Contents lists available at ScienceDirect

Blockchain: Research and Applications

journal homepage: www.journals.elsevier.com/blockchain-research-and-applications

Research Article

Blockchain-as-a-Service and Blockchain-as-a-Partner: Implementation options for supply chain optimization

Paolo Bottoni [a], Claudio Di Ciccio [a], Remo Pareschi [b,*], Domenico Tortola [c], Nicola Gessa [d], Gilda Massa [e]

[a] Department of Computer Science, Sapienza University of Rome, Viale Regina Elena, 295, Rome, 00161, RM, Italy
[b] Stake Lab, University of Molise, C.da Fonte Lappone, Pesche, 86090, IS, Italy
[c] Department of Bioscience and Territory, University of Molise, C.da Fonte Lappone, Pesche, 86090, IS, Italy
[d] TERIN-SEN-CROSS, ENEA - Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile, Via Martiri di Monte Sole 4, Bologna, 40129, BO, Italy
[e] TERIN-SEN-CROSS, ENEA - Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile, Piazzale Enrico Fermi 1, Portici, 80055, NA, Italy

ARTICLE INFO

Keywords:
Supply chain management
Decentralized autonomous organization
Income sharing
Smart contract
Distributed ledger
Hyperledger fabric

ABSTRACT

Smart contracts show a high potential to make supply chain management strategies epochally leap towards higher levels of productivity, not only in the functioning of production processes but also in terms of product innovation and overall economic returns. This article illustrates the principle of Income Sharing as a highly performing economic strategy for supply chains with a natural implementation in blockchain smart contracts. It proposes a blockchain-based architecture that uses smart contracts to implement various algorithmic versions of the Income Sharing principle among companies participating in a supply chain. The formation of the total income and its consequent redistribution are calculated taking into account the role of the technological platform automating these procedures, which therefore becomes a party to the inter-company business project of a supply chain in the alternative roles, as feasible in business practice, of Blockchain-as-a-Service and Blockchain-as-a-Partner. The approach is implemented on Hyperledger Fabric, the most widespread platform for private and consortium blockchains. We compare and justify this design choice with the alternative given by public blockchains, with specific attention to Ethereum.

1. Introduction

The advent of blockchains and Distributed Ledgers (DLs) has brought to the fore, in addition to cryptocurrencies, highly innovative business models such as Decentralized Autonomous Organizations (DAOs) and Decentralized Finance (DeFi). Although these models were designed in the first place for virtual companies, they can be profitably exported to the digital transformation of the traditional economy to contribute to implementing programs such as Industry 4.0. For this to happen, they must be applied to business processes inherent in brick-and-mortar companies. Supply Chain Management (SCM) is, from this point of view, a domain of particular interest, as it provides, on the one hand, the basis for decentralized business ecosystems compatible with the DAO model and, on the other hand, an essential component in the management of the physical goods that underlie the real economy.

In this paper, we intend to contribute to this evolution with a general supply chain model based on the principle of Income Sharing (IS), according to which several companies join forces for a specific process or project as if they were a single one, whereby the income is divided per a previously agreed-upon distribution scheme. This approach is more performing and effective than traditional wholesale agreements, which lack coordination among the participants in the supply chain. Moreover, today it is all the more practicable due to the Internet economy and the consequent availability of platforms such as Amazon, Alibaba, Ozon, and e-Bay, which Small and Medium-sized E1nterprises (SMEs) can partner with to convey their offers on markets that were unreachable before the Internet. These platforms are particularly effective at endowing SMEs with e-commerce, marketing, and logistics functionalities essential for competing globally.

* Corresponding author.
E-mail addresses: bottoni@di.uniroma1.it (P. Bottoni), diciccio@di.uniroma1.it (C. Di Ciccio), remo.pareschi@unimol.it (R. Pareschi), d.tortola@studenti.unimol.it (D. Tortola), nicola.gessa@enea.it (N. Gessa), gilda.massa@enea.it (G. Massa).

https://doi.org/10.1016/j.bcra.2022.100119
Received 21 January 2022; Received in revised form 3 December 2022; Accepted 8 December 2022

In implementing the IS model, various options for distributing income must be considered, partly due to the Internet platforms' role in supporting supply chains. At one end, there is the well-known and studied Revenue Sharing (RS) [1], according to which each of the participants in the supply chain, the platform included, bears its costs and obtains a proportional return in the form of sales revenues. However, other feasible options are those in which part of the costs of the participants are borne by the originator of the supply chain or by the platform regarding its services. These costs are then deducted from the distribution of the proceeds[1] so that it seems appropriate to speak in this case of Profit Sharing (PS).

As shown in Ref. [2], criteria exist for quantifying, based on volumes of goods handled, number of participants, and stages of the supply chain, the preferability of the various options for the supply chain stakeholders according to their roles. We can assume that the option to pursue is evaluated and negotiated on a project-by-project basis. Therefore, the full range of options should be available in an IS implementation. Furthermore, it is not always easy to apply the management level of the selected IS option to the supply chains of SMEs, where trust and transparency are often lacking. In many cases, Internet platforms act as arbitrators in proposing IS options to SMEs and managing the execution of agreed distribution schemes. The downside, however, is that they can favor solutions to their advantage as they participate in income distribution.

With their deployment on blockchains or DLs, smart contracts may well be a panacea for these issues. They can be used to calculate the costs that need to be incurred by the participating companies and thus distribute income, as these activities can be conveniently automated, provided that all interested parties have visibility of, and consequent confidence in, the algorithm used for their execution. Internet platforms could themselves provide Income Sharing as an additional service, as befits their role as promoters of economic agreements among the participants in the supply chain; by adopting Income Sharing, they would trade decision-making arbitrariness for the fairness and trust that derive from income distribution agreements algorithmically executed on the blockchain, thus incurring a loss of bargaining power that would be compensated by an increase in trust in cooperative relationships. Indeed, by making the management algorithm operate as a smart contract, we have a stringent and satisfactory response to these requirements: algorithmic automation substantially mitigates the costs and complications deriving from an additional level of human management that would otherwise be necessary, and transparency resulting from deployment on blockchain or DL ensures the trustability of such an automated management layer.

The Income Sharing platform is the first recipient of criteria of choice among Income Sharing schemes, as its provision is chargeable according to various options. Therefore, this article focuses on explaining and articulating the economic aspects of using a blockchain-based Income Sharing platform and associating them with choice criteria ranging from Revenue Sharing to Profit Sharing. These criteria are transferable in future developments to an expanded platform to support further aspects of service to a production chain, such as marketing, e-commerce, and logistics. Thus, our effort goes toward laying the foundations for an intercompany information system architecture that finds a natural application in supply chains and, more generally, business ecosystems. Specifically, our approach treats SCM itself as a supply chain component. We will demonstrate how such an SCM component fits with Revenue Sharing by being realized as Blockchain-as-a-Service (BaaS), a well-known model studied in the blockchain community [3]. We will also introduce the model of Blockchain-as-a-Partner (BaaP) and demonstrate its realization with Profit Sharing.

Keeping in mind that IS is a general scheme that groups the more specific options provided by RS and PS, engineering the optimal IS should provide core functionalities amenable to specialization. To this end, we design a modular architecture by leveraging an algorithmic basis encompassing all viable options. We illustrate its implementation on Hyperledger Fabric, the most successful platform for implementing private/consortium blockchains, and highlight the advantages of this choice over a public blockchain such as Ethereum. The reasons in favor of Fabric rather than Ethereum extend more generally to opting for private/consortium *vs.* public blockchains. The evidence we will bring to support goes beyond the usual considerations relating to privacy in managing information, which undoubtedly represents a point in favor of the private option. Even more crucial is that public blockchains do not lend themselves to models in which the economic role of the blockchain must be commensurate with the specific needs of an inter-organizational entrepreneurial project as a supply chain is. However, a relationship with public blockchains can be established if necessary through hybrid (i.e., private + public) blockchains, according to an increasingly widespread approach [4–6].

The contribution of this article builds on the two previous works [7, 8], which illustrated, respectively, the implementability on the blockchain of a smart contract corresponding to the Revenue Sharing algorithm, which is one of the possible options for the general method of Income Sharing; and the transferability of the implementation of Ref. [7], carried out on Ethereum, to consortium DLs as buildable through Hyperledger Fabric.

The rest of this paper is organized as follows. After providing extensive background in Section 2, we present a motivating scenario and a general description of our approach in Section 3, also providing details about the specific calculations of income sharing. Then, Section 4 illustrates our proposal for flexible management of smart contracts through the specific options relative to financing the supply chain and sharing income and draws out the processes by which participants cooperatively introduce the information needed to run the selected algorithm, illustrating them with examples of concrete interactions. Finally, Section 5 presents related work and discusses implementation choices and tradeoffs, and Section 6 concludes the paper.

2. Background

The video rental market acted as an incubator for the first uses of Revenue Sharing in the late 1990s, through the efforts of Blockbuster, then the industry leader, to exploit it to stimulate a steady growth in revenues, as described by Dana and Spier in Ref. [9]. It was then systematically investigated in the seminal article by Cachon and Lariviere [1]. An early study involving an Internet platform in the setup and run of an RS supply chain is provided by Wang et al. [10], aimed at investigating the effect of RS on the performance of a sales channel where a supplying company uses Amazon for e-retail and logistics, thus finding out that performance, both of the overall channel and the individual firm, depends on demand price elasticity as well as on the retailer's share of the channel cost. Qian et al. [11] provided a case study in the Chinese dairy sector, where several structural problems, including an unbalanced allocation of profits along the supply chain in favor of retailers (supermarkets) and to the disadvantage of farmers and producers, are addressed by applying the influential three-stage RS model by Giannoccaro and Potrandolfo [12], with an increase in the overall profitability of 12.49%. The algorithmic RS methodology provided by Tononi et al. [13] is the cornerstone of the architecture described here, mainly because it lends itself easily to implementation and is at the same time highly modular and flexible so that variations such as Profit Sharing can be straightforwardly integrated, as will be detailed in Section 3.4. Moreover, it explicitly addresses the problem of trust by giving the best results if the supply chain participants are discouraged from showing production costs higher than the real ones, and none enjoy economic and informative privileges. For these reasons, it perfectly fits with blockchains and DLs.

In the agricultural sector, the practical application of this

[1] Assuming all participants are subject to the same fiscal discipline, the distribution quotas do not depend on the level of taxation. Otherwise, the different taxations can be reflected in the structure of alignment costs (see Section 3.3).

methodology, albeit in a phase of pre-operational technological development, i.e., at the simulation level, was among the results of the LEMURE (Logistics integrates MUltiagent for SME networks) Project[2]. Focused on the tomato processing chain, it enjoyed improvements of up to 17% in overall profitability. Its full-fledged implementation in the form of a smart contract operating on blockchain/DL technologies is one of the objectives of the project WEBEST (Wine EVOO Blockchain Et Smart ContracT).[3]

On the Profit Sharing side, Çanakoğlu and Bilgic [14] analyzed the performance over multiple periods of a two-stage telecommunications supply chain consisting of an operator and a vendor and, for optimization purposes, suggested a PS contract in which companies share both revenues and operating costs. Wei and Choi [15] illustrated an industrial practice of PS in the apparel sector, based on which they explored the use of a wholesale pricing and profit-sharing scheme for coordination of the supply chain according to the criteria of mean-variance. Both of these contributions are part of the background used by Gong et al. [2] to define selection criteria between RS and PS depending on the characteristics of the supply chain and the role of the participants.

As shown in the following sections, there is a direct congruence between Revenue Sharing and the technological and economic concept of BaaS, which is widely known and systematized in the blockchain community [3]. Less investigated is the case of BaaP, in which the service provider participates in the business risk of a supply chain project by anticipating or sharing the costs of other participants. Yet, it is a realistic option in an SCM context: think, for example, of a bank that operates both as a provider of the blockchain-based coordination platform and as a lender for the costs associated with its use. As regards both BaaS and BaaP, the fact that centrally organized entities, like banks and e-commerce platforms, provide access to decentralized technologies such as blockchains and DLs is neither paradoxical nor contradictory. On the contrary, it can be attributed to the recognition that decentralizing, automating, and making decisions transparent pay off through higher returns for all participants in the supply chain.

Initiatives such as FoodTrust [16] and TradeLens [17] provide examples of this flexibility in the interplay between centralized and decentralized worlds. In both cases, software giant IBM teamed up with leaders in large-scale distribution, such as Carrefour and Walmart, as well as in transport logistics, such as Maersk, in implementing blockchain-based platforms aimed at strengthening and streamlining certification and documentation practices in both sectors. Other companies then extended these partnerships through various agreements with the platforms' originators.

Compared to these efforts, which are limited to improving the quality of processes, our framework makes a quantum leap in two respects: it goes straight to the heart of the mechanisms behind income generation, and it provides methods for calculating and comparing the returns deriving from the various optimizations. Therefore, the supply chain optimizations given by Revenue Sharing and Profit Sharing provide the background for the formalizations and implementations that will be illustrated in the next sections. For clarity's sake, we summarize them as follows:

- **Revenue Sharing**: the sharing of costs equitably and proportionally among the participants in the supply chain with the postponement of the returns deriving from internal orders at the time of redistribution of the proceeds deriving from the sale of the product/service on the market;
- **Profit Sharing**: pushing the principle of postponement of returns characteristic of Revenue Sharing further on through the anticipation or financing by one of the participants of the costs, in whole or in part, of other participants.

It follows that the different Profit Sharing options can be seen as variations of the primary mechanism of Revenue Sharing; we refer to this set of options, including the basic one of Revenue Sharing, as **Income Sharing**, a concept that thus generalizes both Revenue Sharing and Profit Sharing.

On the entrepreneurial level, there can be various reasons behind the choice of Profit Sharing: from the returns deriving from financing operations to the facilitation of the creation of supply chains with high commercial capacity that may not be formed due to the lack of financial means of some of the participants. The entrepreneurial risk of Profit Sharing is greater than that of Revenue Sharing. While in one case, negative economic returns are reflected only in the impossibility of recovering one's costs, in the other, they also imply losses deriving from financial exposure.

3. A view on supply chain management

Fig. 1 depicts a standard map of the processes involved in a supply chain [18]. Our approach concerns the first and last stages in the chain (colored in blue in the figure): (1) the building up of the supply chain, during which the criteria determining the demands, prices, costs, production means, and income sharing are defined, and (2) the sharing of the income.

The production and marketing stages in between are out of the scope of this paper.

Note that the beginning stage is critical, as it involves decisions (the selection of the core criteria, the setting of the income sharing scheme, and the subsequent assembly of the chain) that have a reentrant effect on the ex-post phase.

In the remainder of this section, we introduce a motivating scenario and provide a high-level view of the Income Sharing approach, highlighting service providers' special role. We then detail an algorithmic methodology for Income Sharing and show how it fits with instances of Revenue Sharing and Profit Sharing, as well as BaaS and BaaP.

3.1. *A motivating scenario*

In many production environments, behind every product and service, several companies are dedicated to the different activities that, together, are necessary to translate primary resources into a product or service, such being, indeed, the essence of supply chains. In a productive sector comprised of various SMEs, such as the agri-food area, the contribution of the individual companies is fundamental. However, the operational capacity of the chains they participate in implies a higher level of complexity. Problems and decisions may require the whole consortium to be involved since a single company may be unable to solve them. Therefore, solving the problems of chain management and integration becomes fundamental to making them increasingly efficient [18].

For example, agri-food chains are plagued by problems such as the sharp rise in the prices of primary production and excessive intermediation. These issues are tightly connected. One typically observes a lack of both “vertical” and “horizontal” coordination. The former refers to the relationship between supplier and customer, and the latter refers to the cooperation between suppliers of the same resource. However, the low level of integration is, in turn, caused by the difficulty of finding a company that can effectively and authoritatively cover the role of central coordinator among the SMEs in the chain [19]. “Authoritative” here means being able to constrain the behavior of all the involved companies to make them act in the general interest of the chain. We present a system implementing the IS approach to provide an answer to this problem.

The LEMURE project cited above implemented and applied the Revenue Sharing algorithm (part of the IS approach, see Section 3.4) to an agri-food chain for products ready for consumption. The supply chain

[2] Funded by Ministry of Research with Law 297/99, grant agreement 2007/32458.

[3] Funded by the Italian Ministry of Research under the PRIN program: Research Projects of Relevant National Interest—Call 2020.

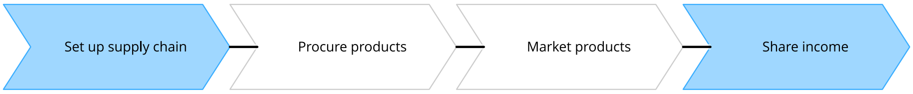


Fig. 1. The process architecture at large.

included a partner dedicated to transforming and processing several raw materials (tomatoes, oil, spices) from multiple suppliers. The production chain also involved elements for packaging coming from other suppliers. The final products were distributed on the market at a price computed in response to the demand.

In the context of a decentralized coordination approach based on the Revenue Sharing algorithm, the production chain was defined by aggregating the costs of the production process (variable costs, fixed costs, production, time of recurrence, and the bill of material) [13]. The solution to optimizing the supply chain relied on constrained linear programming. Data were represented with a matrix structure so that the chosen formulations could be generalized regardless of the length or width of the supply chain under analysis. Similarly, several parameters were used, functional to the definition of the production process as above, and others were defined concerning the income sharing phase (negotiated quota, price, demand of the final product).

The approach thus illustrated is pre-operational, as it does not foresee the costs of the organizational/technological structures necessary to implement the computable optimizations in a real-world supply chain. There are several ways to achieve such an implementation. The most practiced one is when an economically predominant organization elevates itself to the role of implementor and actuator; the burden thus assumed pays back through the authority that goes with it and all the resulting imbalances in contractual and managerial power. Another is to create a human management structure shared by the companies participating in the supply chain, which is a substantially impracticable route due to the inevitable negotiation delays and the implementation complications that would ensue. The technologically innovative way illustrated in Refs. [7,8] hinges, instead, on the automation of management functions through a blockchain-based platform, with costs flexibly shareable according to various schemes. This solution also prevents weight and power imbalances beyond each participant's objective and transparent economic contribution, as contractually defined and provided. In the following sections, we resume the algorithmic treatment of previous publications, enriching it with an essential element, viz., to what extent the cost of the technological platform affects the supply chain's overall performance. The treatment of this aspect lends itself to a variety of implementations, laying the foundation for a generalization of the Revenue Sharing paradigm to the broader one of Income Sharing, which includes, among its variants, Revenue Sharing and various versions of Profit Sharing.

3.2. *Revenue sharing*

Revenue Sharing provides a basic version of Income Sharing because it requires minimal assumptions regarding the underlying organizational structure. We recall here the characteristics of the RS algorithm, revisiting its description in Ref. [7]. The algorithm leverages a series of matrices associated with the supply levels that set up the chain description. In this view, the supply chain is rooted in the request from an *originator*, who advertises the need for several intermediate products and services in given quantities. When a request is announced, available and interested *suppliers* (forming a set M) tender to provide the necessary products and services (collectively called *resources*, in a set K), also relying on other suppliers of resources that precede them in the process.

This establishes a hierarchy of levels, ending with suppliers of raw materials or basic services that are self-sufficient to satisfy a request. We denote levels with an integer $i \in I = [0, n{+}1]$. In a multi-level structure ($n \geq 2$), resources to be provided to the higher level ($i{-}1$) may need other resources from the lower level ($i{+}1$). We adopt the following conventions: we assign level 0 to the market where the final product is sold, we use the levels from 1 to n to describe the actual producers and their products, descending the hierarchy (i.e., following the supply chain) down to suppliers of raw material, and we assign level $n{+}1$ to services of various nature, typically involved with the management of IT platforms or with financing the consortium.

Note that a supplier could operate at different levels by supplying different resources. For example, a farmer could provide fodder for animals to a breeder and vegetable rennet for cheese production in a cheese production process. The structure at the basis of the supply chain is thus represented as a relation $SCS \subseteq I \times K \times M$. Concerning the previous example, the farmer will be involved in the chain at two levels, each provision being represented by a specific node in the chain. The farmer will then receive quotas according to the type of resource and the quantity supplied. Indeed, as discussed in Ref. [13], the construction of the chain is driven by the product resource rather than by supplier identity.

We assume K and M to be finite non-empty sets, so that we can map every $k \in K$ and every $m \in M$ to an integer in the intervals $[0, |K| - 1]$ and $[0, |M| - 1]$, respectively.

In other words, each item in SCS can be identified with the triple indexes (i, k, m), where i corresponds to the level, k is the resource type at a specific level, and m is the supplier of the resource k at level i. We shall also use the (i, k) pair to identify the provision of resource type k at level i, that is, such that there exists $m \in M$ for which $(i, k, m) \in SCS$.

In order to create a request, the originator provides the following parameters:

$d \in \mathbb{N}$: demand for the final product from the market;
$p \in \mathbb{R}^{+}$: price;
$BOM_{i,k} \in \mathbb{R}^{+}$: the *Bill of Material*, mapping each type of resource $k \in K$ in the supply chain at each level $i \in I$ in which k appears to be a quantity—that is, the ratio of the contribution of k to the final product.

The originator can set upper bounds to the number of types of resources to be used ($ress \in \mathbb{N}$, so that $k \leq ress$), the number of levels ($levs \in \mathbb{N}$, hence $i \leq levs$), and the number of participating suppliers ($sups \in \mathbb{N}$, with $m \leq sups$).

A supplier m contributing a resource of type k (at a certain level i) must in turn characterize its contribution with various parameters: $cf_{i,k,m} \in \mathbb{R}^{+}$, fixed production cost; $cv_{i,k,m} \in \mathbb{R}^{+}$, variable production cost; $q_{i,k,m} \in \mathbb{R}^{+}$, quantity of provided resource; $tp_{i,k,m} \in \mathbb{R}^{+}$, time span to cover the cost (in days); $g_{i,k} \in \mathbb{R}^{+}$, income quota (negotiated with other suppliers of k at level i).

3.3. *Services and investors*

In the previous section, we have outlined the input data provided by a supplier, generally summarized in the production costs of the supply chain partner. The IS framework provides for the computation and management of such data on an IT platform, made possible by blockchain/DL and smart contracts, which enables IS automation in a real context. Hence, we need to consider the role of agents whose function is not only (or not at all) productive but also involves the provision or financing of the IT service and see how this is reflected in the originator's configuration of the supply chain, costs, and returns.

In particular, coordination via an IT platform leads us to consider three scenarios in the Income Sharing model:

(1) the distribution of the costs of IT services among all partners: this option coincides with Revenue Sharing, with the blockchain made available in BaaS mode;
(2) the charge of the costs of IT services to the originator, which corresponds to a form of Profit Sharing and is itself compatible with BaaS;
(3) the presence of a new entity for the supply chain in BaaP mode: the IT platform provider. The latter shares both the risk and the final profit of the supply chain, thus implementing another form of Profit Sharing.

Thus, in all cases, we have a way of computing costs and returns in choosing various organizational and economic variants of Income Sharing. In the case where the platform acts as an external service provider (scenario (1)), hence not involved in the production aspect of the chain (as in scenarios (2) and (3)), members of the chain have to bear costs related to aligning themselves to the new process. These alignment costs are then computed and indicated here as *cAll*.

In the RS algorithm, the aim of the alignment matrix is threefold:

(i) alignment of quality characteristics;
(ii) product innovation;
(iii) process innovation (our case).

When the alignment cost occurs, then the *m* member will receive, from the overall proceeds, its compensation as a recovery for these costs.

3.4. Computation of quotas

Based on the structure described in Section 3.2, the algorithm to calculate a fair configuration of Income Sharing (once the final price is realized on the market) proceeds in five steps, which we recap here from Refs. [7,8].

Step 1 For $(i, k, m) \in SCS$, the associated cost $c_{i,k,m}$ is computed as follows:

$$c_{i,k,m} = \frac{cf_{i,k,m}}{tp_{i,k,m} \cdot q_{i,k,m}} + cv_{i,k,m}. \tag{1}$$

Step 2 The value of $c^{min}_{i,k}$ is calculated as the minimum advertised cost $c_{i,k,m}$ among all suppliers for each resource *k* at layer *i*.

$$c^{min}_{i,k} = \min_{\{m \in M:(i,k,m) \in SCS\}} (c_{i,k,m}). \tag{2}$$

Step 3 *R*, the *gross income*, is computed on the basis of the price (*p*) and quantity of products established by the originator. Representing with *m* the request advertised by the originator, we have the following:

$$R = p \cdot \sum_{m \in M} q_{0,0,m}. \tag{3}$$

The share PL_k of revenues to assign to the group of suppliers for resource *k* and the total revenue at level *i*, PL_i, are computed as follows:

$$PL_i = \sum_{k \in K} PL_{i,k}, \quad \text{with} \quad PL_{i,k} = d \cdot c^{min}_{i,k} \cdot BOM_{i,k}. \tag{4}$$

The overall total revenue, *PL*, is as follows:

$$PL = \sum_{i \in I} PL_i. \tag{5}$$

The share of revenues, $PL_{i,k,m}$, is calculated for each supplier *m* of the resource *k* at layer *i* using a coefficient $f_{i,k,m}$, reflecting the contribution of quantity $q_{i,k,m}$ with respect to the sum of contributions for resource *k*:

$$PL_{i,k,m} = PL_{i,k} \cdot f_{i,k,m}, \quad \text{with} \quad f_{i,k,m} = \frac{q_{i,k,m}}{\sum\limits_{m \in M} q_{i,k,m}}. \tag{6}$$

Step 4 The overall profit of the supply chain is calculated taking into consideration the so-called *alignment costs*, i.e., the extra costs that some suppliers of the resource *k* might have incurred with respect to the minimum cost, $c^{min}_{i,k}$. These are typically related to costs required by strongly innovative production processes or investments to keep up with the expected quality standards for the supplied resource. To this end, the parameter $c^s_{i,k,m}$ equates to the minimum advertised cost if the advertised one is higher than that, or 0 otherwise (this deters suppliers from exposing costs due to inefficiencies in their processes). The individual ($cAll_{i,k,m}$) and total (*cAll*) alignment costs are calculated as follows:

$$cAll = \sum_{(i,k,m) \in SCS} cAll_{i,k,m}, \quad \text{with}$$

$$cAll_{i,k,m} = \frac{PL_{i,k,m}}{c^{min}_{i,k}} \left(c_{i,k,m} - c^s_{i,k,m} \right) \quad \text{and} \tag{7}$$

$$c^s_{i,k,m} = \begin{cases} c^{min}_{i,k} & \text{if} \quad c_{i,k,m} > c^{min}_{i,k} \\ 0 & \text{otherwise.} \end{cases}$$

We can now subtract these costs from the total income before proceeding to calculate the sharing of the net income. Hence, denoting with *PC* the net *profit chain*, with *R* and *PL* calculated at step 3, and *cAll* as above, we have the following:

$$PC = R - PL - cAll. \tag{8}$$

Step 5 For each participant, the $PC_{i,k,m}$ share is established based on the quotas of resources provided by each supplier, with respect to each resource *k*, based on previously negotiated requirements $g_{i,k}$, under the following constraint:

$$\sum_{i \in I\ k \in K} g_{i,k} = 1.$$

We denote with ***G*** the matrix consisting of $g_{i,k}$ elements for every $i \in I$, $k \in K$.

Then, each partner receives a specific revenue profit $PC_{i,k,m}$, calculated considering the income quota $g_{i,k}$ (negotiated for every resource) and the specific supplied quantity $f_{i,k,m}$. To this end, we use the following formula:

$$PC_{i,k,m} = f_{i,k,m} \cdot g_{i,k} \cdot PC. \tag{9}$$

Following the scenarios described in Section 3, we now consider the case in which an external service provider supplies the IT platform. Then, the alignment cost can be borne by the originator or divided among all the partners.

This means that $c_{i,k,m} > c^{min}_{i,k}$ and the *m* member should receive the following compensation for the alignment costs:

$$cAll_{i,k,m} = \frac{PL_{i,k,m}}{c^{min}_{i,k}} \cdot \left(c_{i,k,m} - c^s_{i,k,m} \right). \tag{10}$$

Fig. 2 shows the distribution of revenues for the specific supply chain in the use case of the LEMURE project (see Section 2) in the presence of a platform that acts as a service for the chain; this corresponds to BaaS, in the use of the blockchain for the automation of the Income Sharing. In this case, the alignment cost is split among all partners. The costs are divided according to the agreed quotations for income redistribution.

At this stage, the i, k, and m indexes are configured depending on whether the originator or all partners bear the costs. When only the originator provides this cost, assuming that the originator is the second provider ($m = 2$) of resource $k = 1$ at the first level of the chain ($i = 1$), we have:

$$cAll_{1,1,2} = \frac{PL_{1,1,2}}{c^{min}_{1,1}} \left(c_{1,1,2} - c^{s}_{1,1,2} \right). \tag{11}$$

This would be the only value different from 0 in the ***cAll*** matrix. If all partners cover the cost of the IT platform provider instead, then Eq. (10) is applied for each node in the chain. If the IT platform provider is a member of the chain, it takes part of the profit as described in Ref. [2] with a specific income quota. In the revenue sharing model presented in Ref. [8], the ***G*** matrix is used to express the cutting ratio of the different resource providers.

Suppose now that a third-party IT provider joins the consortium. As the matrix indexing scheme is flexible, it can manage new partners, whether in the production chain or at the service level (as in this case, with an IT platform provider). The IT platform provider is not product-oriented. Therefore, following the approach described in Ref. [13], the most suitable solution is to add a final “third party” level under level n. This extra level impacts the ***G*** matrix, where the negotiated requirements are stored. The IS algorithm and the related blockchain implementation can thus easily support the configuration change with a third-party provider simplifying the overall approach and the management of the order request, speeding up working time compared to the other usual collaboration agreements. In this case, we have a new partner with costs related to providing the IT platform, who also participates in sharing the income. Fig. 3 shows the revenue distribution in this setting.

In all cases, we have a rigorous methodology available to calculate different Income Sharing options, included in the spectrum that goes from Revenue Sharing to Profit Sharing, which can be associated, as appropriate, with the economic and organizational models deriving from BaaS and BaaP.

4. Smart contracts for income sharing

We now provide an overall view of our implementation of a system for flexible management of smart contracts for supply chains based on Income Sharing. First, Section 4.1 presents the modular software architecture for IS deployment, where its various options, as well as the role of the coordination platform, ranging from BaaS to BaaP, can be immediately and concretely placed. Then, Section 4.2 presents data structures and aspects of its implementation within Fabric, a framework for implementing DLs for private consortia, which is part of the Hyperledger software ecosystem promoted by the Linux foundation [15].

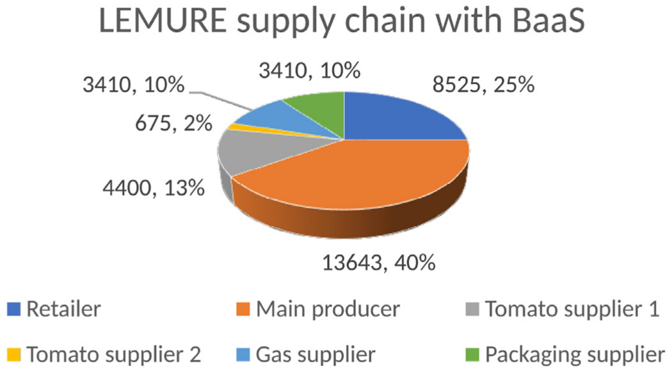


Fig. 2. Revenue distribution when the platform is included in the consortium.

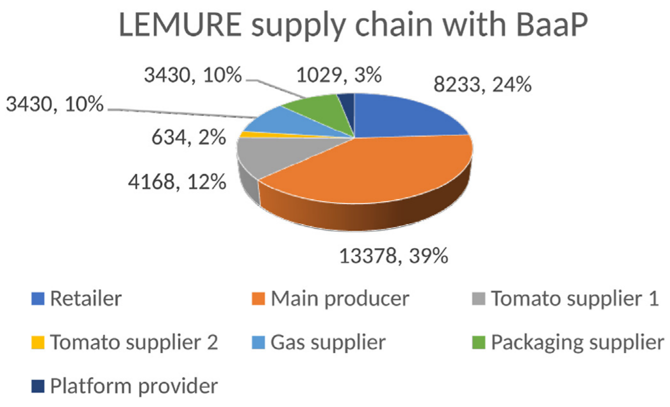


Fig. 3. Revenue distribution when the platform is external to the consortium.

4.1. *A modular perspective on contracts*

Fig. 4 presents a conceptual model of the architecture of our system, supporting the separation of concerns between the description of the production organization in a supply chain and that of its economic structure. The production model is centered on the notions of `Resource`, `Supply`, and `ProductionProfile`, while the economic structure is realized by modeling each `ProductionSupplier` for a given `Supply` in terms of an `EconomicProfile`.

A supply chain is uniquely associated with the Request put forward by an originator and is structured as a collection of `TreeLevels`. In particular, a request is described by attributes corresponding to the parameters discussed in Section 3 (i.e., p for price, d for demand, etc.). Each `ProductionTreeLevel` describes the resources needed to realize a semi-finished product to be used at the level above, up to the finished product to be sold on the market. Each type of resource is characterized by a name, a quota of income to be reserved for the overall production of that resource, the Bill of Material (BOM) describing its contribution to the production, and a list detailing how it will be supplied.

We assume that a supply chain outputs just one type of product (possibly comprised of packaging several products) for the market. This product is seen as described by a request, conceptually thought to be at level 0, thus constituting the root for the tree representation of the supply chain.

The final level is complemented by the information at the only instance of `ServiceLevel`, describing `Services` of different types, each provided by a `ServiceProvider`. We currently envisage that the platform can give access to `FinancialServices`, the provider of which is an investor expecting the investment amount to be remunerated at an income ratio. Other types of services are generic `ITServices`, among which the platform itself, identified by a service Name, accessed through a URI, exposes a cost towards the consortium.

Since the model is agnostic concerning its implementation, we use the dummy type number to indicate that the value is numerical. This corresponds to the type number in an Ethereum implementation (see Ref. [7]), or to floats or integers in a TypeScript/JSON implementation (see Ref. [8]).

The model of Fig. 4 caters to a flexible and extensible realization of consortium supply chains, where the choice of either the Profit Sharing or the Revenue Sharing scheme is orthogonal to the definition of the productive structure. The economy descriptors in the `EconomyProfile` can be used in either scheme to evaluate costs, with the actual remuneration depending on the chosen scheme. The same applies to IT services, which are characterized by alignment costs for the whole consortium. Investors will be remunerated according to the requested

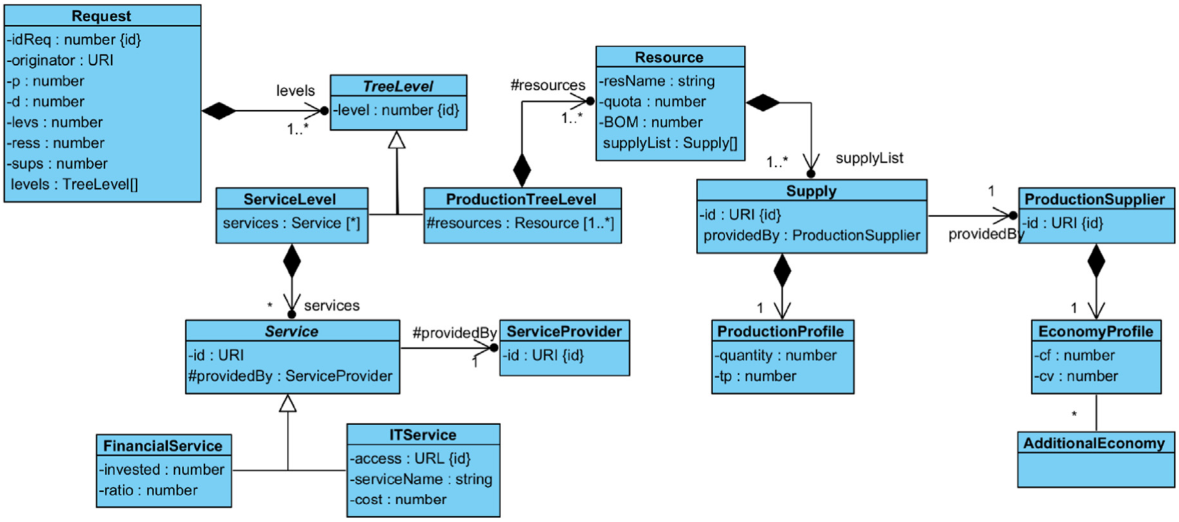


Fig. 4. The conceptual model for the data structure supporting Income Sharing.

ratio on their investment (the latter is also seen as an alignment cost).

Therefore, the actual computation of the amount to be assigned to each participant will consider the whole structure of costs so that the difference between the two currently supported schemes corresponds to either decreasing the income by costs or not. It is immediate to notice that this calls for a realization of the strategy pattern [20] so that the inclusion of a different scheme reduces to providing a different strategy for the scheme and if needed, descriptors in the `AdditionalEconomy` structure useable in that scheme.

4.2. Interaction with smart contracts

We now discuss some aspects of smart contract implementation in Fabric and give an overview of how participants in a supply chain consortium would interact with the platform through interfaces for the specific roles of *originator*, *supplier*, *IT provider*, and *finance provider* (*investor*).

To deploy a smart contract to an active Fabric node in a production environment, a specific procedure must be performed, typically by executing a script directly on one of the Fabric peers that use the contract.

If the peer is deployed as a Docker container, as is standard practice, then it can be accessed through standard communication protocols, such as SSH, to update smart contract files and execute commands via the Command Line Interface (CLI). Fig. 5 shows the status of the Docker container for deploying our system.

In particular, from top to bottom in Fig. 5, we have a “tools” node for interacting in the Fabric network through the CLI, two peers (constituting the network), an orderer node, and three “CA” (Certificate Authority) nodes, one for each peer plus another one for the orderer node.

We have adopted a simplified procedure for deploying smart contracts in our test set and deployed it on all network nodes. In general, however, deployment is limited to the CLI of the interested nodes and does not involve the whole network. The deploy command is parametric with respect to the language in which the contract is coded, the contract name with which it will be encoded, and the path leading to the files with the contract code.

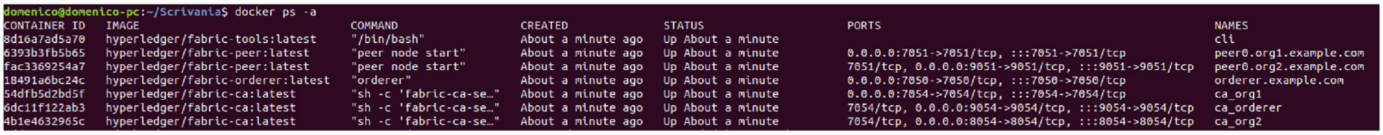
The deployment procedure, also known as chaincode lifecycle, occurs in four steps and includes activities such as naming, versioning, policy definitions, etc.

1. **Code packaging**: the contract code files are packed into a compressed (`.tar.gz`) file to be deployed on a Fabric peer. To this end, using Fabric SDK dedicated commands is the easiest solution, but external tools can also be used. If an external packing tool is employed, the file must be well-formed regarding `config` files, naming, etc.
2. **Chaincode installation**: an admin peer builds the packed code into a chaincode and installs it on the channel. If no error occurs, the chaincode ID is given as a result. Otherwise, an error message is returned.
3. **Chaincode definition approval**: organizations interested in using the chaincode need to approve its definition, which includes chaincode-related parameters that must be consistent across organizations, such as name, version number, contract ID, endorsement policy, and `init` procedure.
4. **Chaincode definition commit**: once the chaincode definition is approved, it can be committed on the channel. To this end, a transaction proposal has to be approved and finally committed by the orderer node.

The Fabric sample network contains an example of an automated script using the `deployCC` function in the `network.sh` script. It executes the deployment on a test setup made by two peers bounded by a single channel. A chaincode update can also be performed using the procedure above: the new code is packed into a `.tar.gz` file, then uploaded and installed. The final commit updates the chaincode ID in the chaincode definition.

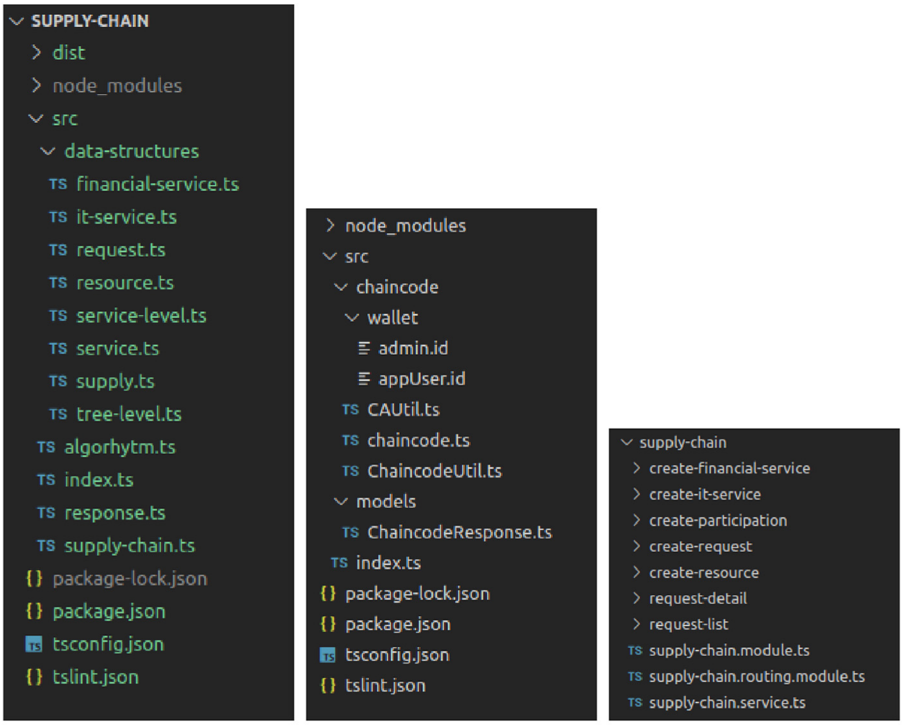
We now turn to aspects of our realization, which is composed of three software modules: the smart contract, coded in TypeScript (see Fig. 6a); a set of APIs developed with TypeScript and NodeJs (see Fig. 6b); a Web application developed with Angular (see Fig. 6c).

In particular, the module realizing the overall smart contract (see



```
domenico@domenico-pc:~/Scrivania$ docker ps -a
CONTAINER ID   IMAGE                                 COMMAND                  CREATED              STATUS              PORTS                                                  NAMES
8d16a7ad5a70   hyperledger/fabric-tools:latest       "/bin/bash"              About a minute ago   Up About a minute                                                          cli
6393b3fb5b65   hyperledger/fabric-peer:latest        "peer node start"        About a minute ago   Up About a minute   0.0.0.0:7051->7051/tcp, :::7051->7051/tcp              peer0.org1.example.com
fac3369254a7   hyperledger/fabric-peer:latest        "peer node start"        About a minute ago   Up About a minute   7051/tcp, 0.0.0.0:9051->9051/tcp, :::9051->9051/tcp    peer0.org2.example.com
18491a6bc24c   hyperledger/fabric-orderer:latest     "orderer"                About a minute ago   Up About a minute   0.0.0.0:7050->7050/tcp, :::7050->7050/tcp              orderer.example.com
54dfb5d2bd5f   hyperledger/fabric-ca:latest          "sh -c 'fabric-ca-se…"   About a minute ago   Up About a minute   0.0.0.0:7054->7054/tcp, :::7054->7054/tcp              ca_org1
6dc11f122ab3   hyperledger/fabric-ca:latest          "sh -c 'fabric-ca-se…"   About a minute ago   Up About a minute   7054/tcp, 0.0.0.0:9054->9054/tcp, :::9054->9054/tcp    ca_orderer
4b1e4632965c   hyperledger/fabric-ca:latest          "sh -c 'fabric-ca-se…"   About a minute ago   Up About a minute   7054/tcp, 0.0.0.0:8054->8054/tcp, :::8054->8054/tcp    ca_org2
```

Fig. 5. The state of the Docker container after deployment.



```
⌄ SUPPLY-CHAIN
  > dist
  > node_modules
  ⌄ src
    ⌄ data-structures
      TS financial-service.ts
      TS it-service.ts
      TS request.ts
      TS resource.ts
      TS service-level.ts
      TS service.ts
      TS supply.ts
      TS tree-level.ts
    TS algorhytm.ts
    TS index.ts
    TS response.ts
    TS supply-chain.ts
  {} package-lock.json
  {} package.json
  TS tsconfig.json
  {} tslint.json
```

(a) The SC module

```
> node_modules
⌄ src
  ⌄ chaincode
    ⌄ wallet
      ≡ admin.id
      ≡ appUser.id
    TS CAUtil.ts
    TS chaincode.ts
    TS ChaincodeUtil.ts
  ⌄ models
    TS ChaincodeResponse.ts
  TS index.ts
{} package-lock.json
{} package.json
TS tsconfig.json
{} tslint.json
```

(b) The API module

```
⌄ supply-chain
  > create-financial-service
  > create-it-service
  > create-participation
  > create-request
  > create-resource
  > request-detail
  > request-list
  TS supply-chain.module.ts
  TS supply-chain.routing.module.ts
  TS supply-chain.service.ts
```

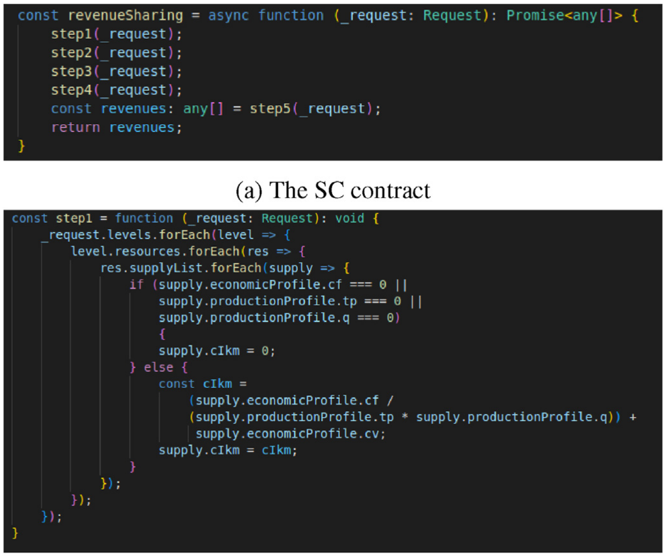
(c) The WebApp module

Fig. 6. The content of the modules composing the application.

Fig. 7a) invokes five submodules, each one corresponding to one of the steps illustrated in Section 3.4. As an example, Fig. 7b shows the realization of the first step.

Fig. 8 illustrates the logical structure of the platform's main system components. The smart contract (SupplyChainSC) runs on the chain (at a HyperLedgerFabricNode). Transactions are submitted via the API, and we realized to convey requests and responses through the Gateway. As the system is blockchain-based, users employ Wallets to transact with it. SCWebApp is the web-based user interface, i.e., the front-end with which all users interact.

The UML sequence diagram in Fig. 9 depicts the handling of a user request. Upon the input of the Participant defined as a request via SCWebApp, the API component checks whether the user has the necessary authorizations to file their request. If this is the case, the Gateway forwards the user input to the HyperLedgerFabricNode, thus triggering the SupplyChainSC. The outcome of the operation is finally returned to the Participant via the user interface.



```
const revenueSharing = async function (_request: Request): Promise<any[]> {
    step1(_request);
    step2(_request);
    step3(_request);
    step4(_request);
    const revenues: any[] = step5(_request);
    return revenues;
}
```

(a) The SC contract

```
const step1 = function (_request: Request): void {
    _request.levels.forEach(level => {
        level.resources.forEach(res => {
            res.supplyList.forEach(supply => {
                if (supply.economicProfile.cf === 0 ||
                    supply.productionProfile.tp === 0 ||
                    supply.productionProfile.q === 0)
                    {
                    supply.cIkm = 0;
                } else {
                    const cIkm =
                        (supply.economicProfile.cf /
                        (supply.productionProfile.tp * supply.productionProfile.q)) +
                         supply.economicProfile.cv;
                    supply.cIkm = cIkm;
                }
            });
        });
    });
}
```

(b) The implementation of the first step

Fig. 7. Aspects of the smart contract implementation.

As discussed before, we assume that the choice of the user configuration of the platform has been made offline during negotiations for forming the consortium. Therefore, the originator has two tasks: (1) to define the overall constraints (e.g., the number of levels) and the top level of the productive structure (i.e., the types of resources and relative quantities to obtain the end product), and (2) to set the platform to operate under the agreed options (Profit or Revenue Sharing, presence of an investor, required IT services).

Fig. 10 shows the form for the specification of the properties characterizing a request. The form is integrated with a dashboard to set options on the sharing scheme and with the presence of external services (see Fig. 11). In particular, the latter lists the possible actions a user can perform on the supply chain tree to add new elements to the structure. The last command starts the execution of the revenue sharing algorithm on the actual supply chain.

Once these options have been chosen, each participant in the system is presented with an interface to enter values for the parameters required by the selected configuration. Typically, the interface provides a field for every descriptor in the production and economy profiles (possibly including the additional economy descriptors) to be set by the supplier.

Fig. 12 presents the interface for a supplier of a semi-finished product (i.e., at an intermediate level in the supply chain). It consists of a form divided into three main sections: in the first section, suppliers can insert data identifying their contribution, while the second one defines the economic profile related to that supply, declaring fixed and variable costs, and the last section must be filled with the production data (supplied quantity and period of cost recurrence).

Figs. 13 and 14 show the interfaces that allow the user to define financial and IT services, respectively. Some descriptors are common to all types of services (e.g., *name*, *ID* of the service as a URL, and *ID* of the provider as a URI). In addition, financial services are defined by the

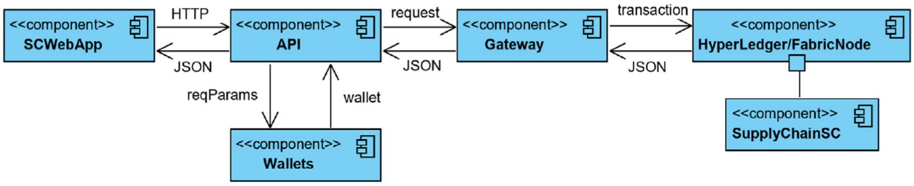


Fig. 8. The component diagram of the system.

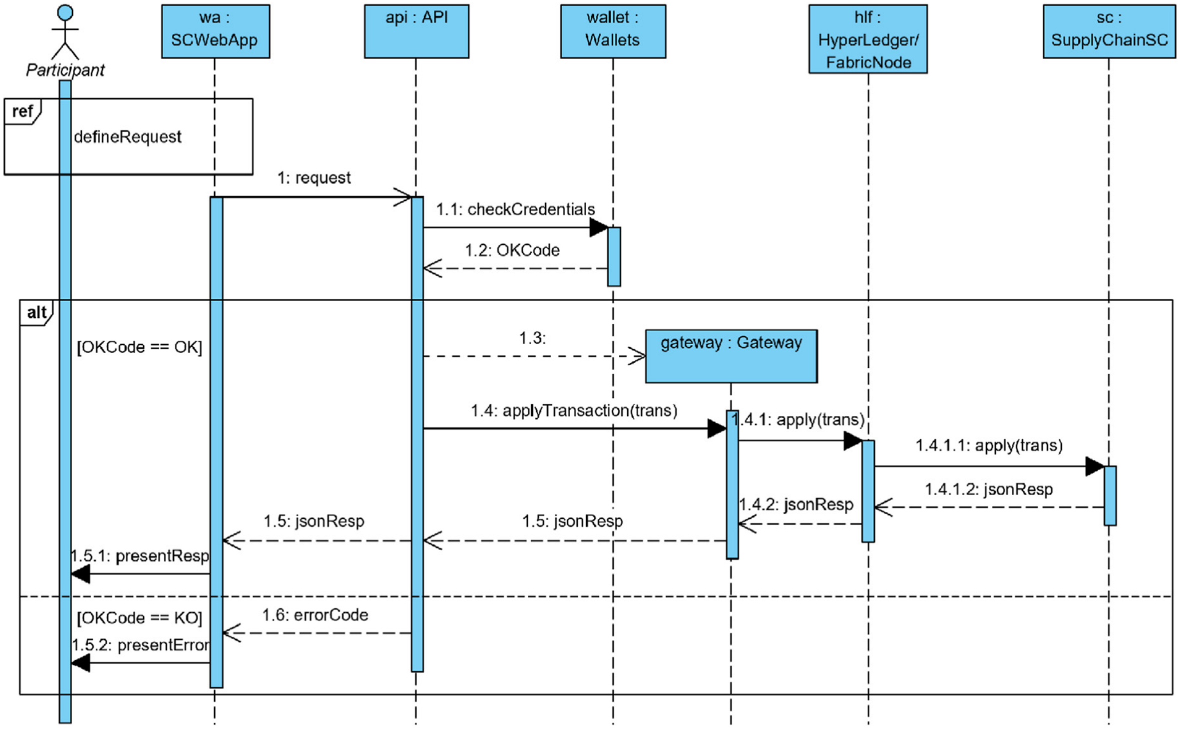
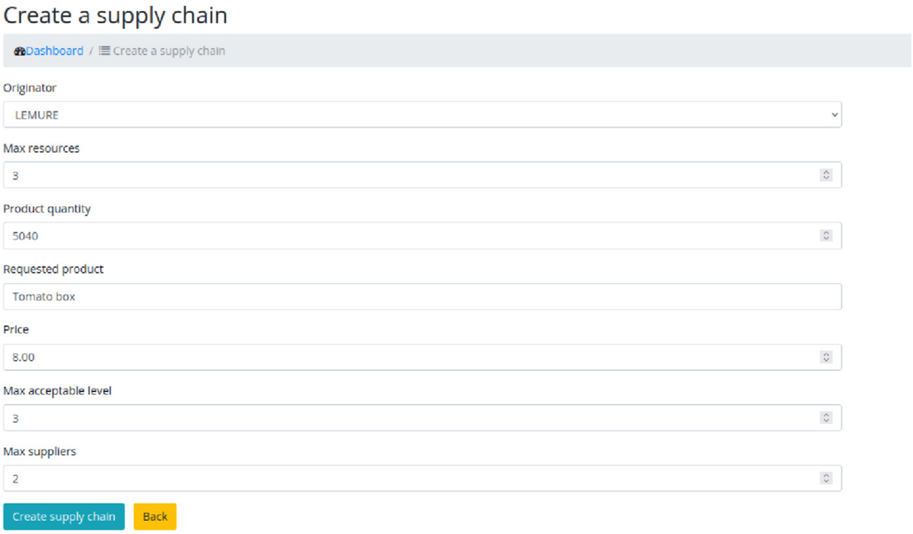


Fig. 9. The interaction of the system components.


Create a supply chain

Dashboard / Create a supply chain

Originator: LEMURE

Max resources: 3

Product quantity: 5040

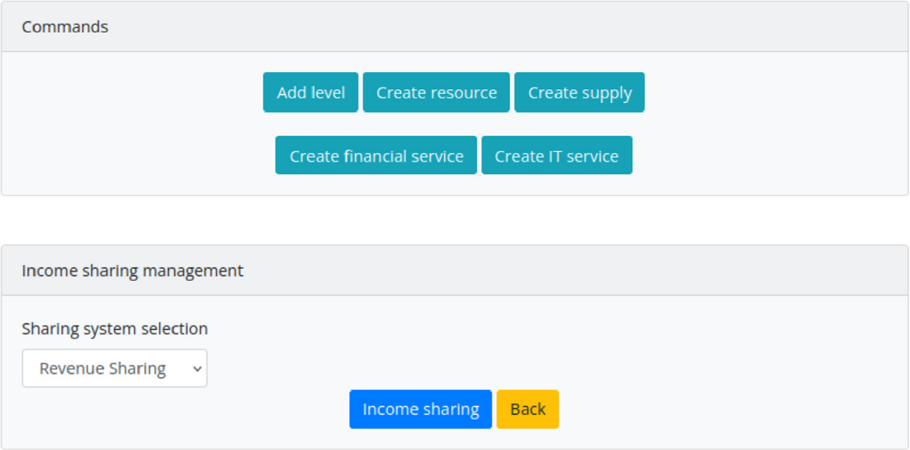
Requested product: Tomato box

Price: 8.00

Max acceptable level: 3

Max suppliers: 2

Fig. 10. Form for the specification of the request properties.



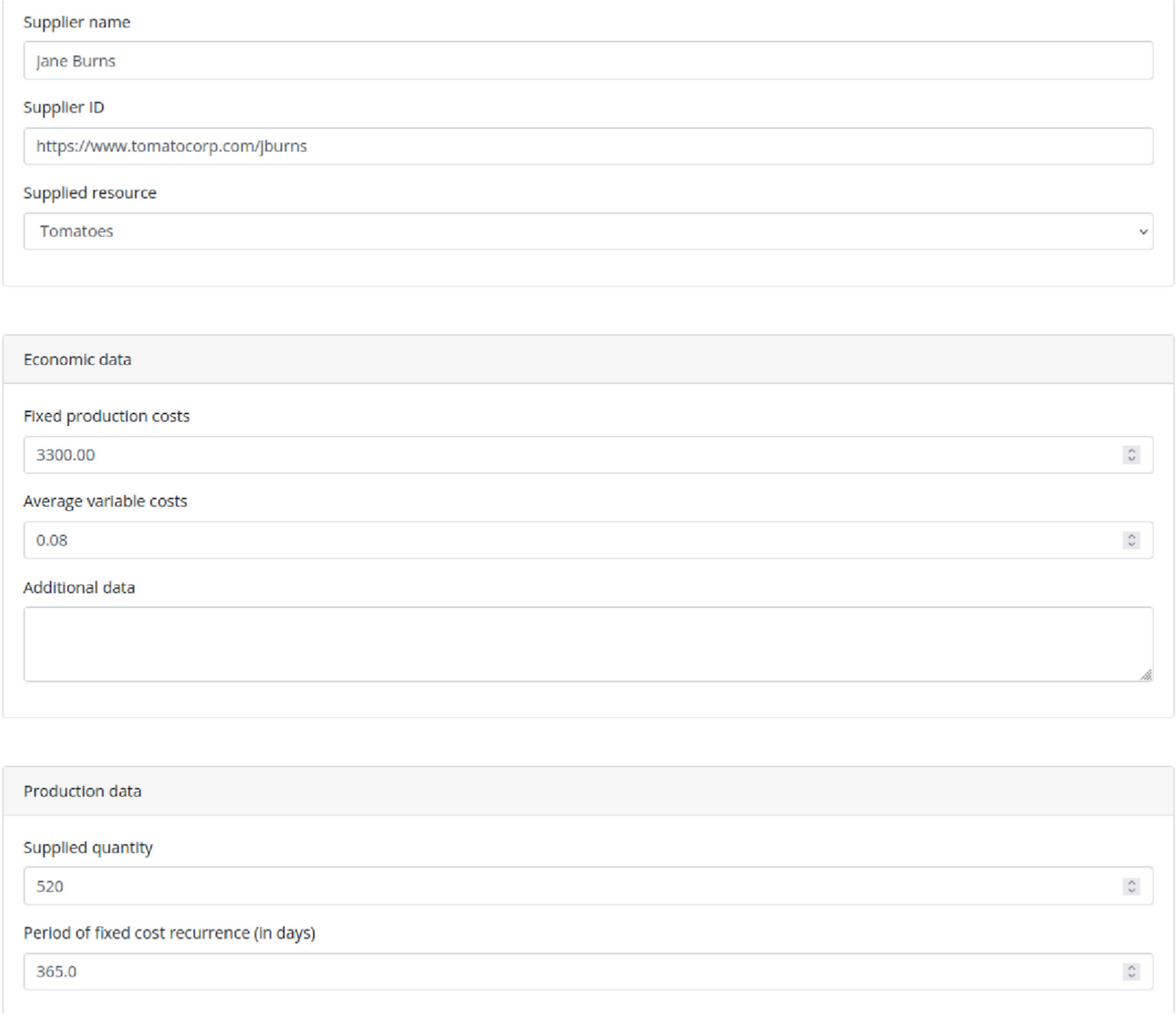
Sharing system selection

Revenue Sharing

Fig. 11. The dashboard for the configuration of the consortium structure.

invested amount of money and the cutting *ratio*, while IT services are defined by their *cost* and the URL for *access*.

When all the suppliers and providers have uploaded their information, the resulting supply chain structure can be stored in a form specified by the JSON scheme of Listing 1 (edited here for the sake of readability). Each class is represented as a structure, scoped by a pair of curly brackets, while square brackets surround lists of elements with a given structure. Numerical types are considered to be implemented as float (associated with the default '0.0') or integers (for which the default is '0'). String types, including URIs and URLs, are set to empty strings if not specified otherwise.


Supplier name

Jane Burns

Supplier ID

https://www.tomatocorp.com/jburns

Supplied resource

Tomatoes

Economic data

Fixed production costs

3300.00

Average variable costs

0.08

Additional data

Production data

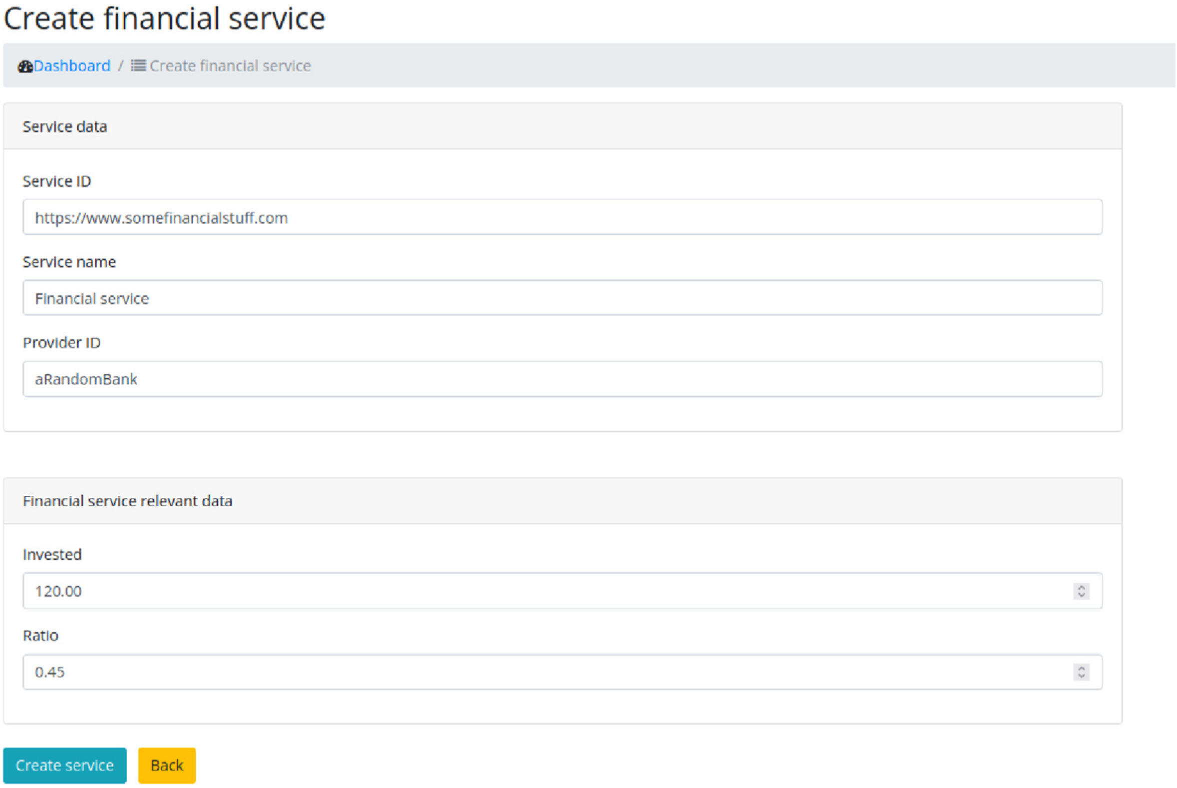
Supplied quantity

520

Period of fixed cost recurrence (in days)

365.0

Fig. 12. The interface for the product supplier.



Create financial service

Service data

Service ID
https://www.somefinancialstuff.com

Service name
Financial service

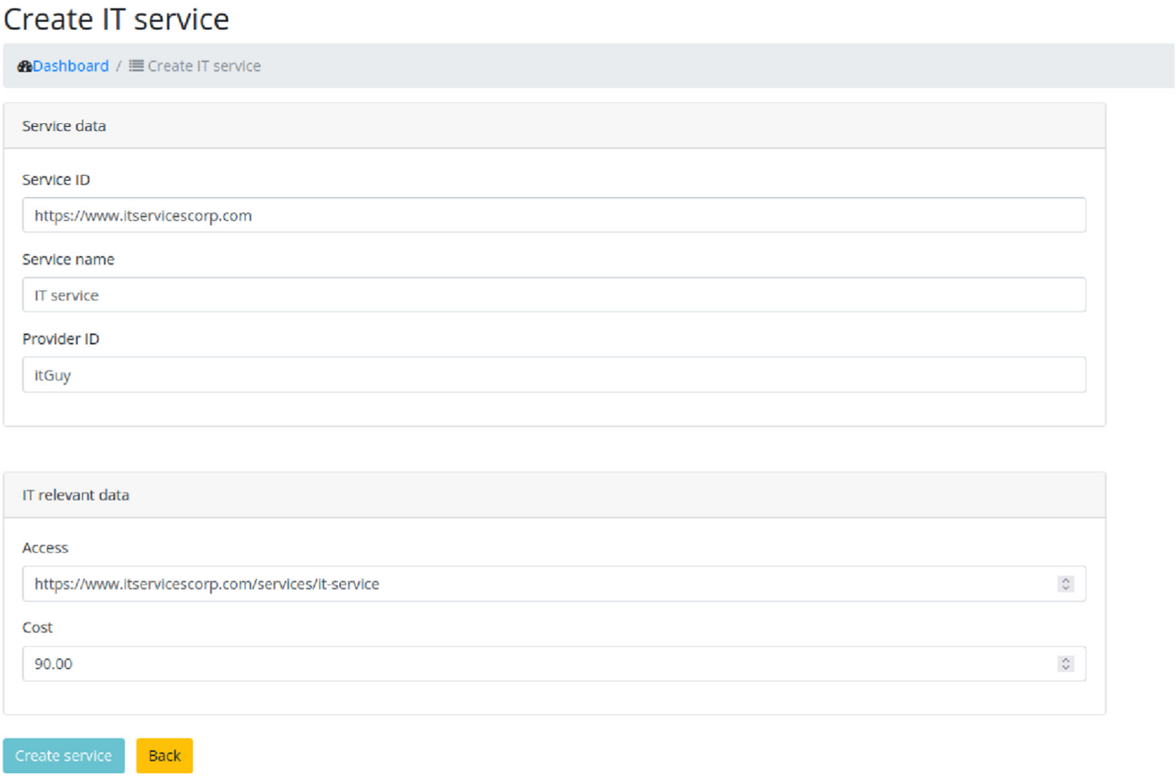
Provider ID
aRandomBank

Financial service relevant data

Invested
120.00

Ratio
0.45

Fig. 13. The interface for providers of financial services.



Create IT service

Service data

Service ID
https://www.itservicescorp.com

Service name
IT service

Provider ID
itGuy

IT relevant data

Access
https://www.itservicescorp.com/services/it-service

Cost
90.00

Fig. 14. The interface for providers of IT services.

```
{
    "id": 1, "originator": "LEMURE", "k": 3, "m": 2, "requestedProduct": "Tomato box",
    "d": 5040, "p": 8, "i": 3,
    "levels": [
        { "level": 1, "resources": [
                {   "resource": "Tomato box", "bom": 0, "g": 0.25, "supplyList": [
                        {   "supplierData": { "supplierName": "London Grocery",
                                "supplierId": "http://www.london-grocery.uk" },
                            "economicProfile": { "cf": 0.02, "cv": 0.002, "additionalData": { } },
                            "productionProfile": { "q": 5040, "tp": 1 } } ] } ] },
        { "level": 2, "resources": [
                 {  "resource": "Mid-transformed products", "bom": 0, "g": 0.4, "supplyList": [
                        {   "supplierData": { "supplierName": "TomatoWorkers SpA",
                                "supplierId": "https://www.tomatoworkers.com" },
                            "economicProfile": { "cf": 222441, "cv": 6, "additionalData": {} },
                            "productionProfile": { "q": 1450, "tp": 30 } } ] } ] },
        { "level": 3, "resources": [
                 {  "resource": "Tomatoes", "bom": 0, "g": 0.15, "supplyList": [
                        {   "supplierData": { "supplierName": "John Doe",
                                "supplierId": "https://www.gas-supply.com/john-doe/" },
                            "economicProfile": { "cf": 2500, "cv": 0.08, "additionalData": {} },
                            "productionProfile": { "q": 3419, "tp": 365 } },
                        {   "supplierData": { "supplierName": "Jane Burns",
                                "supplierId": "https://www.tomatocorp.com/jburns" },
                            "economicProfile": { "cf": 330, "cv": 0.08, "additionalData": {} },
                            "productionProfile": { "q": 520, "tp": 365 } } ] },
                 {   "resource": "Gas", "bom": 0, "g": 0.1, "supplyList": [
                        {   "supplierData": { "supplierName": "Richard Roe",
                                "supplierId": "https://www.prodtomato.com/richard-roe/" },
                            "economicProfile": { "cf": 5000, "cv": 1.33, "additionalData": {} },
                            "productionProfile": { "q": 33, "tp": 365 } } ] },
                {   "resource": "Packaging materials", "bom": 0, "g": 0.1, "supplyList": [
                        {   "supplierData": { "supplierName": "Daniel Brown",
                                "supplierId": "https://www.packageservice.com/dbrown" },
                            "economicProfile": { "cf": 1000, "cv": 20, "additionalData": {} },
                            "productionProfile": { "q": 183, "tp": 365 } } ] } ] } ],
        "serviceLevel": {
            "financialServices": [ { "name": "Financial service",
                "uri": "https://www.somefinancialstuff.com",
                "providerId": "aRandomBank", "invested": 120, "ratio": 0.45 } ],
            "itServices": [ { "name": "IT service", "uri": "https://www.itservicescorp.com",
                "providerId": "itGuy", "access": "https://www.itservicescorp.com/services/it-service",
                "cost": 90 } ] } }
```

Listing 1. An excerpt of the JSON descriptor for the case at hand.

As a visual counterpart, Fig. 15 shows the page presented to participants to summarize the resulting configuration of the supply chain (request) data. The summary starts with the overall descriptors of a request and then presents a list of supplied resources for each level in the chain. Properties that pertain to the resource itself and the related supplies are shown for each resource type. The first level presents information about financial and IT services. With each set period, the algorithm applies the strategy for the selected sharing scheme to evaluate participants' quotas, given the reaped incomes in that lapse of time. The screenshot in Fig. 16 depicts the situation after computing the income quotas for the participants (in this case, two service providers: the financing actor and the product supplier).

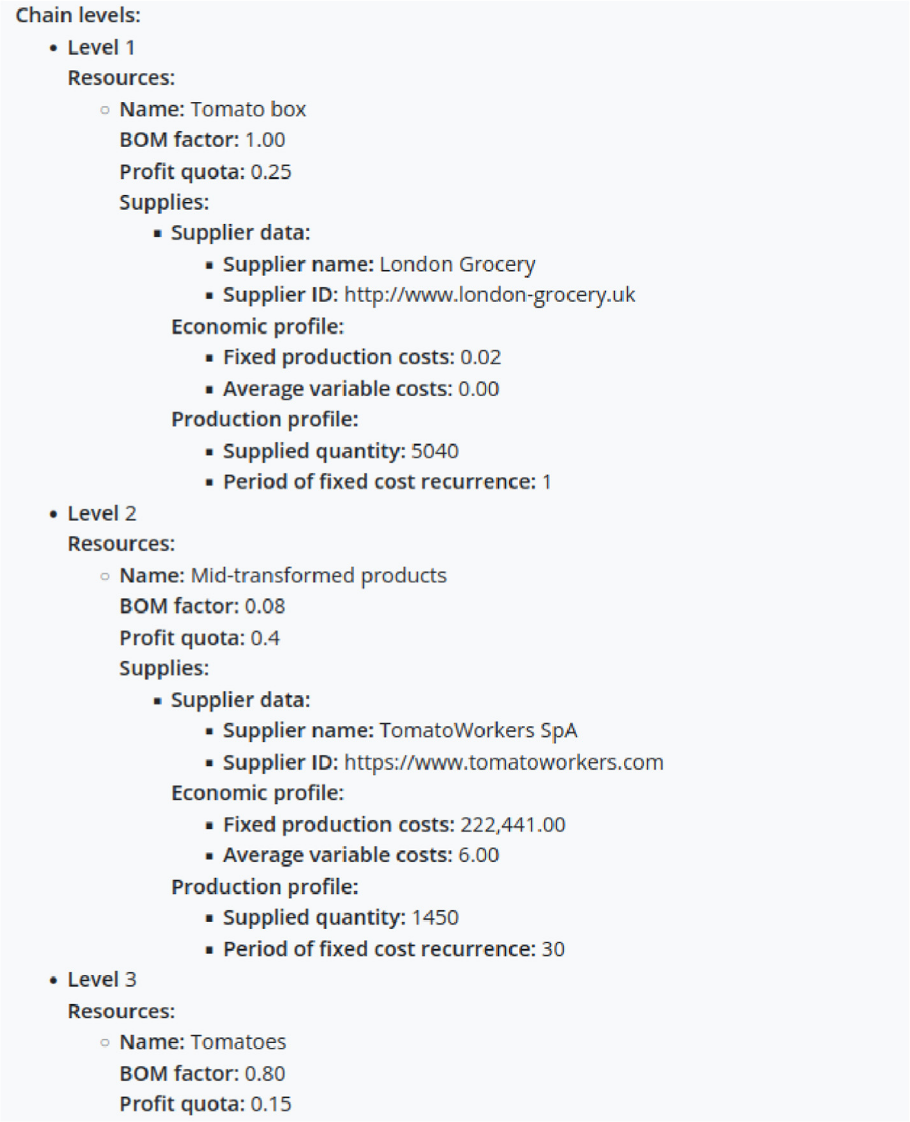
5. Discussion

We discuss here the contribution of the article both from the point of view of other blockchain-based methods and projects aimed at automating the processes underlying innovative business ecosystems and of the implementation options for the presented approach to Income Sharing.

5.1. Related work

From an organizational standpoint, Income Sharing can be seen as the evolution and transposition into the real economy of the DAO, a model widely known and discussed within the blockchain community, with a significant case history of implementation. The brainchild of Ethereum founder Vitalik Buterin[4], a DAO, is an entity that lives on the Internet and exists autonomously, relying on individuals to carry out tasks aimed at realizing a project or providing a service and on algorithms to coordinate them. In the spectrum of variations spanning RS and PS, this definition fits Income Sharing, and we could consider DAO and IS technological cases of convergent evolution. However, unlike DAO, IS is aimed at companies rooted in the real economy rather than subjects in the virtual world, with all the resulting concreteness, reflected in the procedural characteristics of supply chains, focused on bringing a product or service

[4] https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/.



Chain levels:

- **Level** 1

  Resources:

  - **Name:** Tomato box

    BOM factor: 1.00

    Profit quota: 0.25

    Supplies:

    - **Supplier data:**
      - **Supplier name:** London Grocery
      - **Supplier ID:** http://www.london-grocery.uk

      Economic profile:
      - **Fixed production costs:** 0.02
      - **Average variable costs:** 0.00

      Production profile:
      - **Supplied quantity:** 5040
      - **Period of fixed cost recurrence:** 1
- **Level** 2

  Resources:

  - **Name:** Mid-transformed products

    BOM factor: 0.08

    Profit quota: 0.4

    Supplies:

    - **Supplier data:**
      - **Supplier name:** TomatoWorkers SpA
      - **Supplier ID:** https://www.tomatoworkers.com

      Economic profile:
      - **Fixed production costs:** 222,441.00
      - **Average variable costs:** 6.00

      Production profile:
      - **Supplied quantity:** 1450
      - **Period of fixed cost recurrence:** 30
- **Level** 3

  Resources:

  - **Name:** Tomatoes

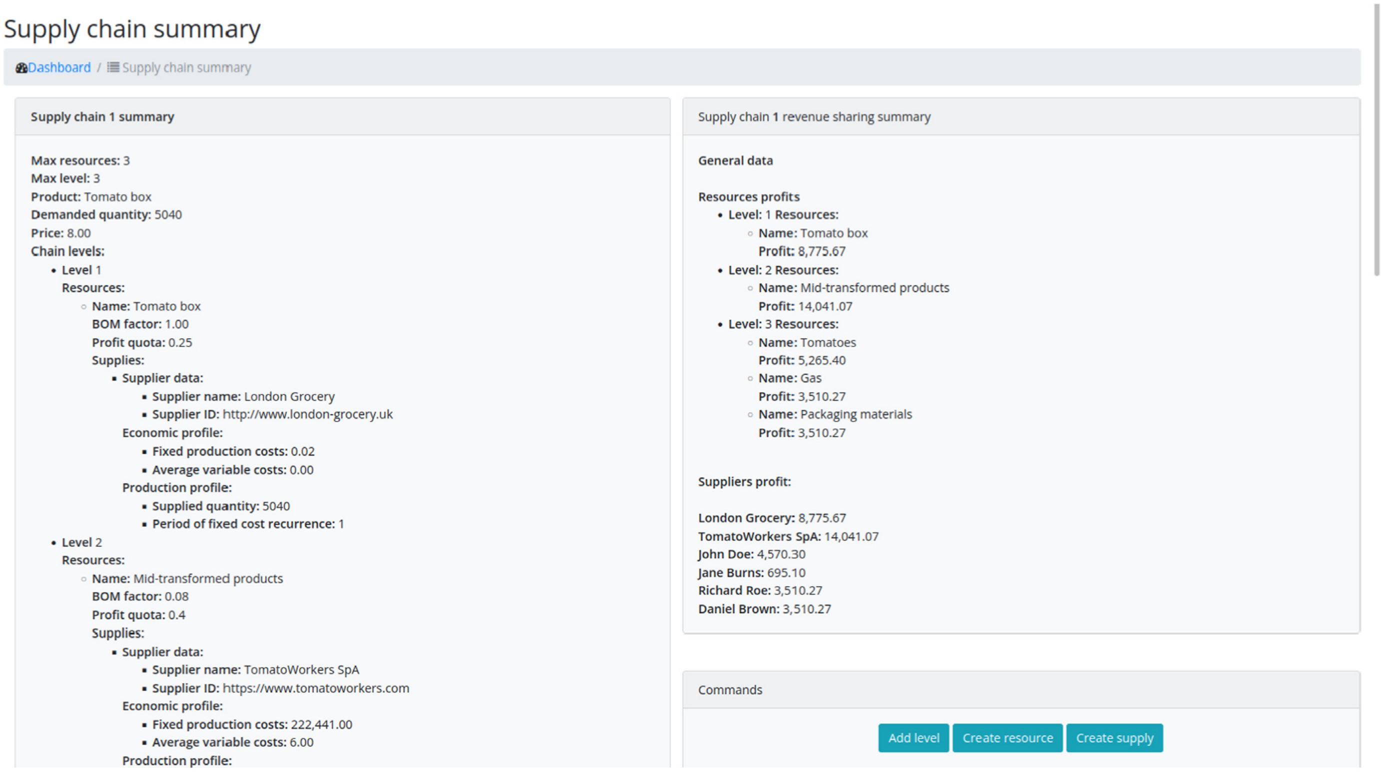
    BOM factor: 0.80

    Profit quota: 0.15

Fig. 15. A summary of a configuration for a supply chain.

to the market, with participants who, once on board, have all the reasons to collaborate closely to get results as quickly and profitably as possible. Contrast this with what happened with an early and most ambitious DAO project, eponymously dubbed "TheDAO" [21] and amounting to a purely virtual venture capital fund launched on the Ethereum blockchain in April 2016, only to be disabled within a few months due to opportunistic exploitation by an unknown "attacker" of a smart contract bug [22]. As argued in Ref. [23], lack of common goals, as well as missing collaboration and interaction between participants, largely explains the untimeliness in reacting effectively to the threat posed by the DAO exploit, which was well on its way to embezzle approximately $60 M. A legally questionable solution was found in extremis via a hard fork of the Ethereum blockchain. Predictably, this led to a temporary slowdown of DAO projects.

There is abundant literature, as well as a considerable number of ongoing projects, on the use of blockchain and DL technologies to support supply chains and collaborative processes [24–28], but relatively little effort has been devoted to their use for business innovation. Indeed, the bulk of these contributions was directed towards using blockchains and DLs to notarize the production steps along the supply chain to document the quality standards and compliance with current regulations. However, a few theoretical contributions align with our approach, which aims for radical business innovation. To begin with, Korpela et al. [29] took a Transaction Cost Economics (TCE) [30,31] point of view to provide, based on feedback from firms and business managers, an overview of the perspectives opened by blockchain technology for SCM. Treiblmaier [32] broadened this perspective to include Positive Agency Theory, Resource-based View Theory, and Network Theory. Bottoni et al. extensively discussed in Ref. [7] the correspondence between the methodological indications deriving from TCE and the blockchain-based IT architecture for the deployment of Revenue Sharing illustrated therein, which is expanded in this article to include Profit Sharing as well as Internet platforms such as Amazon and its likes to the extent that they contribute to realizing the general principle of Income Sharing. As TCE favors keeping core competencies in the firm and externalizing all else, the fundamental reasons for this correspondence are the ability to easily create global supply chains that promote the core competencies of companies and outsource all secondary ones thanks to digital trust. Value-adding alliances are thus enabled even between partners hitherto utterly unknown to each other, freeing companies from the need to do business only with their neighbors and paving the way to truly "glocal" excellence by combining the best, production-wise or service-wise, of a given territory with the best of another area placed at an arbitrary geographical and cultural distance. This road gets wider through the



Supply chain summary

Dashboard / Supply chain summary

Supply chain 1 summary

Max resources: 3
Max level: 3
Product: Tomato box
Demanded quantity: 5040
Price: 8.00
Chain levels:

- Level 1
  Resources:
  - **Name:** Tomato box
    BOM factor: 1.00
    Profit quota: 0.25
    Supplies:
    - **Supplier data:**
      - **Supplier name:** London Grocery
      - **Supplier ID:** http://www.london-grocery.uk

      Economic profile:
      - **Fixed production costs:** 0.02
      - **Average variable costs:** 0.00

      Production profile:
      - **Supplied quantity:** 5040
      - **Period of fixed cost recurrence:** 1
- Level 2
  Resources:
  - **Name:** Mid-transformed products
    BOM factor: 0.08
    Profit quota: 0.4
    Supplies:
    - **Supplier data:**
      - **Supplier name:** TomatoWorkers SpA
      - **Supplier ID:** https://www.tomatoworkers.com

      Economic profile:
      - **Fixed production costs:** 222,441.00
      - **Average variable costs:** 6.00

      Production profile:

Supply chain 1 revenue sharing summary

General data

Resources profits

- **Level:** 1 **Resources:**
  - **Name:** Tomato box
    Profit: 8,775.67
- **Level:** 2 **Resources:**
  - **Name:** Mid-transformed products
    Profit: 14,041.07
- **Level:** 3 **Resources:**
  - **Name:** Tomatoes
    Profit: 5,265.40
  - **Name:** Gas
    Profit: 3,510.27
  - **Name:** Packaging materials
    Profit: 3,510.27

Suppliers profit:

London Grocery: 8,775.67
TomatoWorkers SpA: 14,041.07
John Doe: 4,570.30
Jane Burns: 695.10
Richard Roe: 3,510.27
Daniel Brown: 3,510.27

Commands

Fig. 16. The interface showing the final sharing of proceeds.

architectural expansion illustrated here, which includes Internet platforms with their abilities to outsource and optimize logistics, marketing, and e-commerce. Therefore, our architecture guarantees trust in the interaction between companies and the ever more pervasive Internet platforms, thus letting the former take advantage of the latter without bending to their bargaining power. Lumineau et al. [33] provided a comprehensive discussion on the possibility of using blockchain technology for innovative forms of governance, which appear strongly congruent with Blockchain-as-a-Service and BaaP as contextualized here in the various Income Sharing options. They also point out the perspectives opened for governance automation, which fit with our framework for Income Sharing deployment. Morrison et al. [23] explored, from an Agency Theory point of view, the corporate governance implications of the DAO in its original formulation and identify possible vulnerabilities deriving from a simplistic application of algorithmic trust for its implementation. They address these weaknesses through proposals of hierarchization and focus on goals that fit our architecture.

Finally, several research works, e.g., Refs. [34–37], provide analytical models demonstrating that RS as a SCM methodology is optimally transferable to the technologies of blockchain and smart contracts. They can be seen as giving a formal background, albeit independently developed, to the implementations illustrated here and in Ref. [7].

5.2. Implementation trade-offs

The prototypes for Revenue Sharing have been implemented in Ethereum [7] and Hyperledger [8]. We have extended the latter to encompass the full-fledged Income Sharing architecture, including Revenue Sharing and Profit Sharing. In the implementation of Income Sharing, the preference for a platform for private blockchains such as Fabric over public blockchains such as Ethereum is indeed justifiable in terms of flexibility for the entrepreneurial entity that provides the technology in BaaS or BaaP mode, which is released in the provisioning of the service from the fluctuations of the underlying cryptocurrency, as these would directly affect the cost of transactions and thus prevent the economically reliable provision of services for the supply chain participants. Indeed, we believe that we are at a crossroads that will define a clear separation between two uses for business purposes of the blockchain [38]: a financial one, widely practiced in the context of cryptocurrencies and DeFi on public blockchains; another one of an industrial type that will materialize in a new kind of information systems, oriented to the inter-company context and business ecosystems of which supply chains are an instance, and which will more naturally pertain to private blockchains. The quantitative indications in this regard are in this direction, as shown in a study documenting the greater resilience of SCM projects based on Fabric versus those based on Ethereum [39]. We can expect their definitive consolidation once blockchains and DLs scale up from traceability to profitability in supply chains—a move our framework aims to contribute to.

Opting for private blockchain moderates the role of algorithmic trust enforced in public blockchains through very robust (albeit computationally and energetically inefficient) validation protocols such as Proof-of-Work [40]; yet, this aspect may be compensated by governance mechanisms where the various participants know each other. Privacy, a significant concern in industrial applications of public blockchains, can be flexibly managed at multiple levels, starting with the permissioned nature of participation [15]. Nevertheless, the exchanged information accessible to the whole consortium participating in the supply chain could reveal strategic details of processes and decisions to other parties. To overcome this issue, techniques that allow data owners to determine who can access exchanged information selectively have been proposed [41–44]. Their integration with our system and the implementation in the context of Income Sharing is an exciting challenge for future endeavors.

Computational efficiency is increased by the smaller number of nodes in private blockchains and the computationally less demanding consensus protocols that can be applied assuming a higher level of human trust (e.g., the Crash-Fault Tolerant protocols adopted in Fabric[5]). In any case, the trust management systems guarantee, despite having a more limited role than in public blockchains, a level of automation and reliability that warrants, in putting business partners together, doing away with the lengthy and expensive preliminary checks typical of the pre-digital economy.

Furthermore, the fact that governance is not entirely alienated in favor of an algorithmic setup facilitates corrective intervention in the face of emergencies and anomalies, an aspect that has been decisively lacking in the case of the DAO exploit, as pointed out by Morrison et al. [23]. This aspect calls for studies on adopting and tuning mechanisms capable of freezing, disabling, or compensating operations or entire smart contracts involved in the supply chain in response to dangerous events. This feature, combined with the focused nature and goal-orientation of the Income Sharing consortia, appears promising for their robust and effective functioning and paves the path for future work. Additionally, Fabric lends itself to the programmability of robust systems for the prevention and mitigation of cyber-attacks and other malicious actions based on the real-time analysis of logs through artificial intelligence technologies, as shown in Ref. [45], as well as of smart contract vulnerabilities, as shown in Ref. [46], to avoid situations of generalized “panic mode”, as occurred in the context of the Ethereum DAO exploit, even in the worst case scenario.

At the same time, the worlds of public and private blockchains, if they end up technologically distancing themselves, will still be able to communicate as needed: think of the case of a tender with a public administration in which the successful bidder is the originator of a chain coordinated through Income Sharing, which in turn must publicly announce milestones and achievements of the contractual execution. In this case, while the distribution of income can be managed internally to the supply chain through a private blockchain, the announcements required by contractual/regulatory obligation must be made publicly. Thus, private and public blockchains must be integrated, and the Hyperledger ecosystem provides the Besu and Burrow frameworks that can be directly integrated with Ethereum [15]. Fabric itself is amenable, albeit not so automatically, but without restrictions concerning the public blockchain to be combined with (as illustrated in Ref. [4] regarding integration between Fabric and the Stellar public blockchain).

Other relatively minor points concern some specific choices for software developments that differentiate Fabric and Ethereum. Ethereum provides its contract-oriented programming language, Solidity, executable on the Ethereum Virtual Machine (EVM). This setup works effectively for developing DeFi applications but does not appear to be well-suited for the large-scale developments required by industrial environments. By contrast, Fabric supports a large variety of widely used languages, including Java and JS, in addition to the more recent Go. It can be deployed through much more widespread deployment setups such as NPM, Node, Go, and Docker, all widely proven for industrial applications.

A further limiting factor is that Solidity's relatively small memory stack is dedicated to declarable variables. For IS algorithms that involve a large number of parameters, this makes it necessary to distribute the arguments over several functions that need to call each other, with consequent unnecessary complications in the structure of the code. Yet another problematic aspect is that numeric variables in decimal format are poorly supported, which is no help for calculating income shares, where operations with decimal numbers are the order of the day.

6. Conclusions and future work

We presented an architecture for virtual consortium organizations, but with members corresponding to companies rooted in the real economy, based on the general principle of Income Sharing among participants in a supply chain. We have illustrated how it can be implemented on a blockchain or DL infrastructure to guarantee its optimality and reliability through smart contracts. We have shown how this architecture makes available a comprehensive menu of optimized executions of supply chains attributable to different Income Sharing options in a spectrum of choices ranging from Revenue Sharing to Profit Sharing. We have included in the scenario the role played by service providers such as Amazon and other Amazon-like Internet platforms, whose part is ever more relevant in the global economy. To this end, we formally modeled the role of the blockchain platform itself as an entrepreneurial entity in the supply chain, in the alternative roles of BaaS and BaaP, depending on which Income Sharing veers towards Revenue Sharing or Profit Sharing. We have also indicated the relationship between this architecture and the well-known DAO, whose primary vision it strengthens regarding economic concreteness and computational feasibility. This architecture was implemented in Fabric, the most successful platform for private and consortium-distributed ledgers, developed within the Hyperledger ecosystem. We have indicated that this choice is the most appropriate for the entrepreneurial role played by the blockchain platform in this context. It also brings advantages in privacy management and implementation efficiency.

While we have tested both the algorithms and the architecture implemented on real data from an agri-food supply chain obtained in the context of the LEMURE project funded by the Italian Ministry of University and Research (MUR), we have not yet actually deployed a prototype tested in full operation. This is the future goal of the ongoing research pursued through the WEBEST project, also funded by MUR. In this context, efficiency tests of the Income Sharing implementation will also be performed according to various parameters, such as the size and depth of the supply chain. Also, the integration of mechanisms to recover from, or freeze in, the status of compromised executions and attacks constitutes an interesting avenue for future research. Finally, we aim to design and implement techniques that let data to be accessible by the sole partners that are interested in the exchanged information to avoid leakage of strategic knowledge from other parties.

Author contribution

Paolo Bottoni: Methodology, Software, Writing—Original draft, Writing—Review & Editing. **Claudio Di Ciccio:** Methodology, Formal analysis, Writing—Original Draft, Writing—Review & Editing, Visualization. **Remo Pareschi:** Conceptualization, Methodology, Writing—Original Draft, Writing—Review & Editing. **Domenico Tortola:** Software, Data Curation, Validation. **Nicola Gessa:** Conceptualization, Methodology, Validation, Writing—Original draft, Writing—Review & Editing. **Gilda Massa:** Conceptualization, Methodology, Validation, Writing—Original draft, Writing—Review & Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Work performed under the MoU between ENEA, the Department of Computer Science at Sapienza, and the University of Molise. C. Di Ciccio and P. Bottoni were partly supported by the Italian Ministry of University and Research (MUR) under PRIN grant B87G22000450001 (PINPOINT). C. Di Ciccio also received funding from the MUR grant “Dipartimenti di eccellenza 2018–2022” (for the Department of Computer Science at Sapienza), the Sapienza SPECTRA research project, and the Cyber 4.0 project BRIE. We thank the anonymous reviewers for their precious suggestions.

[5] https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf.

References

[1] G.P. Cachon, M.A. Lariviere, Supply chain coordination with revenue-sharing contracts: strengths and limitations, Manag. Sci. 51 (1) (2005) 30–44, https://doi.org/10.1287/mnsc.1040.0215.

[2] D. Gong, S. Liu, M. Tang, L. Ren, J. Liu, X. Liu, Revenue sharing or profit sharing? An internet production perspective, Adv. prod. eng. manag. 13 (1) (2018) 81–92, https://doi.org/10.14743/apem2018.1.275.

[3] J. Song, P. Zhang, M. Alkubati, Y. Bao, G. Yu, Research advances on blockchain-as-a-service: architectures, applications and challenges, Digital Commun. Netw. 8 (4) (2022) 466–475, https://doi.org/10.1016/j.dcan.2021.02.001.

[4] F. Carlini, R. Carlini, S.D. Palma, R. Pareschi, F. Zappone, The Genesy model for a blockchain-based fair ecosystem of genomic data, Frontiers Blockchain 3 (2020) 483227, https://doi.org/10.3389/fbloc.2020.483227.

[5] S. Franzoni, Blockchain and Smart Contracts in the Fashion Industry. Master's Thesis, Politecnico di Torino, Torino, Italy, 2020.

[6] L. Zhang, L. Hang, W. Jin, D. Kim, Interoperable multi-blockchain platform based on integrated rest apis for reliable tourism management, Electronics 10 (23) (2021), 2990, https://doi.org/10.3390/electronics10232990.

[7] P. Bottoni, N. Gessa, G. Massa, R. Pareschi, H. Selim, E. Arcuri, Intelligent smart contracts for innovative supply chain management, Frontiers Blockchain 3 (2020) 535787, https://doi.org/10.3389/fbloc.2020.535787.

[8] P. Bottoni, R. Pareschi, D. Tortola, N. Gessa, G. Massa, Distributed ledgers to support revenue-sharing business consortia: a hyperledger-based implementation, in: Proceedings of IEEE Symposium on Computers and Communications, IEEE, 2021, pp. 1–6, https://doi.org/10.1109/ISCC53001.2021.9631473.

[9] J. Dana Jr., K. Spier, Revenue sharing and vertical control in the video rental industry, J. Ind. Econ. 49 (3) (2001) 223–245, https://doi.org/10.1111/1467-6451.00147.

[10] Y. Wang, L. Jiang, Z.J. Shen, Channel performance under consignment contract with revenue sharing, Manag. Sci. 50 (2004) 34–47, https://doi.org/10.1287/mnsc.1030.0168.

[11] G. Qian, Y. Zhang, J. Wu, Y. Pan, Revenue sharing in dairy industry supply chain - a case study of hohhot, China, J. Integr. Agric. 12 (2013) 2300–2309, https://doi.org/10.1016/S2095-3119(13)60585-7.

[12] I. Giannoccaro, P. Pontrandolfo, Supply chain coordination by revenue sharing contracts, Int. J. Prod. Econ. 89 (2004) 131–139, https://doi.org/10.1016/S0925-5273(03)00047-1.

[13] R. Tononi, G. Massa, R. Raimondi, G. Spagna, La Supply Chain Integrata, vol. 6, Energia Ambiente e Innovazione, 2007.

[14] E. Çanakoğlu, T. Bilgiç, Analysis of a two-stage telecommunication supply chain with technology dependent demand, Eur. J. Oper. Res. 177 (2) (2007) 995–1012, https://doi.org/10.1016/j.ejor.2006.01.006.

[15] Y. Wei, T.M. Choi, Mean–variance analysis of supply chains under wholesale pricing and profit sharing schemes, Eur. J. Oper. Res. 204 (2010) 255–262, https://doi.org/10.1016/j.ejor.2009.10.016.

[16] I. Allison, Ibm's Food Blockchain is Going Live with a Supermarket Giant on Board. https://www.coindesk.com/ibm-food-supply-chain-blockchain-carrefour-live-production. (Accessed 25 July 2022).

[17] R. Aitken, Ibm forges global joint venture with maersk applying blockchain to 'digitize' global trade. https://www.forbes.com/sites/rogeraitken/2018/01/16/ibm-forges-global-joint-venture-with-maersk-applying-blockchain-to-digitize-global-trade/#b232864547e8, 2018. (Accessed 25 July 2022).

[18] M. Dumas, M. La Rosa, J. Mendling, H.A. Reijers, Fundamentals of Business Process Management, second ed., Springer, Berlin, Heidelberg, 2013.

[19] R. Tononi, G. Amorosi, Networks of SMEs as virtual web organizations: an experimental program aimed at supporting SMEs in depressed areas of Italy, in: U.J. Franke (Eds.), Managing Virtual Web Organizations in the 21st Century: Issues and Challenges, IGI Global, Hershey, PA, 2002, pp. 198–212.

[20] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Boston, MA, 1994.

[21] S. Bannon, The Tao of "The DAO" or: how the autonomous corporation is already here. https://techcrunch.com/2016/05/16/the-tao-of-the-dao-or-how-the-autonomous-corporation-is-already-here/ 2016. (Accessed: 25 July 2022)..

[22] V. Dhillon, D. Metcalf, M. Hooper, et al., The DAO hacked, in: Blockchain, Enabled Applications, Apress, Berkeley, CA, 2017, pp. 67–78, https://doi.org/10.1007/978-1-4842-3081-7_6.

[23] R. Morrison, N.C.H.L. Mazey, S.C. Wingreen, The DAO controversy: the case for a new species of corporate governance? Frontiers Blockchain 3 (2020) 25, https://doi.org/10.3389/fbloc.2020.00025.

[24] C.D. Ciccio, G. Meroni, P. Plebani, On the adoption of blockchain for business process monitoring, Software Syst. Model 21 (2022) 915–937, https://doi.org/10.1007/s10270-021-00959-x.

[25] F. Corradini, A. Marcelletti, A. Morichetta, A. Polini, B. Re, F. Tiezzi, Engineering trustable and auditable choreography-based systems using blockchain, ACM Trans. Manag. Inf. Syst. 13 (31) (2022) 1–53, https://doi.org/10.1145/3505225.

[26] C. Di Ciccio, A. Cecconi, M. Dumas, L. García-Bañuelos, O. López-Pintado, Q. Lu, J. Mendling, A. Ponomarev, A.B. Tran, I. Weber, Blockchain support for collaborative business processes, Inform. Spektrum 42 (2019) 182–190, https://doi.org/10.1007/s00287-019-01178-x.

[27] M.F. Madsen, M. Gaub, T. Høgnason, M.E. Kirkbro, T. Slaats, S. Debois, Collaboration Among Adversaries: Distributed Workflow Execution on a Blockchain, FAB, 2018, pp. 8–15.

[28] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, J. Mendling, Untrusted business process monitoring and execution using blockchain, in: M. La Rosa, P. Loos, O. Pastor (Eds), Business Process Management https://doi.org/, Springer, Cham, 2016, pp. 329–347, https://doi.org/10.1007/978-3-319-45348-4_19.

[29] K. Korpela, J. Hallikas, T. Dahlberg, Digital supply chain transformation toward blockchain integration, in: T. Bui (Ed.), Proceedings of the 50th Hawaii International Conference on System Sciences, HICSS, 2017, pp. 1–10.

[30] R.H. Coase, The nature of the firm, Economica 4 (4) (1937) 386–405, https://doi.org/10.2307/2626876.

[31] O. Williamson, The Economic Institutions of Capitalism: Firms, Markets, Relational Contracting, Free Press, 1985.

[32] H. Treiblmaier, The impact of the blockchain on the supply chain: a theory-based research framework and a call for action, Supply Chain Manag.: Int. J. 22 (6) (2018) 545–559, https://doi.org/10.1108/SCM-01-2018-0029.

[33] F. Lumineau, W. Wang, O. Schilke, Blockchain governance—a new way of organizing collaborations? Organ. Sci. 32 (2) (2020) 500–521, https://doi.org/10.1287/orsc.2020.1379.

[34] T. Choi, Supply chain financing using blockchain: impacts on supply chains selling fashionable products, Ann. Oper. Res. (2020) 1–23, https://doi.org/10.1007/s10479-020-03615-7.

[35] S. Hayrutdinov, M. Saeed, R. Azamat, Coordination of supply chain under blockchain system-based product lifecycle information sharing effort, J. Adv. Transport. 2020 (2020) 5635404, https://doi.org/10.1155/2020/5635404.

[36] K. Korsten, A Blockchain-Based Supply Chain Contract - an Analysis to the Development of a Supply Chain Contract on a Blockchain. Master's Thesis, Eindhoven University of Technology Eindhoven, the Netherlands, 2019.

[37] L. Liu, F. Li, E. Qi, Research on risk avoidance and coordination of supply chain subject based on blockchain technology, Sustainability 11 (7) (2019) 2182, https://doi.org/10.3390/su11072182.

[38] C. Mohan, State of public and private blockchains: myths and reality, in: Proceedings of the 2019 International Conference on Management of Data, ACM, 2019, pp. 404–411, https://doi.org/10.1145/3299869.3314116.

[39] N. Vadgama, P. Tasca, An analysis of blockchain adoption in supply chains between 2010 and 2020, Frontiers Blockchain 4 (2021) 610476, https://doi.org/10.3389/fbloc.2021.610476.

[40] M. Garriga, S.D. Palma, M. Arias, A.D. Renzis, R. Pareschi, D.A. Tamburri, Blockchain and cryptocurrencies: a classification and comparison of architecture drivers, Concurrency Comput. Pract. Ex. 33 (8) (2021) e5992, https://doi.org/10.1002/cpe.5992.

[41] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: the blockchain model of cryptography and privacy-preserving smart contracts, in: Proceedings of the 2016 IEEE Symposium on Security and Privacy, (SP), IEEE, 2016, pp. 839–858, https://doi.org/10.1109/SP.2016.55.

[42] B. Li, Y. Wang, RZKPB: a privacy-preserving blockchain-based fair transaction method for sharing economy, in: Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, 2018, pp. 1164–1169, https://doi.org/10.1109/TrustCom/BigDataSE.2018.00161.

[43] B. Li, Y. Wang, P. Shi, H. Chen, L. Cheng, FPPB: a fast and privacy-preserving method based on the permissioned blockchain for fair transactions in sharing economy, in: Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, 2018, pp. 1368–1373, https://doi.org/10.1109/TrustCom/BigDataSE.2018.00189.

[44] E. Marangone, C. Di Ciccio, I. Weber, Fine-grained data access control for collaborative process execution on blockchain, in: A. Marrella, R. Matulevicius, R. Gabreyelczyk, et al., (Eds.), Business Process Management: Blockchain, Robotic Process Automation, and Central and Eastern Europe Forum, Springer, Cham, 2022, pp. 51–67, https://doi.org/10.1007/978-3-031-16168-1_4.

[45] P.D. Pilla, R. Pareschi, F. Salzano, F. Zappone, Delta - distributed elastic log text analyser, June 20, 2022, in: M. Pizzonia, A. Vitaletti (Eds.), Proceedings of the 4th Workshop on Distributed Ledger Technology Co-located with the Italian Conference on Cybersecurity 2022 (ITASEC 2022), CEUR-WS.org, Rome, Italy, 2022, pp. 48–59, http://ceur-ws.org/Vol-3166/paper04.pdf.

[46] P.D. Pilla, R. Pareschi, F. Salzano, F. Zappone, Hyperledger fabric attacks mitigation (extended abstact), in: FOCODILE 2022 - 3rd International Workshop on Foundations of Consensus and Distributed Ledgers, 2022, 00–01.