure, only the most efficient enterprises will fully recover their costs within this step. This condition is introduced as an incentive to efficiency for member enterprises and, analogously, as a disincentive to opportunistic behavior.

Step 4 (Recovery of Alignment Costs)

In this step, the recovery of alignment costs (P_{All}) must be distributed; these are the costs incurred by some members to align themselves with the interests of the chain.

These costs include those due to alignment to quality characteristics and product/process innovation. These installments will be deducted from the part $R - PL$, where R is the total revenue collected from market by the retail level of the supply chain.

After this step, what remains to be distributed is called PC , Chain Profit:

$$PC = (R - PL) - P_{All} \tag{4}$$

At the end of this step we have the Chain Profit value (PC) that represents the revenue of the chain after the payment of efficient production costs (PL) and the alignment costs P_{All} .

Step 5 (Profit Distribution)

At this point, after the above steps, what is left to the market collection of revenues is the part called “Chain Profit.” A previous negotiation among nodes (including those that have one single enterprise) has set the quotas of sharing of the

Chain Profit. This negotiation need not be held at each market supply, but quotas can be reviewed periodically or on demand by member enterprises.

The distribution of profits must be preceded by a phase of negotiation between the groups in the Request of Tender phase. What needs to be negotiated is what share of the PC profit should be assigned to the various resource (k) group providers. Said G_k this quota, it must be

$$\sum_k G_k = 1 \quad (5)$$

There is no need for bargaining within the same resource provider group; in fact, it is sufficient to assume as the opening bargaining quota, within the group, the request of the most demanding member, and as the minimum closing quota the request of the least demanding member.

Generally, it will not be necessary for bargaining to take place with every supply of the chain: where satisfactory agreements are reached, these can be maintained for more supplies.

Finally, each enterprise will receive the portion of proceeds equal to:

$$PC_{sme} = f_{sme} \times G_k \times PC \quad (6)$$

Empirical and Analytical Evidence for Revenue Sharing

To provide empirical evidence of the proven benefits as well as of the as yet unexpressed potentials of Revenue Sharing, we report here some of the results from the research project LEMURE (for Logistica intEgrata MUltiagente per REti di PMI) funded by Italian Ministry of Research under law 297/99, grant agreement 2007/32458, where various approaches to supply chain management were simulated and compared on the basis of data collected from the various participants in a chain. The results of the project presented in the final public event highlighted both the power of vertical integration and the benefits of Revenue Sharing. In the eyes of today, the simulations that were carried out give also solid ground to the desirability and potential of the technological enhancement of Revenue Sharing through blockchains and smart contracts. In fact, they provide evidence that, in traditional technological configurations, Revenue Sharing does work effectively, but only if implemented as an additional component of vertical integration, in which the dominant partner takes on the role of manager of trust and coordination. Here we highlight this aspect as it emerges specifically from the LEMURE case study, while Wang et al. (2004), Chen et al. (2011) and Gong et al. (2018) analyse and discuss the models used by giants of the Internet economy such as Airbnb, Amazon, Uber and eBay to effectively exploit Revenue Sharing in a vertically integrated context where they act as coordinators and managers of trust.

In LEMURE the methodological framework described in Massa et al. (2007) was applied to a supply chain composed of six SMEs, based in the Puglia Region in southern Italy. The agri-food chain was the focus of the project and the production process of a IV gamma product (semidried tomatoes) was tested. The supply chain was composed of four raw material suppliers (two tomato suppliers, a gas supplier and a pack supplier),

a semidried tomatoes producer, and to simplify the approach all the retailers were collected in one single virtual subject. The semidried tomatoes producer, Ortoreale s.r.l. was the chain leader, partner in the LEMURE project and data collector for all the supply chain. In the experimentation phase a set of data about production and transportation of a semidried tomatoes supply was collected: fix production costs (aggregated data), variable production costs (aggregated data), production time, production range (related to variable costs), holding costs, fixed and variable logistic costs, warehouse capacity. Applying the methodological framework of which Revenue Sharing is part yielded an average +8.3% increase of revenue for each partner. The experimentation phase was developed starting from a market demand of 36 pallets of semidried tomatoes, equal to a total amount of 5,040 trays of IV gamma product. To reach this amount and minimize supply chain costs the raw material partners had to provide 4,032 kg of tomatoes, 5 l of gas and 10 kg of packaging materials. With a market price of €3.94 for each tray of semidried tomatoes the proceeds of the supply chain were €19,850. **Figure 2** shows in detail the return of each partner from the lot of supply tested in the project.

These results can be compared with the wholesale approach as in the graph in **Figure 3**, where historical data were used for the comparison. As shown there, by virtue of this choice, with tomatoes of the same quality supplied by the two participants “Tomato supplier 1” and “Tomato supplier 2,” “Tomato supplier 1” reaped a 17% improvement compared to the wholesale approach, the highest profit increase in the chain, by keeping prices at production cost at supply time so as to receive an increased order from the chain, and subsequently fully recover the costs from the first phase as well as benefit from a higher final profit. By contrast, the returns for “Tomato supplier 2” remained unchanged, as it was unable to adapt in order to seize the opportunities offered by the new type of contractual relationship and instead followed the old pricing methods for its products. This case shows also quite well the co-opetitive scenario fostered by the application of Revenue Sharing.

Qian et al. (2013) provides another relevant case study in the food sector, carried out in the Chinese dairy industry, which suffers from a series of structural problems, including an unbalanced allocation of profits along the supply chain to the favor of retailers (supermarkets) and to the detriment of farmers and manufacturers. This imbalance seriously jeopardizes the sustainability of the entire sector. The study was carried out in the North China’s Inner Mongolia Autonomous Region, where most of the dairy production for the whole China takes place through two of the country’s largest dairy enterprises, Mengniu and Yili. The aim of the introduction of Revenue Sharing in the management of this dairy supply chain is to increase its overall profits in order to benefit farmers and manufacturers without affecting the profits of the retailers. This is indeed achieved by applying a three-stage version of the Revenue Sharing model provided in Giannoccaro and Pontrandolfo (2004) with an increase of overall profitability of 12.49%. However, the practical implementation beyond the experimental level of this RS-based dairy supply chain model poses the usual problems of coordination and administrative control that can be effectively

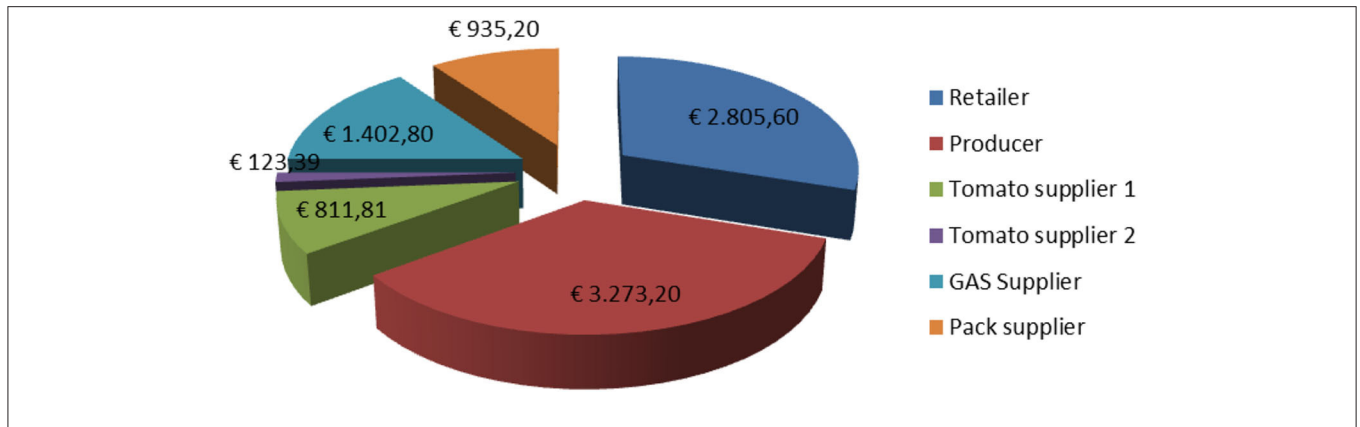


FIGURE 2 | Returns from vertical integration with a Revenue Sharing component in LEMURE.



FIGURE 3 | Comparison of supply chain profits from the wholesale approach and from vertical integration with Revenue Sharing adopted by “Tomato supplier 1”.

addressed with (intelligent) smart contracts. In this way, the negotiating power and the consequently higher returns of stronger participants can be constrained within the acceptable boundaries indicated by the authors of the study, without impairing the overall win-win effect deriving from the adoption of Revenue Sharing. By contrast, if the stronger participants took over the role of controllers and coordinators their advantage could grow to a far larger extent.

Hence, the indications we have from these results are that Revenue Sharing, if disengaged from the centralized control intrinsic to vertical integration, could give even more effective and wide-ranging results, by transferring the benefit of the approach to large-scale business ecosystems through the algorithmic management of relationships of trust and coordination, thus eliminating power asymmetries and balancing the role and weight of the participants

in the decision-making processes for supply terms and conditions.

A number of analytical models related to the benefits of blockchain technology in the support of supply chains provide criteria confirm the validity of this direction (Korsten, 2019; Liu et al., 2019; Choi, 2020; Hayrutdinov et al., 2020). In the next section we show how these findings can be concretized in a software architecture for Revenue Sharing that hinges on state-of-the-art blockchain technology.

INTELLIGENT SMART CONTRACTS FOR SUPPLY CHAIN COORDINATION

Real-world supply chains rarely come in a simple structure where the service or the product is supplied, produced, and

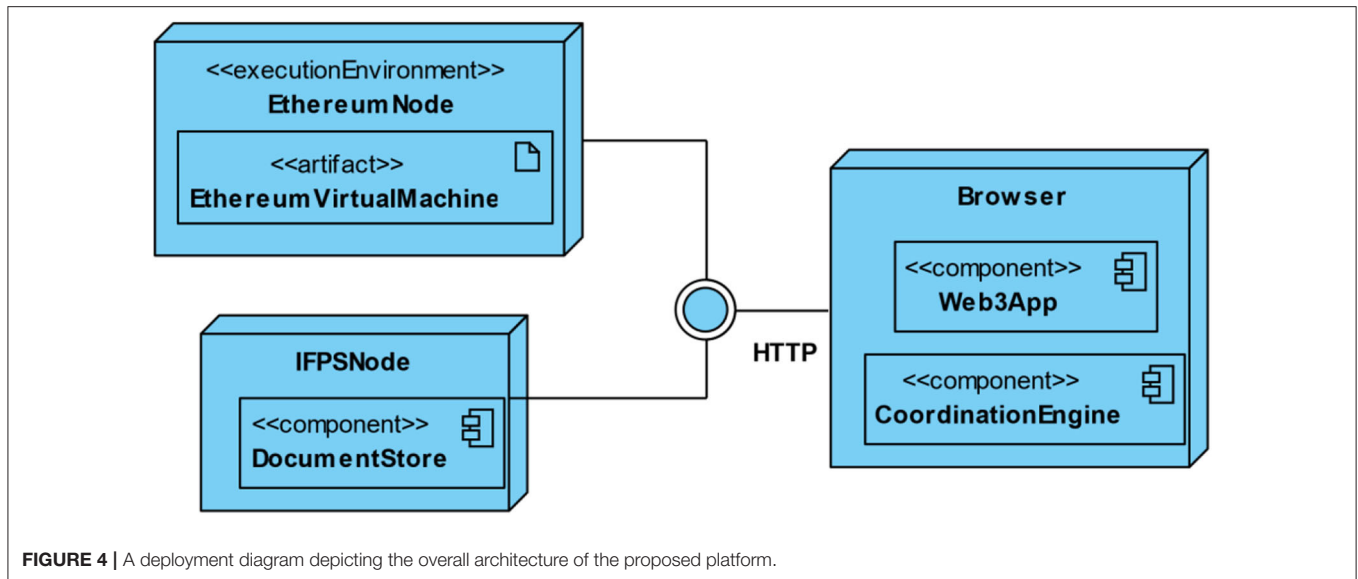


FIGURE 4 | A deployment diagram depicting the overall architecture of the proposed platform.

delivered on a single level. In most business practices, supply chains are layered over multiple levels involving stakeholders both downstream, toward end-customers, and upstream, toward raw material suppliers. The upstream refers to all the activities needed to collect the materials or inputs required to create a product or deliver a service. Conversely, the downstream process involves processing the materials or inputs collected during the upstream stage into a finished product or a delivered service, up to the actual sale of that product (Min et al., 2019).

As mentioned in the previous section, a collective agreement among all interested parties to adopt a common revenue sharing criterion is an essential prerequisite of our approach. The management of revenue sharing increases the operational complexity, which is already complicated by the multilevel nature of supply chains, where it is a matter of coping, among other things, with the challenge represented by the logistics of geographically distributed stakeholders, as well as with disruptions due to delays or changes in previously defined conditions. To deal effectively with all this and take advantage of the revenue sharing approach, we propose Intelligent Smart Contracts in direct technological continuity with Blockchain and Smart Contracts. Intelligent Smart Contracts not only manage the complexities in the implementation of the revenue sharing algorithm in multilevel supply chains, but also contribute to significant improvements in efficiency, transparency, security, traceability and streamlined integration among the different levels of the supply chain.

The proposed Supply Chain Coordination Platform leverages various IT technologies (see **Figure 4** for a high-level overview in the form of a UML deployment diagram). In particular, we adopt the Ethereum² implementation of the blockchain technology as the core back-end engine (see component *EthereumBlockchain* in **Figure 4**), on which to record all the effects of the transactions involved, from the establishing

of the supply chain, up to the distribution of revenues. The blockchain also supports calculations of optimal shares, implemented in Solidity³, while a Web-based client (component *WebClient* in **Figure 4**) incorporating a DApp (written with the Web3.js⁴ framework, component *Web3App* in **Figure 4**) is provided as a front-end, accessible from a Web browser, to facilitate the interactions between the different stakeholders of each Intelligent Smart Contract. It is assumed that the Web client will maintain a reference to some Ethereum node through which the blockchain can be accessed and transactions executed on an Ethereum Virtual Machine. The interaction events generated by the *WebClient* through the *Browser* and interpreted by the *Web3App* component are communicated to a component responsible to enable the execution of tasks as requested by the overall process (*CoordinationEngine*). Both front and back-end engines can refer to a suitable repository containing, among others, the documents describing the actual production of the goods involved in the smart contract (*DocumentStore*). The document store can be implemented in a distributed way, with no need for a central database, for example using the IPFS (InterPlanetary File System) infrastructure, as indicated in **Figure 4**, where reference is made to some generic IPFS node. Communication between the Web application, the blockchain implementation and the concrete databases will occur over the HTTP protocol, typically exploiting the JSON communication framework.

The approach to the construction of the application takes inspiration from tools such as *Lorikeet* and *Caterpillar* that derive Solidity implementations from models of the contract realized with the Business Process Modeling Notation (BPMN) (Tran et al., 2018; Di Ciccio et al., 2019) and from the idea, illustrated in Marchesi et al. (2018) of modeling the needed interaction by use cases and deriving a domain model from the description

²<https://ethereum.org>

³<https://github.com/ethereum/solidity>

⁴<https://github.com/ethereum/web3.js/>

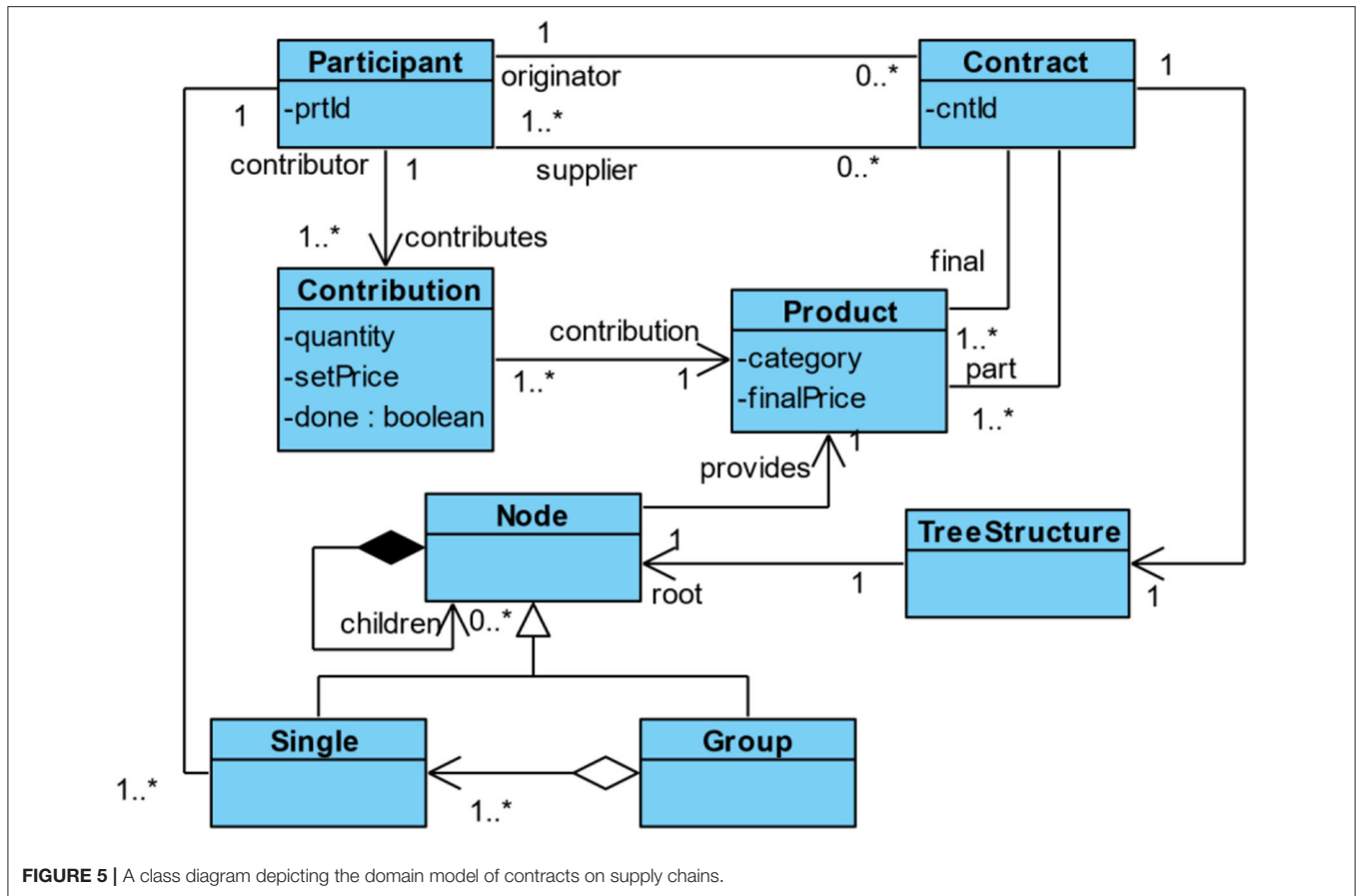


FIGURE 5 | A class diagram depicting the domain model of contracts on supply chains.

of the elements touched upon in the cases themselves. In our approach, we define a Use Case model to identify the actors and their interactions with the system, and the BPMN model to define the overall coordination and message exchange to be realized. For the domain model (see the UML class diagram in Figure 5), we consider the kind of information required to provide an integral view of the supply chain, taking into consideration the tree-like structure induced by the composition of the product. Specific categories of sub-products can be provided by different suppliers, at different costs and in different quantities, and one participant can produce different categories of sub-products, thus appearing at different nodes. Each contribution of a participant to the overall contract can be described in terms of the type of product provided, the quantity to be provided and the price. Depending on the state of the contribution, e.g., offered or accepted, these values can be considered as temporary or finalized. Similarly, the product to be provided may have been required or delivered. Only once the final product, for which the contract was established, is sold at a final price, will the revenues be calculated. Hence, we assume that only one participant is in charge of originating the contract, and setting the final price.

In the treatment below, we assume that all interested parties run an implementation of the WebClient, obtained by the developers, and that the endpoints for interacting with the

blockchain implementation of the supply chain are made known to them in due time.

The platform supports use cases for two types of actors, that we call *Originator* and *Supplier* (see Figure 6), roles which can be taken by any *Participant* registered in the system.

- *Originator*. A person or organization that initiates the whole process by requesting the supply of a product with specifications. This actor defines the main conditions that the system will later transfer into a new Intelligent Smart Contract. Any originator can initiate and participate in different smart contracts at the same time.
- *Supplier*. A person or organization that has the possibility of providing either product or service within an existing Intelligent Smart Contract. When placing a bid, suppliers provide competitive specifications for the sub-products they can offer to the contract. Any supplier can participate in different smart contracts at the same time. In certain cases, a supplier can start a subcontracting activity, thus acting as a sort of originator for a sub-branch of the supply chain.

Figure 6 shows the use cases for the originator and supplier actors. Both types of actor can perform their use cases, according to the coordination mechanism, through their accounts on the Web application, which also gives access to information

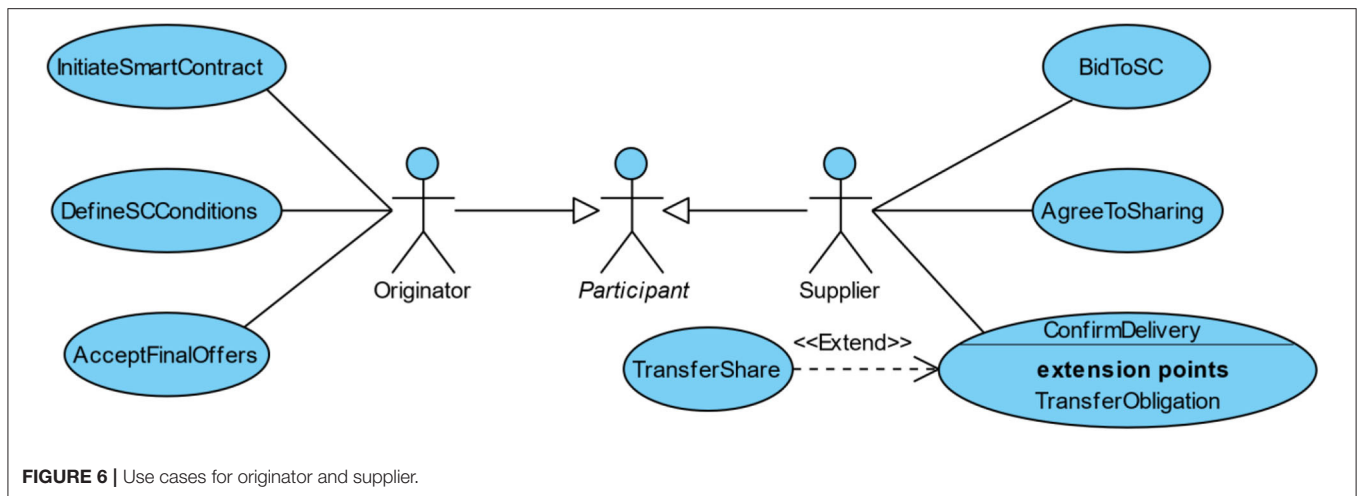


FIGURE 6 | Use cases for originator and supplier.

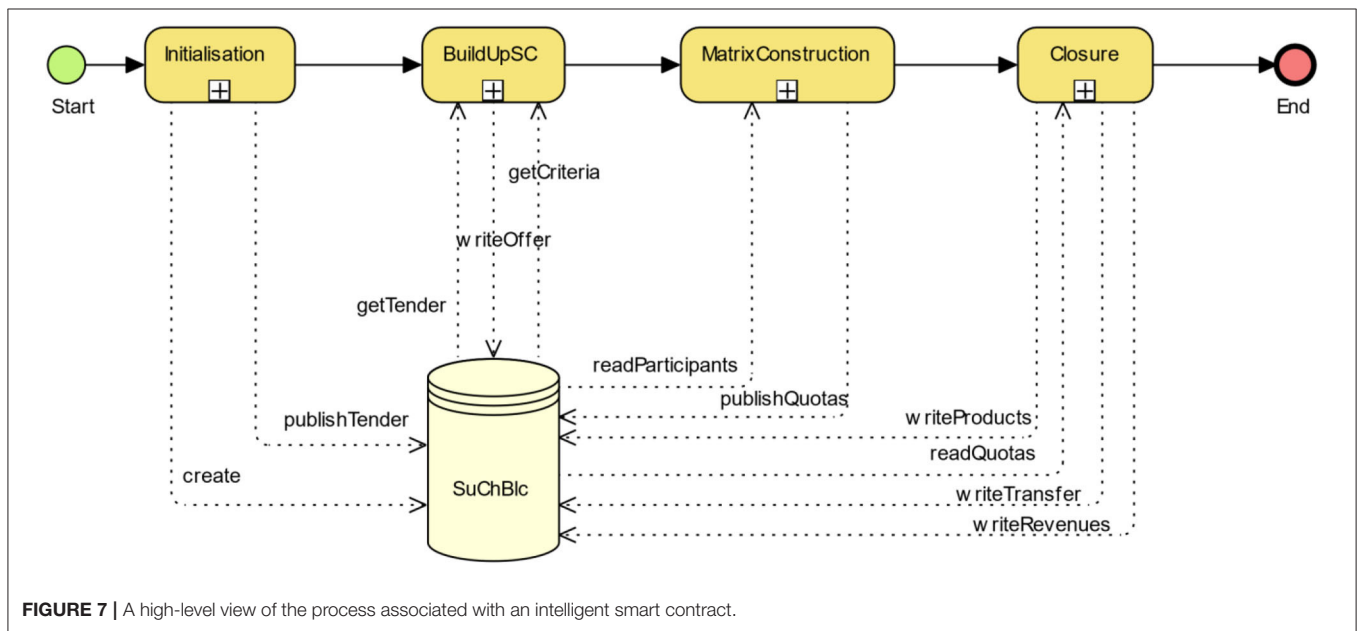


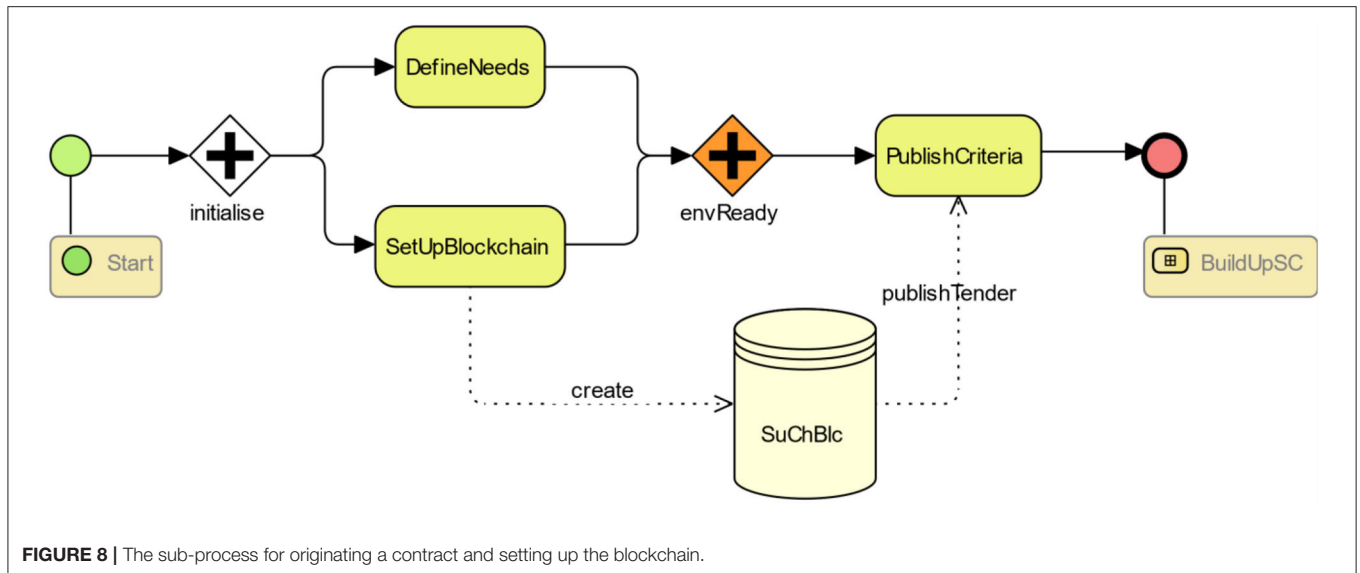
FIGURE 7 | A high-level view of the process associated with an intelligent smart contract.

concerning their active and past smart contracts. With respect to a specific smart contract, each participant can play only one role.

Actors' actions are executed according to a process model, described at high-level by the BPMN diagram of **Figure 7**, and detailed in the subsequent diagrams depicting various sub-processes. We only consider the actions of the tasks on the blockchain associated with the supply chain, indicated here as SuChBlc and assume that the associated DocumentStore is updated consistently with information on the interactions executed via the Web application. In particular, we consider that an intelligent smart contract is executed through four sub-processes: Initialization; definition of the smart contract (BuildUpSc); computation of optimal prices and shares according to the algorithms described in Section Algorithmic Supply Chain Coordination and storage of the result in the form of matrices,

memorized on the blockchain (MatrixDeployment); and actual execution of production, sale and revenue sharing (Closure).

Figure 8 details the initialization process, as performed by an originator. This actor defines the requests and set up a new Smart Contract on a blockchain, on which the criteria for participation and conditions for fulfillment will be published (e.g., required parts, quantity, quality, delivery methods, deadlines, and supplier selection criteria). Any new contract virtually defines a number of shares that suppliers bid to own. The number of Smart Contract shares each supplier owns will represent how much profit they will make by the end of this contract. All the interested suppliers are made aware of the publication of the new Smart Contract. Suitable and interested suppliers—from different downstream and upstream levels of the supply chain—bid/apply for fulfillment of this new smart contract.



The **Appendix** provides a series of snapshots from an implementation of Web interfaces for the originator and the suppliers within Revenue Sharing contracts. The texts in the interfaces are in Italian, as this prototypical implementation is meant to be used by Italian SMEs.

Once criteria are published, interested parties, becoming aware of the call, can produce their offers, as shown in **Figure 9**. This might entail offline negotiations, or the originator may select beforehand the possible participants, but offers to participate need to be performed through the Web application in any case, so that they can be documented and maintained through the blockchain. Suppliers might recursively start a call for sub-providers, which however will participate in the same contract, i.e., the blockchain will maintain the whole generated tree-like structure of the supply chain. This is indicated in **Figure 9** as a nested call to `BuildUpSC`. The system receives bids at the different levels recursively until the deadline for receiving bids. The final list of suppliers is defined by matching the received bids to the selection criteria provided by the originator in the new Smart Contract request (hence, it is described as the business rule task `AwardOffers` in **Figure 9**).

In **Figure 10**, we summarize the steps through which the matrix resulting from the previous sub-process is computed, according to the algorithm of Section Algorithmic Supply Chain Coordination, and published on the blockchain.

Finally, **Figure 11** offers a synthetic view of the sub-process leading to the actual production. The task `ProduceAll` will lead to loading on the `DocumentStore` all the relevant documents, e.g., delivery notices, acceptance forms, etc., with references to them written on the blockchain. Once the actual sales are performed (task `Sell`), the algorithm on the blockchain is executed, based on the saved matrices (hence it is again described as a business rule task, `ComputeRevenues`). The computed shares, transformed from percentages of a virtual quota to actual currencies, are then distributed (task `DistributeRevenues`), and the corresponding transactions

recorded. **Figure 11** also shows the interactions of these tasks with the blockchain.

The model also incorporates methods (not discussed here for the sake of simplicity) to recover from exceptions, where for example quotas of production have to be shifted from one supplier to another.

The implementation of the Web client is derived from both the process model and the deployed supply chain Smart Contract. In particular, a view in the Web3 application is created for each task in the process, while the coordination engine authorizes the execution of tasks according to the model. Constraints on the actions allowed in the different views derive from the different roles of originator and supplier assumed by users at the different layers in the supply chain. Suitable mechanisms for access control make sure that only relevant parts of the stored documents are available to different users. For example, an originator should be enabled to retrieve all the documents pertaining to its immediate relationship with an immediate supplier, but not those pertaining to the interaction of the supplier with its sub-providers, except for pieces of information which might be relevant to checking whether specific constraints are complied with, e.g., if products from certain countries cannot be included in the final product.

DISCUSSION AND FUTURE WORK

The radically evolutionary outlooks opened by blockchain technology for supply chain management have been a source of experimentation and discussion in recent years. The interest in this direction cuts indeed across the whole spectrum of possible chains, and includes co-opetitive business eco-systems -target of our contribution- as well as vertically integrated systems characterized by dominant companies. In both cases, the main objective is bringing transaction costs down. In one instance such reduction entails distributing larger profits to a larger business community, thus increasing the common good; in the other, it means increased returns to dominators

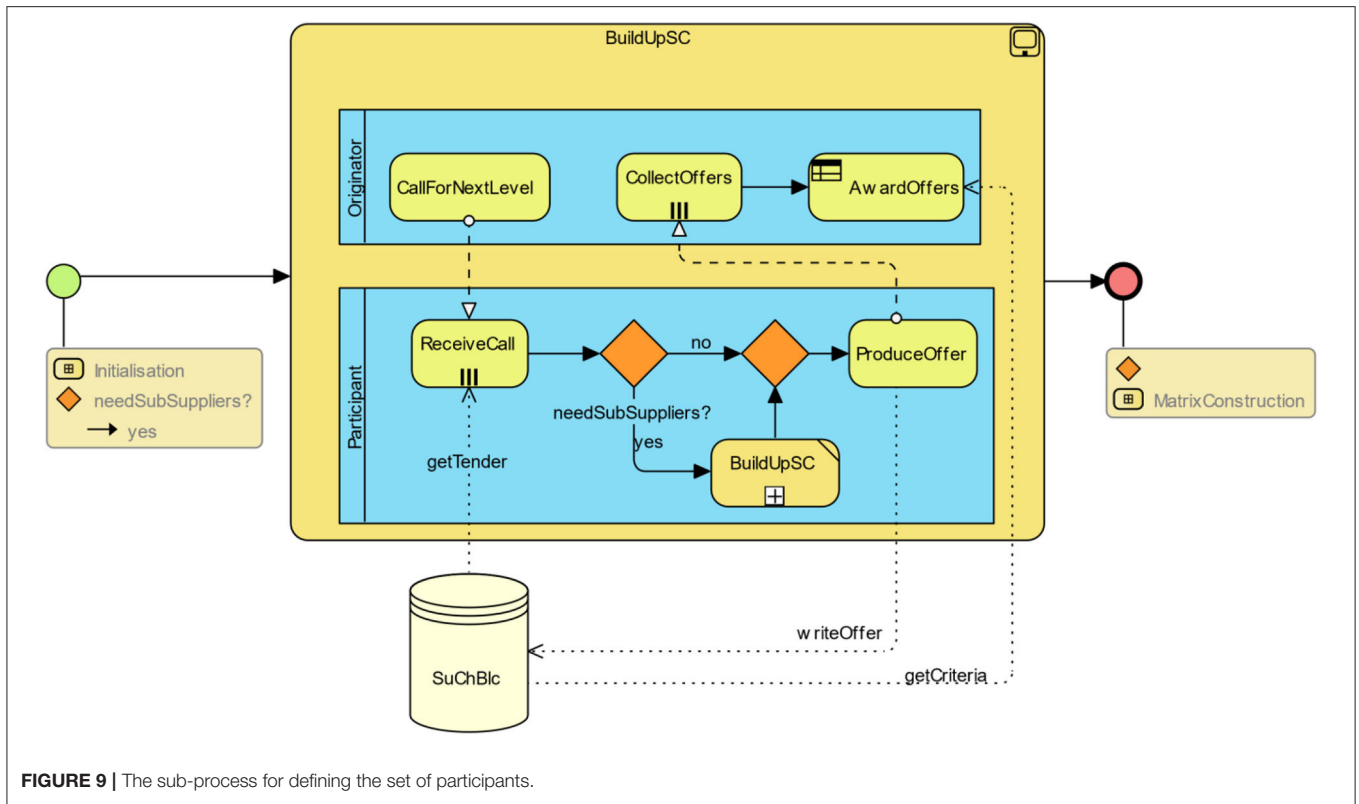


FIGURE 9 | The sub-process for defining the set of participants.

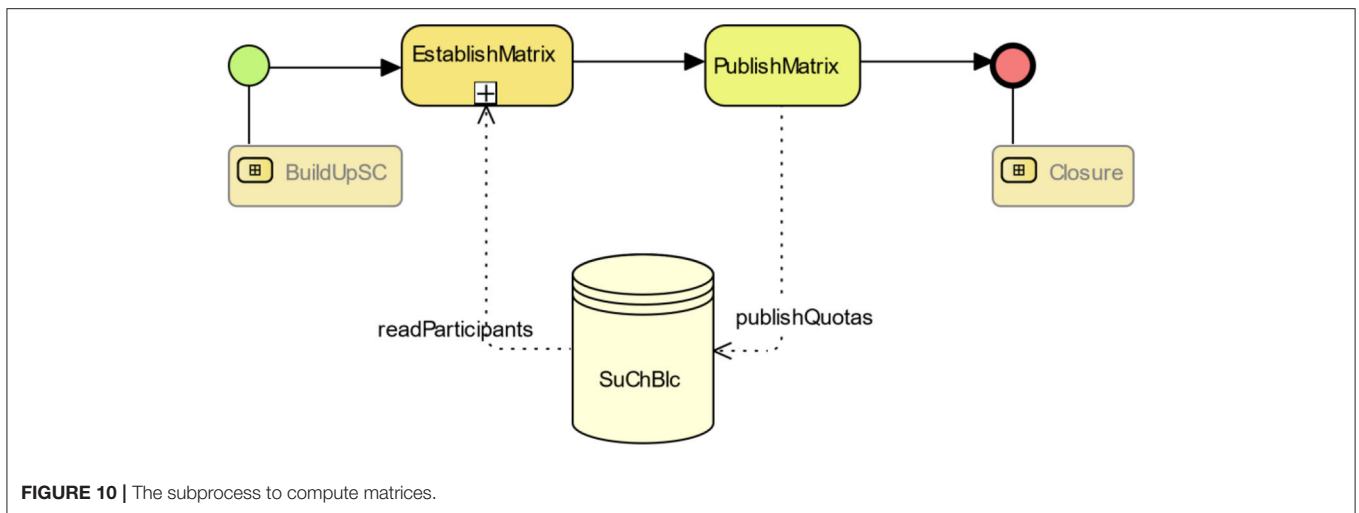


FIGURE 10 | The subprocess to compute matrices.

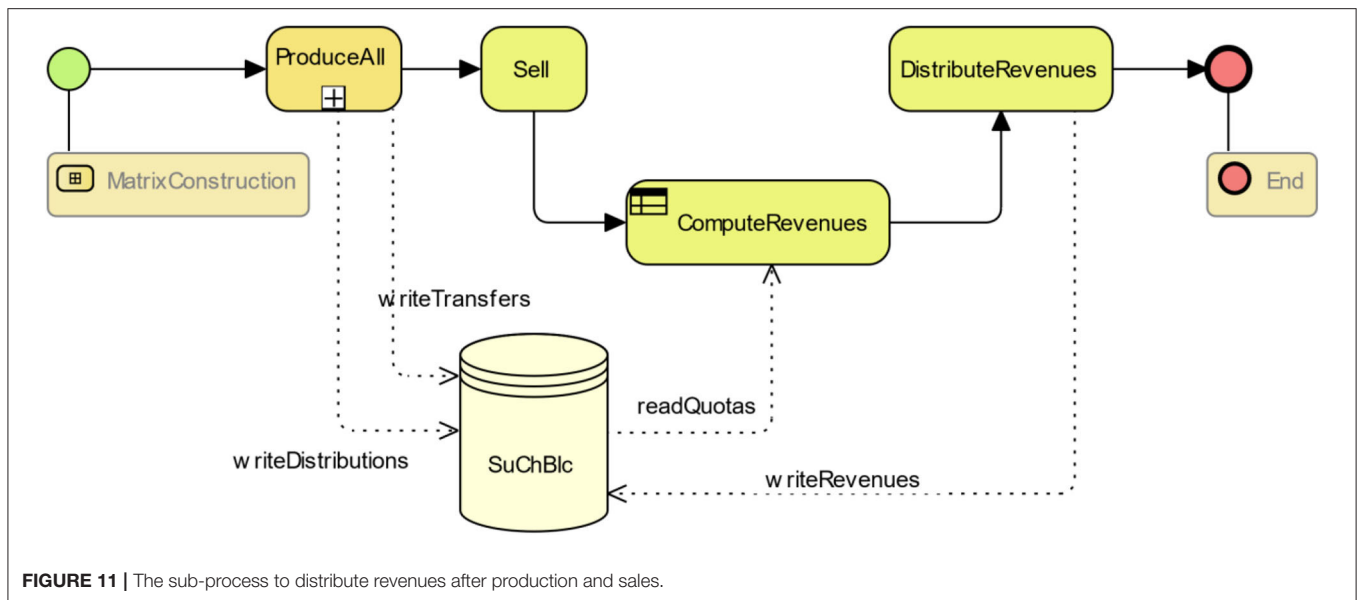
by virtue of further cutting supply costs and capillary control on the overall functioning of the chain. Effective examples of vertically dominated supply chains that have been strengthened through blockchain technology are the TradeLens platform, jointly developed by IBM and shipping giant Maersk⁵, and the supply chain of distribution giant Walmart, where blockchain technology, also powered by IBM⁶, is exploited to track provision disruptions and ill-functioning, such as sources of unsafe food.

Clearly, these are private blockchains, where the validation of transactions is strictly managed by the dominant company, which is thus even further strengthened in its role of manager of trust. Therefore, the algorithmic consensus and the digital trust that follows from it, which are key aspects of public blockchains, play no role here, while the specific feature from blockchain technology that gets exploited is the ability to maintain historical transaction logs for control and audit.

We must also mention those applications that, without setting themselves the aim of redefining the economics and organization of supply chains, leverage blockchains just to trace and notarize

⁵<https://www.tradelens.com/>

⁶<https://www.ibm.com/annualreport/2018/walmart.html>



the various elements and steps contributing to final products within supply chains, so as to endow end users with greater transparency and verifiability on these very products, with the expectation of beneficial effects on brand value and customer loyalty. Relevant among such sectors, in which blockchain-based boosting of supply chains for notarization purposes has been implemented, are agri-food (e.g., the Food Trust initiative managed by IBM on HyperLedger Fabric, that includes among its participants food giants Carrefour and Nestlé⁷ and luxury and fashion (e.g., the Aura initiative launched by the LVMH group⁸). It goes without saying that such notarization and tracing capabilities come as by-products in the context of blockchain-based projects that aim at a broader and more strategic reorganization of the fundamentals of production and supply chains.

From the existing literature on the subject, Korpela et al. (2017) provide, on the basis of empirical evidence obtained from interviewing firms and business managers, a broad overview of the perspectives opened by blockchain technology in the context of supply chains from a Transaction Cost Economics point of view (Coase, 1937; Williamson, 1985) and with a focus on the ensuing evolution of IT architectures for supply chain management. Treiblmaier (2018) defines a broad methodological framework to measure the impacts and organizational advantages from applying blockchain technology to supply chains where, in addition to Transaction Cost Economics, evaluation criteria are obtained from Positive Agency Theory (Jensen and Meckling, 1976) (which measures companies' competitive advantage according to their ability to establish partnerships with trustworthy agents), Resource-based View Theory (Wernerfelt, 1984) (which identifies companies'

ability to access and protect scarce resources as a key advantage) and Network Theory (Johanson and Mattsson, 1987) (which considers networking, namely companies' ability to dynamically establish advantageous business partnerships, as a key advantage). Our approach concerns not just the improvements in transaction costs deriving from the systematic adoption of digital trust made available by the blockchain, but also the optimal management of resources as implemented in the intelligent smart contract of Revenue Sharing, as well as the possibility of dynamically creating new supply agreements between partners that may lack former business relationships; it thus appears assessable also within this wider evaluation grid.

Liu et al. (2019), Korsten (2019), Choi (2020) and Hayrutdinov et al. (2020), already referred to in Section Algorithmic Supply Chain Coordination, provide analytical models demonstrating that Revenue Sharing is a supply chain management methodology optimally transferable to blockchains and smart contracts. Therefore, these contributions complement our own, which provides concreteness, in the form of implementation guidelines, for such formal demonstrations. Specifically, Choi (2020) considers the performance of an abstract Revenue Sharing model operating on blockchain in the fashion industry with monetary transfers between participants made in cryptocurrency, with the consequent disintermediation of the bank entities traditionally in charge of this task. Beside bank disintermediation, digital trust also plays a role in the model as a risk mitigation factor, and both of them are weighted against a number of assumed overheads for deploying Revenue Sharing on blockchain (in particular, the costs of adding transactions to the blocks). It follows from these assumptions that Revenue Sharing on blockchain is highly rewarding compared to traditional supply chains burdened by high banking fees, while performing non-inferiorly in the case of low banking fees. Hayrutdinov et al. (2020) provide a formal model that predicts that Revenue Sharing on blockchain can

⁷<https://www.ibm.com/blockchain/solutions/food-trust>

⁸https://content.consensys.net/wp-content/uploads/AURA_ConsenSys_Press-Release_May-16-2019-2.pdf

outperform traditional centralized solutions by incentivizing participants upstream in the supply chain, with the additional benefit that its supporting coordination mechanism sustains long-term relationships among the participants of decentralized supply chains. Along similar lines, Liu et al. (2019) demonstrate that, by reducing the overall transaction costs among supply chain members, an automated Revenue Sharing contract on blockchain can boost the performance coordination of the supply chain up to the levels of centralized decision making, while fostering superior information sharing and long-term relationships. Korsten (2019) demonstrates the performance of Revenue Sharing as a smart contract on the blockchain in the presence of risk averse participants.

Berg et al. (2019) outline a vision that stands out as very close in spirit to our proposal, and in fact forecasts a forthcoming economic scenario toward which our architecture of Intelligent Smart Contracts for Revenue Sharing lays out a viable path. This vision in turn aims at resurrecting the Electronic Markets Hypothesis put forward by Thomas Malone and his associates (Malone, 1987; Malone et al., 1987, 1989) in the late 1980's and the early 1990's, namely that the availability of a global IT infrastructure sets the premises for breaking down transaction costs, with the effect of boosting the cooperation between lean entrepreneurial subjects, to the detriment of hierarchical corporate behemoths weighed down by functions dedicated to carrying out activities that fall outside the core competences of the firm. That is, the Electronic Market Hypothesis envisages very slender companies because they are dedicated only to products and services for which they possess the key skills, while, in the face of near-zero transaction costs, anything else they need they get it from other companies, thus inexorably tilting the pendulum toward the buy option of the make-or-buy dilemma. However, the Electronic Markets Hypothesis has translated into reality only to a very limited extent, principally because it did not adequately take into account the need of companies to fend themselves from incorrect or opportunistic behaviors from untested business partners. Fact is that, as easy as it has become, by virtue of a global IT infrastructure, to establish relationships between companies on an unprecedented scale so as to pool key skills, rather than disperse resources and energies in secondary activities, in the end distrust still prevails over the attractiveness of doing business together. If anything, those who have benefited from the new set-up are the usual big players, which may have found convenient trimming down governance overheads by outsourcing secondary functions; at the same time, they have put further bonds over minor partners through information technology, while protecting themselves against misbehaviors through suitable legal fencing, which they can easily afford, and even more through their financial power, which leaves little room to smaller parties to play against the rules. But, as argued in Berg et al. (2019), digital trust as made available by the blockchain can create the pre-conditions necessary for generalized cooperation between companies of all types and hence turn the Electronic Market Hypothesis into reality. Our Intelligent Smart Contracts for Revenue Sharing can be a substantial step in this direction, as they couple digital trust with a ready-to-wear fair cooperative process model for enterprises participating in a supply chain.

Intelligent Smart Contracts for Revenue Sharing can be seen as a flexible form of planning, as they provide cost indicators that must be used in order to achieve optimal performance within plans for building and delivering products in the context of current market constraints, and then execute the plans defined from such premises. There has been recently a resurgence of interest for planning in economy (Phillips and Rozworski, 2019; The Economist, 2019), a notion until recently dismissed and even ridiculed as a relic of the socialist economies of the twentieth century. This interest stems indeed from the observation that colossuses of capitalist economy such as Walmart and Amazon make extensive and effective use of planning to squeeze costs and boost revenues in their logistics and supply chains. This has prompted some authors to revive the idea of a socialist economy as “planning for the greater good” which would not be hindered by the shortcomings of the old socialist economies in virtue of the algorithmic weaponry already exploited by Walmart and Amazon for their own good. It would also be supported by computer technology hugely improved compared to that available to communist countries in the age of multi-year economic plans, as well as by extensive and timely access to relevant “big data.” By contrast, our approach to planning can cohabit with entrepreneurial initiative as practiced in market economies; nonetheless, it implements those redistribution mechanisms that are at the very core of the Revenue Sharing model so as to pursue the greater good in the fairer context of an ecosystem of widespread producers. Without the need for major institutional upheavals, it could therefore be a panacea against excessive concentration of decision-making power and over-privileged access to economic and financial resources.

Intelligent Smart Contracts for Revenue Sharing can also be seen as Decentralized Autonomous Organizations (DAOs), a concept that has been widely discussed within the Blockchain Community and has a significant case history of implementations. In the definition of Vitalik Buterin⁹ “*The ideal of a decentralized autonomous organization is easy to describe: it is an entity that lives on the internet and exists autonomously, but also heavily relies on hiring individuals to perform certain tasks that the automaton itself cannot do.*”

This definition fits perfectly with Revenue Sharing, since the intelligent smart contract that implements it is an autonomous organization that lives on the Internet and hires humans, in the form of companies rather than individuals, to perform the complex task of working together on a product or service. It therefore operates as a combination of algorithmic automation and human activity. Indeed, Revenue Sharing further refines the concept of DAO as introduced and defined by Buterin. In Buterin's organizational taxonomy, in fact, the relevant parameters are automation at the center, that is, in the control room, and automation at the edges, that is, during the execution of specific tasks. Traditional human organizations lack automation both at the center and at the edges. Automation at

⁹<https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/>

the edges under human control is exemplified by robots without decision-making autonomy as used in manufacturing assembly lines. Automation both at the center and at the edges points in the direction of Strong Artificial Intelligence, which is at present a still unfinished construction. DAOs are organizations in which automation resulting from smart contracts resides at the center, in the automated management of processes and procedures composed by tasks whose execution is left to humans (active at the edges). Paradoxical as this may sound, while human organizations “employ” non-autonomous robots for the purpose of replacing automatable tasks traditionally performed by human workers, DAOs “employ” autonomous humans by automating the task of coordinating them, which is traditionally carried out by human managers and administrators. Known examples of DAOs are “The DAO” (Bannon, 2016), an investor-directed venture capital fund residing on the Ethereum blockchain launched in April 2016, when it raised the equivalent of over \$ 150 million, and largely disabled a few months later due to regulatory issues and security vulnerabilities; and Aragon (<https://aragon.org>), also Ethereum-based, a platform letting organizations work together on a general basis. On the other hand, Revenue Sharing pushes the level of automation further, in that it lifts to the center an aspect otherwise sub-optimally addressed at the edges, namely the allocation of costs and returns to executors of individual tasks, thus combining the “smartness” of process automation with the “intelligence” of planning, while still reserving the execution of individual tasks to human creativity and ability. For this reason, Revenue Sharing appears located in an area that partly borders on the region of Artificial Intelligence, and which we have labeled in **Figure 12** as intelligent DAOs (iDAOs) [We suspect there are several others of these dual areas, for example one given by “Augmented Intelligences” where the inferential capacity of the machine is constantly fed by interaction with humans, as in the case of the renowned problem solver Watson (Ferrucci et al., 2010)].

Future work will be along two lines, one technological and the other organizational. Technologically, it is a matter of identifying the key features in the existing blockchain platforms that suit best the performance of intelligent smart contracts for Revenue Sharing. We shall start here from the existing experience based on the Solidity prototype running on Ethereum, by building on existing indicators for blockchain-based supply chains like those in Litke et al. (2019) and by making use of decision-making methodologies for blockchain-based software developments like the one illustrated in Garriga et al. (2020). Organizationally, the aim is to further refine the existing model, by introducing tokenization systems to facilitate Revenue Sharing transactions and defining bank credit methods geared to the supply chains of peer participants, so as to relieve especially those stakeholders who, being at the beginning of the supply chain, most heavily bear the burden of early exposure to costs. The answer here is likely to lie once again in the blockchain itself, in that, as shown in Omran et al. (2017), the results of greater transparency and collaboration deriving from its adoption in supply chain management makes participating partners more eligible for funding, possibly by resorting to innovative forms of financing.

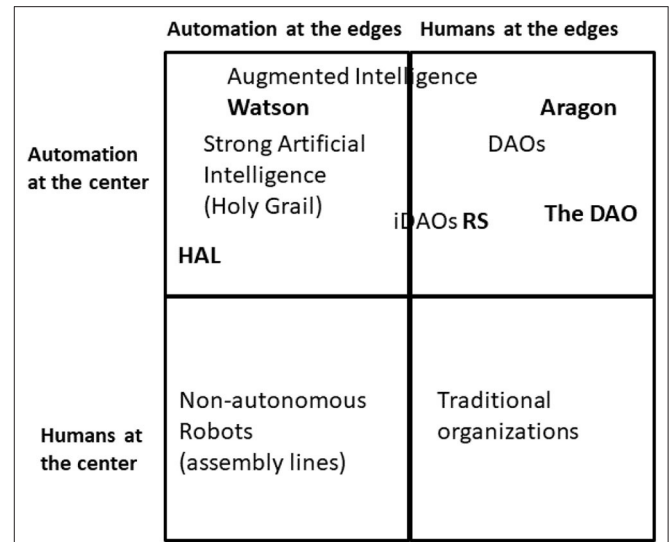


FIGURE 12 | A quadrant of organizations. Adapted from <https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/>

CONCLUSIONS

In this article, we have provided a framework to exploit blockchains and smart contracts as enabling technologies for an innovative type of supply chain management, aimed at achieving higher levels of collaboration between the companies participating in the chain. To this end, we have introduced the concept of Intelligent Smart Contract, namely a contract executable on the blockchain and characterized not only by “smartness,” that is, automated execution on blockchain, but also by “intelligence,” that is planning and optimization. We have then illustrated a specific instance of an intelligent smart contract, both from an algorithmic and an architectural standpoint, as given by Revenue Sharing, a methodology for supply chain management that can greatly boost the profitability of the supply chain for the benefit of all participants, but must be fully automated to achieve its full potential.

An RS-based intelligent smart contract takes advantage of the digital trust guaranteed by the blockchain to replace human trusted third parties in coordinating the various parties in the management of a supply chain, thus making it much easier and less expensive to set up supply chains based on peer-level participation. In addition, it defines and executes plans for bringing the products of the supply chain to the market with the highest returns and fairly distributes the revenues from the proceeds of sale to all participants. The net result will be the greater good that follows from better business ecosystems, with a fairer share of returns among participating companies, greater collaboration and communication between participants, better quality of results, and overall benefits that will extend to the community of stakeholders at large.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

PB and HS cooperated in defining the process and the guidelines for implementation and wrote section Intelligent Smart Contracts for Supply Chain Coordination. NG and GM developed the algorithm and wrote section Algorithmic Supply Chain Coordination together with HS. RP originated the proposal of a blockchain-based implementation, wrote sections Introduction, Blockchains, (Intelligent) Smart Contracts and Digital Trust, Discussion and Future Work, and Conclusions and coordinated the writing of the whole article. EA had the original intuition for cooperative supply chains. All authors collaborated in revising the whole text.

REFERENCES

- Albrecht, M. (2009). "Supply chain coordination mechanisms—new approaches for collaborative planning." in *Series Lecture Notes in Economics and Mathematical Systems, Vol. 628* (Berlin; Heidelberg: Springer), 5–34. doi: 10.1007/978-3-642-02833-5_2
- Bannon, S. (2016). *The Tao of The DAO or: How the Autonomous Corporation is Already Here*. Available online at: <https://techcrunch.com/2016/05/16/the-tao-of-the-dao-or-how-the-autonomous-corporation-is-already-here/> (accessed November 7, 2020).
- Berg, C., Davidson, S., and Potts, J. (2019). Blockchain technology as economic infrastructure: revisiting the electronic markets hypothesis. *Front. Blockchain* 2:22. doi: 10.3389/fbloc.2019.00022
- Bitran, G. R., Gurumurthi, S., and Sam, S. L. (2007). The need for third-party coordination in supply chain governance. *MIT Sloan Manag. Rev.* 48, 30–37.
- Cachon, G. P., and Lariviere, M. A. (2000). Supply chain coordination with revenue-sharing contracts: strengths and limitations. *Manag. Sci.* 51, 30–44. doi: 10.1287/mnsc.1040.0215
- Chen, J.-M., Cheng, H.-L., and Lin, I.-C. (2011). On channel coordination under price-dependent revenue-sharing: can eBay's fee structure coordinate the channel? *J. Operat. Res. Soc.* 62, 1992–2001. doi: 10.1057/jors.2010.174
- Choi, T. (2020). Supply chain financing using blockchain: impacts on supply chains selling fashionable products. *Ann. Operat. Res.* doi: 10.1007/s10479-020-03615-7. [Epub ahead of print].
- Coase, R. H. (1937). The nature of the firm. *Economica*. 4, 386–405. doi: 10.1111/j.1468-0335.1937.tb00002.x
- De Kok, A. G., and Graves, S. C. (2003). "Supply chain management: design, coordination and operation," in *Handbooks in Operations Research and Management Science*, Vol. 11 (Elsevier, B.V.).
- Di Ciccio, C., Ceconi, A., Dumas, M., García-Bañuelos, L., López-Pintado, O., Lu, Q., et al. (2019). Blockchain support for collaborative business processes. *Inform. Spektrum* 42, 182–190. doi: 10.1007/s00287-019-01178-x
- Ferrucci, D. A., Brown, E. W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., et al. (2010). Building watson: an overview of the deepQA project. *AI Mag.* 31, 59–79. doi: 10.1609/aimag.v31i3.2303
- Garriga, M., Dalla Palma, S., Arias, M., De Renzis, A., Pareschi, R., and Tamburri, D. (2020). Blockchain and cryptocurrencies: a classification and comparison of architecture drivers. *Concurrency Comput.* doi: 10.1002/cpe.5992. [Epub ahead of print].
- Giannoccaro, I., and Pontrandolfo, P. (2004). Supply chain coordination by revenue sharing contracts. *International. J. Prod. Econ.* 89, 131–139. doi: 10.1016/S0925-5273(03)00047-1

FUNDING

Work partially supported by Sapienza, project Consistency problems in distributed and concurrent systems, 2017–2019.

ACKNOWLEDGMENTS

We thank Claudio Facciabene and Riccardo Garbo for providing a basis for the implementation of the Web application. We are grateful to Pietro Terna for many interesting discussions about the role of planning in economy, and to Hervé Gallaire for many helpful and insightful comments on previous versions of the article.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fbloc.2020.535787/full#supplementary-material>

- Gong, D., Liu, S., Tang, M., Ren, L., Liu, J., and Liu, X. (2018). Revenue sharing or profit sharing? An internet production perspective. *Adv. Prod. Eng. Manage.* 13, 81–92. doi: 10.14743/apem2018.1.275
- Hayrutdinov, S., Saeed, M., and Rajapov, A. (2020). Coordination of supply chain under blockchain system-based product lifecycle information sharing effort. *J. Transp.* 2020, 1–10. doi: 10.1155/2020/5635404
- Jensen, M. C., and Meckling, W. H. (1976). Theory of the firm: managerial behavior, agency costs and ownership structure. *J. Finan. Econ.* 3, 305–360. doi: 10.1016/0304-405X(76)90026-X
- Johanson, J., and Mattsson, L.-G. (1987). Interorganizational relations in industrial systems: a network approach compared with the transaction-cost approach. *Inter Organ. Stud. Manag. Organ.* 17, 34–48. doi: 10.1080/00208825.1987.11656444
- Korpela, K., Hallikas, J., and Dahlberg, T. (2017). "Digital supply chain transformation toward blockchain integration," in *Proceedings of 50th Hawaii International Conference on System Sciences* (Honolulu). doi: 10.24251/HICSS.2017.506
- Korsten, K. (2019). *A blockchain-based supply chain contract-an analysis to the development of a supply chain contract on a blockchain* (M.Sc. Thesis). Eindhoven: University of Eindhoven.
- Litke, A., Anagnostopoulos, A., and Varvarigou, T. (2019). Blockchains for supply chain management: architectural elements and challenges towards a global scale deployment. *Logistics* 3, 1–5. doi: 10.3390/logistics3010005
- Liu, L., Li, F., and Ki, E. (2019). Research on risk avoidance and coordination of supply chain subject based on blockchain technology. *Sustainability* 11:2182. doi: 10.3390/su11072182
- Malone, T. (1987). Modeling coordination in organizations and markets. *Manag. Sci.* 33, 1317–1332. doi: 10.1287/mnsc.33.10.1317
- Malone, T., Yates, J., and Benjamin, R. (1987). Electronic markets and electronic hierarchies. *Commun. ACM* 30, 484–497. doi: 10.1145/214762.214766
- Malone, T., Yates, J., and Benjamin, R. (1989). The logic of electronic markets. *Harvard Business Review*, p. 166–169.
- Marchesi, M., Marchesi, L., and Tonelli, R. (2018). "An agile software engineering method to design blockchain applications," in *Proceedings of 14th Central and Eastern European Software Engineering Conference Russia (CEE-SECR '18)* (New York, NY: ACM), 1–8. doi: 10.1145/3290621.3290627
- Massa, G., Tononi, R., Arcuri, E., and Areste, F. (2010). "A system of chain management aimed at high levels of collaboration through increase levels of chain efficiency," in *Proceedings of 9th Wageningen International Conference on Chian and Network Management. Conference Bulletin* (Wageningen).

- Massa, G., Tononi, R., Raimondi, R., and Spagna, G. (2007). *La Supply Chain Integrata*. Energia, Ambiente ed Innovazione. p. 48–63.
- Min, S., Zacharia, Z. G., and Smith, C. D. (2019). Defining supply chain management: in the past, present, and future. *J. Bus. Logist.* 40, 44–55. doi: 10.1111/jbl.12201
- Munson, C., Rosenblatt, M., and Rosenblatt, Z. (1999). The use and abuse of power in supply chains. *Bus. Horiz.* 42, 55–65. doi: 10.1016/S0007-6813(99)80049-4
- Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Academic Press.
- Omran, Y., Henke, M., Heines, R., and Hoffmann, E. (2017). “Blockchain-driven supply chain finance: towards a conceptual framework from a buyer perspective,” in *Proceedings of IPSERA* (Budapest).
- Pappas, J. L., and Brigham, E. F., and Hirschey, M. (1983). *Managerial Economics, Chapt. 10: The Firm's Price/Output Decision*. Fort Worth, TX: CBS College Publishing; The Dryden Press. p. 374–378.
- Phillips, L., and Rozworski, M. (2019). *The People's Republic of Walmart. How the World's Biggest Corporations are Laying the Foundation for Socialism*. Brooklyn, NY: Verso.
- Qian, G.-X., Zhang, Y.-P., Wu, J.-G., and Pan, Y.-H. (2013). Revenue sharing in dairy industry supply chain—a case study of Hohhot, China. *J. Integr. Agric.* 12, 2330–2309. doi: 10.1016/S2095-3119(13)60585-7
- Sridharan, R., and Simatupang, T. (2013). Power and trust in supply chain collaboration. *Int. J. Value Chain Manag.* 7, 76–96. doi: 10.1504/IJVC.2013.057344
- Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, p. 2. doi: 10.5210/fm.v2i9.548
- The Economist (2019). Can technology plan economies and destroy democracy? *The Economist*. Available online at: <https://www.economist.com/christmas-specials/2019/12/18/can-technology-plan-economies-and-destroy-democracy> (accessed December 19, 2019).
- Tononi, R., and Amorosi, G. (2002). “Experiences gained applying concurrent engineering tools to networks of SMEs,” in *Proceedings of 8th International Conference on Concurrent Enterprising* (Nottingham: University of Nottingham), 47–54.
- Tran, A. B., Qinghua, L., and Weber, I. (2018). “Lorikeet: a model-driven engineering tool for blockchain-based business process execution and asset management,” in *16th International Conference on Business Process Management* (Ultimo, NSW: Springer).
- Treiblmaier, H. (2018). The impact of the blockchain on the supply chain: a theory-based research framework and a call for action. *Supply Chain Manage. Int. J.* 2, 545–559. doi: 10.2139/ssrn.3224145
- Wang, Y., Jiang, L., and Shen, Z.-J. (2004). Channel performance under consignment contract with revenue sharing. *Manage. Sci.* 50, 34–47. doi: 10.1287/mnsc.1030.0168
- Wernerfelt, B. (1984). A resource-based view of the firm. *Strateg. Manage. J.* 5, 171–180. doi: 10.1002/smj.4250050207
- Williamson, O. E. (1985). *The Economic Institutions of Capitalism*. New York, NY: Macmillan.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Bottoni, Gessa, Massa, Pareschi, Selim and Arcuri. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.