

## APPLIED RESEARCH

# A Blockchain-Based Decentralized Marketplace for Trustworthy Trade in Developing Countries

MANUEL PEREIRA LAMELA<sup>1</sup>, JESÚS RODRÍGUEZ-MOLINA<sup>1</sup>,  
MARGARITA MARTÍNEZ-NÚÑEZ<sup>2</sup>, AND JUAN GARBAJOSA<sup>3</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Telematics and Electronics Engineering, Universidad Politécnica de Madrid, 28040 Madrid, Spain

<sup>2</sup>Department of Organization Engineering, Business Administration and Statistics, Universidad Politécnica de Madrid, 28040 Madrid, Spain

<sup>3</sup>Higher Technical School of Engineering of Information Systems, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Jesús Rodríguez-Molina (jesus.rodriiguez@upm.es)

This work was supported by in part by the Agencia Estatal de Investigación/Ministerio de Ciencia e Innovación through the project Sustainability-Aware IoT Systems Driven by Social Communities (SIoTCom) under Grant PID2020-118969RB-I00.

**ABSTRACT** The possibilities that Distributed Ledger Technologies (DLTs) offer for cooperation, development, and achievement of the Sustainable Development Goals (SDGs) are remarkable. This is because DLTs enable several key features, such as sharing complete information about every data transaction in the distributed system that participants belong to, the immutability of the recorded data transactions, or consensus among what data can be regarded as true, of great usefulness for the implementation of the SDGs. As far as developing countries are concerned, this information could be useful in trading locally produced goods, as it could enhance the reputation and profitability of Small-Scale Producers (SSPs). Unfortunately, it is rare to find a digitalized marketplace that has been specifically implemented for this application domain. This paper puts forward a blockchain-based marketplace that makes use of Smart Contracts and offers information about how the sold goods were produced and can be traced to their very origin. Besides, cloud computing has been conceived to be used in this development from the beginning to reduce the computational resources required by end user operations. An implementation with cloud computing facilities, software components running on the Ethereum blockchain and a web front end have been tested with satisfactory performance results.


**INDEX TERMS** Distributed computing, sustainable development, peer-to-peer computing.

## I. INTRODUCTION

Ever-increasing awareness regarding the importance of Sustainable Development Goals (SDGs) for the future development of the global economy is changing many different areas of the supply chain and how goods and services can be traded [1], [2]. SDGs are permeating not only most significant aspects of research and development, but also many other activities related to how trade is carried out and how products are made available to consumers. However, among the many possible domains, developing countries are the ones where significant work has still to be carried out. Indeed, lack of production visibility is one issue that Small Scale Producers (SSPs) suffer with intensity in developing countries, as it is difficult for them to get their products advertised for a

larger audience that could be interested in their purchase. In addition to that, SDGs works in developing countries related to markets and supply chains [3], though present, are often scarce and hard to find. This poses a significant issue related to trade, as it locks developing countries in a circle that cannot be broken easily: goods and services to sell are offered to a very limited range of clients, which thwarts increasing outputs production, which in turn ends up offering little incentives towards greater profitability through market access. Additionally, this situation prevents further development of the goals set by the SDGs in terms of productivity, equality, and progress.

This manuscript describes how to build a blockchain-based, SDG-driven marketplace oriented towards developing countries where information related to development, growth and monitoring of primary sector products has been recorded. There are several features that make this marketplace unique:

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro .

a) it makes use of Smart Contracts to record transactions regarding foodstuffs produced in developing countries as a way to guarantee transparency in trade and price formation, b) it is capable of registering information about how products have been grown or treated during their lifecycle and c) it offers two technological solutions to supersede the challenges in terms of data storage and computational power required by a blockchain: on the one hand, the Proof-of-Authority (PoA) algorithm has been used rather than Proof-of-Stake (PoS) or Proof-of-Work (PoW) ones to minimize the resources that would be required to validate the newly added blocks. On the other hand, an inexpensive cloud computing infrastructure has been utilized to store all the blocks and transactions-related data. The usage of PoA and cloud computing is of great usefulness for developing countries, since costs in access and hardware ownership can be minimized, if there is a reliable Internet connection for trade purposes. In this way, it can be ensured that the blockchain where transactions are taking place is kept with no constraints regarding data or computational resources.

#### A. MARKET PLATFORMS TO ENHANCE SSP TRADE AND SDGS

We refer to SSPs as units of small associations of persons focused on producing foodstuffs, developing goods, or offering services in a comparatively small size. According to [4], “*small-scale farmers produce crops primarily for subsistence and only the marketable surplus is sold. In other words, farmers’ market participation is directly related to marketable surplus generation*”. While they are not the only production unit for primary sector outputs, SSPs are widespread in developing countries, since they are local producers and farms that do not rely on just one massively produced crop oriented towards exportation. It is mentioned in [5] how there are different kinds of SSP exploitations (subsistence, transition, pre-commercial, commercial) depending on their level of development, and to what extent they are oriented either to pure subsistence or towards profitability through surplus sale. Unfortunately, there are several issues that prevent SSPs from reaching their full potential, ranging from technical (security, cost of digital tools, mobile networking) to economical ones (digital literacy, safety, information policies). Among those, one is the lack of visibility and access to large markets for SSPs. This issue has a major impact in the profitability of SSP farms, due to the severe limitations imposed by how many customers are aware of products that might be interested in purchasing. The role of market platforms as trade catalyzers is of major importance for economic development, as they enable producers to sell their outputs and obtain a profit from them. In the case of developing countries, SSPs may use both local and online market platforms to sell the outputs that their farms can produce. This is of major importance because when combined with other problems (lack of technification, strained access to financial products, transport, delivery inefficiencies), lack of visibility ends up keeping productivity lower than it could be due to the small size of the markets that can effectively be accessed. Therefore, creating a market

platform that offers a higher degree of awareness for SSPs is a desirable objective. Although there are different alternatives to creating such a market, an online market platform can be deemed as one of the most suitable options, as it has a relatively small cost of development and is greatly effective when creating awareness about SSPs’ outputs being sold to a wider audience.

Furthermore, online market platforms can be used to pave the way to the accomplishment of several SDGs. The latter can be defined as seventeen objectives aimed at improving the standards of life of all humans on a worldwide scale, tackling issues as inequality, poverty, economic development, or governance [6]. Among them, SDG 9 is the one that can benefit the most from the system that is put forward in this case (Industry, Innovation and Infrastructure, [7]). Due to the advantages that an online market platform can offer for trade and visibility, and the features that the one that is presented in this manuscript is putting forward, the described solution can be regarded as a distributed infrastructure of innovative characteristics applied to agroindustry. In addition to SDG 9, the distributed, secure trade platform oriented to developing countries that is described in this manuscript also provides support to SDGs number 8 (Decent Work and Economic Growth), 10 (Reduced Inequality) and 12 (Responsible Consumption and Production). This is so because the platform is oriented towards economic enhancement of the SSPs that offer their outputs and, since it would make possible knowing how crops were grown or cattle was fattened, it can promote transparency in production and consumption of the displayed foodstuffs.

#### B. BLOCKCHAIN AND CONTRIBUTIONS TO SUSTAINABLE DEVELOPMENT GOALS IN DEVELOPING COUNTRIES

Aside from the advantages that are offered by the online market platforms regarding trade for SSPs, there are other aspects related to the technology used in our proposal that must be considered, especially the ones related to Digital Ledger Technologies (DLTs). Among the latter, blockchain must be highlighted as one of the most compelling technologies that a trade platform has to offer. Blockchain can be defined as a distributed ledger where information about any transaction that takes place within the system where it is deployed is recorded and shared among all the participant nodes. Some features of blockchain can be described as “*high security, decentralization, and tamper-proof, which enables all participants to realize value interconnection and transmission at a very low cost*” [8]. Indeed, there are several characteristics that, from our point of view, make blockchain unique: *distribution* (nodes of a blockchain are located in different places connected to each other via network), *redundancy* (the same information in every node is virtually present in every location of the blockchain), *transparency* (information about blockchain transactions and involved identifiers is shared and available for every participant in the system), *immutability* (transactions and data present in the block-chain cannot be modified), and *consensus* (information considered as valid is decided by, at least, the users with the majority of the

computational power in the blockchain). Due to the kind of information that is used in the platform conceived by the authors of this manuscript, the usage of blockchain comes as extremely handy for the purposes shown here. In addition to that, blockchain makes possible the deployment of Smart Contracts to integrate both new products and customers into the market and keep a track record of their activities to check whether they can be trusted or not. To secure the identities of the actual people and companies that are participating, a simple mechanism can be created where new participants are assigned an identifier based on the public key from the asymmetric cryptography key pair that they use to validate transactions. This public key will be not only used as an identifier, but also as the wallet address within the system used to send and receive funds related to trade activities.

Lastly, blockchain can be extremely useful for any application related to sustainability, as it can provide a high level of traceability on the resources that have been used for production, delivery or consumption of goods and services, such as wine supply chain [9] or energy [10].

### C. CONTRIBUTIONS OF THE PAPER

The contributions done in this paper are linked to the design and implementation details of a blockchain-based marketplace oriented towards production and trade of goods and services produced in developing countries. It has been put forward in this proposal that blockchain will be useful for two purposes: on the one hand, it can be used for the traceability of the products that are sold in it, so their origin, delivery and elaboration procedures become transparent. This feature will result in benefits for the product sellers (who can prove that their products have been grown or matured in an environment that enhances their characteristics) and buyers (who become aware of the goods production conditions in the whole supply chain). On the other hand, blockchain also offers the possibility of knowing how final prices for the sold products are composed, and what party is acquiring each share of the costs and benefits. Finally, cloud infrastructure is used to minimize the computational capabilities that end user devices must have.

While some of these contributions are already used, it is the authors' opinion that it is unseen how all these different features are combined onto a platform with the purpose, or the aim (developing countries) shown in this manuscript. A system like the one described in this paper can provide significant usability: for example, it might happen that someone wants to buy a product that they cannot obtain locally on their own, so they decide to get it from an online marketplace. However, their purchase might be riskier than expected because there is no information on the online store about when the raw materials of the product were collected, transported or transformed, and the producers are so small in size that are virtually unknown to the end user, who in the end might be unwilling to pay for the products and not make the deal due to the fact that they do not trust neither the kind of product they are buying nor the producer they are

buying it from (due to their lack of information about the manufacturing process of the product and/or the reputation and reliability of the seller).

This example shows the motivation behind the system that is put forward in this manuscript. By deploying a blockchain where a) information about how a product is elaborated in every stage (from its raw state or raw materials to what is sold to end customers) is traceable, b) reputation of the seller/producer in terms of product and trade (the quality of the product, issues with purchases or payments, etc.) is transparent and clear and c) Smart Contracts are deployed to get instantaneous engagement in terms of offer, demand and payment so transactions can be executed in a much faster and reliable way, even and precisely among parties that have never done business before, the usefulness of the proposed system can be proven. While some of these features are already existing outside blockchain developments, the usage of the latter makes possible a more decentralized, harder to manipulate storage of seller information, which prevents its tampering or spurious usage by a party prominent enough to centralize information about trade operations. Furthermore, traceability of the whole number of procedures used to manufacture the product sold increases trust in buyers, who might be more inclined to buy something if they are fully aware of how it is made, even if that means paying a higher price.

### D. PAPER STRUCTURE

This paper is organized as follows: a first introductory section with the main ideas of the manuscript (such as the background that it is encased within), as well as the contributions and benefits that the proposed marketplace offers has already been presented. Section 2 involves the State of the Art of the existing solutions that intend to tackle, to a greater or a lesser extent, the same problem from a different perspective or technology, along with the advantages, disadvantages and open issues that have been found during this study. Next section describes the analysis and design features related to the solution conceived for the decentralized marketplace, as well as the security analysis done on the solution. Section 4 shows the implementation and tests carried out to test the performance of the solution in terms of time required to satisfy a request and how they escalate during a certain amount of time. How information can be obtained from the good that is being traded from a client perspective or the sales record of another fellow vendor from the selling side has been considered as well. Section 5 contains the conclusions that can be extracted from this development and the future works that can be attempted to go further what is put forward in this solution. Finally, bibliographical references used to compose this manuscript are included.

## II. STATE OF THE ART

The main purpose of this section has been finding out about the available solutions, both in terms of research and actual deployments, that make use of technology resembling the one put forward here by us in a similar application domain.

### A. STUDY OF THE STATE OF THE ART

It is described in [11] how Fog Computing can provide infrastructure for Industrial Internet of Things (IIoT) marketplaces, so it becomes an intermediary technology between field components (edge devices, Apps in a smartphone) and cloud ones (developer store, blockchain). The objective is establishing what the authors refer to as an *IIoT Bazaar*, describing the main components that a marketplace for industrial edge applications would have. A private Ethereum blockchain is used for transparency, as well as to enhance the traceability of application installations in edge devices. Their proposal, though, lies on a scope different from the one that the authors of this manuscript are putting forward: our development creates a marketplace for goods and services produced in developing countries to support transparent trade oriented towards agricultural production, whereas [11] main drive is providing a place for trade on industrial edge applications of hardware and software nature. Additionally, [12] shows how a Blockchain-Based Decentralized Digital-Content Marketplace can be built. The authors of the reviewed paper create a service that can offer more security and user experience than other peer-to-peer systems, without having the disadvantages of a centralized one. They use a blockchain system called LBRY [13], one synchronized namespace and a global index for content related to metadata. Blockchain is claimed to be used to have a decentralized content platform to carry out several actions in an easier way. Several of the technologies used share significant similarities with our proposal, but the marketplace described is oriented towards digital content, whereas the one put forward by us has been conceived for physical goods produced in developing countries.

João Martins *et al.* [14] show how Customer Bargaining and E-Procurement can be provided via a decentralized market. In this context, blockchain can both allow the interconnectivity and transparency in Supply Chains and enable payments in a faster manner. The authors of this manuscript have created a Smart Contract architecture where listing interaction relies on contracts that make use of tracking tokens, along with procedures for listing, aggregation, and auctioning. Overall, Smart Contracts are regarded as of critical importance in this kind of marketplaces, as they are needed to establish in an accurate manner how payments will be done among parties. The pricing mechanism followed consists of reverse auction bids, which does not apply to the context of foodstuffs and agricultural production from developing countries. Furthermore, [15] describes the foundations on using an IoT-Blockchain Enabled Optimized Provenance System for Food Industry that also makes use of Artificial Intelligence. The authors of the reviewed manuscript describe a methodology that combines AI and blockchain to create a system ensuring that the provenance of marketed food is legitimate. The authors have built a system with four actors (farmer, warehouse, retailer, and consumer) traced via blockchain facilities. While the scope of our manuscript is different, product traceability will indeed benefit from

blockchain usage. The work done by Park *et al.* [16] shows how DLTs can also be used to develop an IoT data marketplace where reputation of the data owner is enabled via blockchain. A system with four different layers has been conceived for all the stages mandatory for the system. The authors of the manuscript put forward a remarkable proposal for the data that takes into consideration how they can be traded in a more transparent way, while taking into account the reputation of the data owner. However, this proposal is neither oriented towards goods produced in developing countries, nor it considers how physical products can be traced during their growth.

Another example of marketplace on the Ethereum blockchain is shown in [17]. The main issues about a centralized market application (lack of privacy users' data, transaction fees, unilateral blocking of merchants) are pointed out. The authors of the reviewed manuscript have implemented a model where a Smart Contract written in solidity is included in the Ethereum Network, accessed via a web browser by means of the Web3.js Application Programming Interface (API). Fund quantity and availability is kept in a Metamask wallet [18]. The Inter Planetary File System (IPFS) is used in a compelling manner, as it allows fully decentralized data storage [19]. Performance is at 3.8 seconds. However, there is not a mention on how to use such an infrastructure in developing countries, what kind of actions can be carried out to make the required blockchain capabilities easier to port or how to trace the goods sold. Guido Perboli *et al.* describe in [20] how a decentralized marketplace for Machine-to-Machine (M2M) economy can be created in Smart Cities. They put forward a decentralized marketplace called PEGASUS where a Smart City is regarded as a collection of objects connected among them. IOTA [21] is the cryptocurrency used to fund transactions taking place among M2M communications and delivered services. This architecture is described as consisting of two applications: a) an extension for Google Chrome installed in devices owned by end users and b) a program that gathers data from sensors and writes them down on the Tangle used by IOTA. Overall, this system is oriented to users buying data from a marketplace where data collected from the Smart City will be sold, so its main purpose is different from what we put forward in our manuscript because it is oriented towards digital goods and it does not consider the particularities of such a system for developing countries. *Hermes*, a platform for trading sensor data that makes use of DLTs, is put forward in [22] to make stored data profitable. The architecture for this platform consists of several elements typical of blockchain marketplaces, such as wallets for every agent involved in the transactions and the usage of cryptocurrencies for payment procedures. IOTA is again used for this purpose. The architecture has been designed to have a streaming agent registering to the marketplace, streaming data to IOTA and, once it has identified, search for data of their interest. When a data buyer is found, funds will be transferred to the marketplace wallet and the streaming agent will be notified about the interest buyer and their public key. Finally, data and funds will be

transferred to the corresponding parties. Overall, this proposal focuses on selling data and is not related to the specific needs of developing countries. Jeong *et al.* explain in [23] the concept of the *Connected Car* and how they can benefit from the usage of blockchain. Their system has been especially conceived to deal with black box video data. *Blockchain Data-Owner-based Attribute-Based Encryption* (DO-ABE) is used as part of the data sharing scheme. The authors of this reviewed manuscript also mention *off-chain* technology, which consists of storing reference values in the blockchain that are linked to specific encrypted data that is stored outside of the blockchain. The system designed consists of a Data Owner (DO), a Data Consumer (DC) a blockchain network, a market platform and external storage (IPFS, cloud storage), whereas the specific vehicle data marketplace model is based on a decentralized application to upload and/or consume vehicle data. As before, this marketplace is not oriented to the end users or purposes than the ones shown in our manuscript.

It is shown in [24] how a marketplace can also be created for Fog/Edge computing resources. The authors attempt to offer an accurate view about the kind of tradeoff required to have efficient applications that can become as decentralized as possible. Their manuscript depicts a fully distributed software architecture for their electronic marketplace that consists of a public blockchain network, market clients and computational resources used for trading operations. The researchers have also considered in their manuscript the interactions of potentially interested parties. The decentralized market shown in this reviewed paper, though, refers to the possibility of purchasing both fog and/or edge computational resources rather than developing countries goods. Furthermore, [25] depicts how a real-world traceability system for the use case of traditional bakery can benefit from the use of blockchain. The authors describe a system working with Radio Frequency Identification (RFID) and Near Field Communication (NFC) for sensing equipment, along with blockchain itself for safely distributing and storing information through the supply chain. Combined, they guarantee traceability of the main features in the raw materials used to produce Carasau bread. It also makes use of IPFS, which enables further decentralization in the traced information about every aspect of the Carasau bread. However, the system has some significant differences compared to ours: a) theirs is oriented towards a specific product, whereas the one put forward in this manuscript involves many potential different kinds of them, b) their system is more oriented towards traceability and the one presented here is driven towards marketability for SSPs and c) the paper explicitly mentions the need for a central authority to regulate which parties will get into the system (a feature derived from the particularity of the foodstuffs that is being traced), whereas our solution makes use of PoA as the consensus algorithm for data control. It is also described in [26] how blockchain is a major contribution to data transparency in the Agri-Food industry. The proposed model makes use of three different layers (the first one deals with interactions among entities

in the supply chain, the second one contains the blockchain and copes with the transactional information of the trading and delivery operations, and the third one is used for data storage, also making use of IPFS to save data as decentralized as possible). Products are gathered in *lots* uniquely identified with lot numbers and traded according to Smart Contracts in the Ethereum network. Simulations carried out show that the usage of gas for the transactions that must be considered is acceptable. However, the system still relies on a centralized-like authority (referred to as *off-chain arbitrator*) to monitor the network or solve disputes (as opposed to using the mechanisms derived from consensus algorithms). Besides, while gas cost has been taken into consideration, there is little information about latency of the system when performing transactions, except for transaction mining times.

It is described in [27] how consensus mechanisms based on blockchain can be used for food traceability. The authors present a system where RFID and NFC are used to trace the supply chain procedures for the processing and delivery for foodstuffs. A consensus algorithm (*Proof Of Supply Chain Share* or *PoSCS*) specifically tailored for food supply chains has been developed to have the main stakeholders of the supply chain as the validators that create the new blocks. However, as it happened with the previous research work, the proposed system is focused on food traceability; while this is a major aspect in our own piece of research, the marketability of these products via website is given a greater weight than in the reviewed paper. Furthermore, the system presented by the authors of the reviewed manuscript tends to mimic PoS as the consensus algorithm, whereas we make use of PoA. It is also described in [28] how to add Deep Reinforced Learning to Agri-Food supply chains. The authors highlight how a *Deep Reinforcement Learning-based Supply Chain Management* (*DR-SCM*) method oriented towards making smart decisions on production and storage of foodstuffs to maximize profitability and summarize their contributions has been built with a) a blockchain framework that makes use of PoW as the consensus algorithm, b) the DR-SCM method itself and c) the experiments that have tested the performance of the proposed solution. They have identified the main participants of Agri-Food supply chains as actors in their system and make use of Deep Reinforced Learning to take actions in the overall deployment by interacting with the environment. However, this work is solely focused on Agri-Food traceability and offers no information about creating a market for the products, especially when they have been grown by SSPs in developing countries, thus having a purpose different from the one that we have formulated. Furthermore, the usage of PoW as the consensus algorithm presents a major difference compared to PoA. A piece of research resembling the ones shown before is depicted in [29]. In this case, the researchers have created a blockchain-driven framework for secure monitoring and reporting about Afro-Food goods delivered through the supply chain. Aside from the blockchain deployment that is performed in the supply chain, the authors included Supply Chain Management (SCM) backorder prediction procedures

that work with several machine learning algorithms. These research works, however, are more strongly focused on back-order prediction than in creating any kind of web market with the available information, so the purpose sought by the authors is different from the one that we are presenting.

Other development works to consider are the ones shown in [30]. The research activities contained in the reviewed manuscript deal with a system oriented towards food safety in China that makes use of blockchain and overall distributed Ledger Technologies (DLTs). They focus on the fact that market failures hinder decision-making processes for end customers. Three different kinds of contracts are used: Food Exit, Food Circulation and Food Sampling. As far as security is concerned, this solution proves that a malicious node uses an arithmetic power that can only be a fraction from an honest Ethereum node, with a measured chance of success not higher than 0.3%. While our solution is more oriented towards offering a web market for SSPs in developing countries, the reviewed manuscript proves that the Ethereum network provides a resilient tool against data tampering in distributed Smart Contracts. Another development to consider is the Origami Network [31] which self-defines itself as “*A protocol for building decentralized marketplaces using the Ethereum blockchain*”. The Origami Network makes use of three different self-sufficient platforms: *Origami Marketplace*, *Origami Payment*, and *Origami Review*. The first one is directed towards creating front-end developments (the authors describe how it makes use of a Representational State Transfer (REST) compliant (referred to as *RESTful*) API, a Front-office, and administration portal for sellers and another one for operators). The second one is described like “*a decentralized payment system powered by the Ethereum blockchain with decentralized escrow for secure payments*” that is claimed to require lower fees. Overall, this development aims to speed up transactions, minimize escrow payments and improve the marketplace reviews, but it is not oriented towards developing countries or SDGs and does not explicitly describe any procedure to add traceability to the goods and services that might be sold in it, as it is done with the sensor data provided in our proposal. Additionally, the Origami Network is more driven to providing a protocol to build marketplaces based on blockchain, rather than a particular marketplace. Another online market based on blockchain that has become important over the years is OpenBazaar [32]. It provides two mechanisms to protect end users’ funds when engaging in transactions: a) a reputation system that shows how trustable are sellers in this decentralized market (it works the same way as with other companies, like Amazon or eBay) and b) moderators that act as mediators in any dispute. Trade operations are done by having the buyer creating a Smart Contract between the digital signatures that participants use in this market (the one from the buyer and the one from the seller) and sending it to a moderator, and once the good has been received, the funds that were allocated for the operator will be sent to the seller. This is a fully decentralized system that makes use of blockchain to prevent

any centralized shutdown and it offers a significant degree of anonymity. However, it does not offer any default information about the growth or manufacturing of the products that are sold in it. Feasibility of the market itself is questionable too: at the moment of composing this manuscript, the company that maintains OpenBazaar (OB1 Company), is becoming unable to cover the operational costs of the platform [33]. ModulTrade [34] is another example of decentralized marketplaces based on blockchain. It is utilized to connect Small and Medium Enterprises (SMEs) to the global markets by making use of four different components: a) a blockchain based Smart Contract platform, b) a trade related services platform, c) a trade and reputation network and d) the marketplaces themselves. The developers of ModulTrade have created their own token used for trade operations. This platform also works with Smart Contracts as described before, in the sense that payments are locked and executed after the buyer has received the good rather than before. Developers mention that tracking the supply chain of the goods sold is possible, thus providing a way to check the authenticity and provenance of products. This marketplace offers a decentralized platform where traceability of products is explicitly mentioned, but it does not have an explicit orientation towards goods or services produced in developing countries and their more specific needs. There are also several proposals that show a clear interest in the topic of decentralized marketplaces based on blockchain. It is referred in the research work done by Chang et al. [35] that blockchain might enable increased security, trust, and privacy. To maximize these advantages, the authors of the manuscript point out that they have created a e-marketplace that makes use of self-enforced Smart Contracts that guarantee what payment or penalties will be applied. Also, it is said in [36] that a digital marketplace for IoT data can address successfully challenges typical of this application domain, such as limited resource and computational capabilities. In this way, a three-tier system architecture with a regulator, a facilitator and devices accompanied by participants is proposed. Lastly, it is claimed in [37] that, in addition to using blockchain for automatic and secure trading, secure data communication at the network layer based on Name Data Networking (NDN) can be used to upgrade security features.

## B. OPEN ISSUES

The described solutions offer their own advantages and disadvantages when compared to the main topics of this manuscript. A summary with all of those has been depicted in Table 1. Some of the disadvantages have been regarded as such due to a focus on different objectives than the one shown by us, rather than having any significant flaw.

The available research works regarding what kinds of marketplaces are available in developing countries to sell goods is profuse in variety. However, there are several open issues that have yet to be solved. They can be described as:

1. Blockchain-based supply chains tend not to offer a suitable front-end marketplace or alike: while the purpose of a supply chain is not necessarily offering a frontend with

TABLE 1. List of proposals with their advantages and disadvantages.

Proposal	Advantages	Disadvantages
Seitz et al. [11], <i>IIoT Bazaar</i>	Application for industrial Internet of Things.	No stress on developing countries or suitable infrastructure.
Li et al. [12], <i>LBRV</i>	Innovative name scheme.	No stress on developing countries or suitable infrastructure.
João Martins et al. [14]	Traceability in supply chains.	Methodology (reverse auction bid) might not adapt to the context of our manuscript.
Prince et al. [15]	AI is used for demand forecasting.	No stress on developing countries or suitable infrastructure.
Park et al. [16]	IoT data marketplace.	Physical goods are not considered.
Ranganthan et al. [17]	Integration of IPFS	No information about deployment. Infrastructure
Perboli et al. [20]	Blockchain application for Smart city. Usage of IOTA.	Physical goods are not considered.
Tzianos et al. [22], <i>Hermes</i>	Usage of IOTA.	Physical goods are not considered.
Jeong et al. [23]	Black box video data storage. Concept of off-chain storage.	Oriented towards connected cars.
Pincheira et al. [24]	Opens possibilities in Fog/Edge computing.	No stress on developing countries or suitable infrastructure.
Cocco et al. [25], <i>Carasau bread</i>	Realistic implementation with IoT wide-spread solutions	Only one specific product. Centralized authority is needed.
Shahid et al. [26]	Sophisticated and very systematic approach. Acceptable operation costs with Smart Contracts (gas)	Centralized authority is needed. Little information about system latency.
Tsang et al. [27], <i>BIFTS</i>	Tailored consensus algorithm (PoSCS). Usage of IoT hardware solutions.	Different purpose. Consensus algorithm less likely to be optimal
Chen et al. [28]	Innovative use of Deep Reinforced Learning. Experimental performance	Different purpose. Consensus algorithm less likely to be optimal
Bhutta et al. [29]	Framework for Supply Chain Management. Backorder predictions using machine learning algorithms.	Different purpose and focus (backorder predictions).
Yan et al. [30]	Tailored Smart Contracts. Proves	Oriented towards a different purpose (food

TABLE 1. (Continued.) List of proposals with their advantages and disadvantages.

	Ethereum network can be used in a very secure way	safety) than our proposal
<i>Origami Network</i> [31]	Attempt to minimize escrow payments. Based on self-sufficient platforms.	No stress on developing countries or suitable infrastructure. A protocol to create marketplaces instead of a marketplace.
<i>OpenBazaar</i> [32]	Usage of Smart Contracts. Mechanisms to guarantee good trade practices.	No stress on developing countries or suitable infrastructure. Financial difficulties to run the platform.
<i>ModulTrade</i> [34]	Usage of Smart Contracts. Product traceability. Integration of SMEs.	No stress on developing countries or suitable infrastructure. Requires own token.
Chang et al. [35], <i>E-marketplace</i>	Efforts to implement Smart Contracts and blockchain in marketplace.	Not enough information available.
Gupta et al. [36]	Efforts to implement Smart Contracts.	Not enough information available.
Yoo et al. [37]	Efforts to implement Smart Contracts.	Not enough information available.

information about information interchanges, having those data available could be a significant asset to know where value and/or cost are generated during a manufacturing process. This information is usually withheld from end users.

2. Marketplaces based on blockchain focus on data and digital assets rather than services or tangible, manufactured goods: in the reviewed literature, markets tend to be based around the idea of purchasing and selling data often relate to IoT deployment and projects. Having physical products or services provided by people as part of the marketplace is often overlooked.

3. Marketplaces are not focused on developing countries: marketplaces based or making use of blockchain tend not to be an option for goods produced in developing countries, since they tend to develop their main production activities in application domains that are not directly related to IoT or Cyber-Physical System (CPS) data.

4. Lack of technological descriptions: most of the solutions that have been included in the state of the art tend to describe how data are traded or what goods are available but offer scarce information about what kind of Smart Contracts are published in order to include new customers and goods in it, what procedures are used to include new participants are

followed or the infrastructure used to deploy the blockchain and their data.

### III. SOLUTION DESCRIPTION

The technical core of the infrastructure put forward in this manuscript is a blockchain-based, SDG-oriented marketplace where information about products has been stored and is available for every participant of the system. Blockchain adds some features unseen combined in a single platform: a) price formation is transparent, as prices in transactions can be seen by everyone, b) it makes use of a single Smart Contract, which accurately describes the transactions that are going to take place and c) it merges pricing and features data in the same system. Each of the subsections that has been added here describes how all these features have been achieved. The described system attempts to solve some of the open issues found in the studied literature regarding online markets for foodstuffs in developing countries. Indeed, there are three disadvantages that have been found for the latter, when compared to developed countries, that have shaped the proposed system to better target their end users (SSPs in developing countries) in the best possible manner:

- 1) Local, small-sized markets. Markets that are accessible by common SSPs in developing countries tend to be small and localized, which in turn imply that the number of clients willing to buy these products is usually not too large. The proposed system offers the possibility of making products available online for a larger audience, so it is expected that it will have a positive impact on the SSPs revenues.
- 2) Fragile supply chain. The sometimes-subpar transport networks or utility infrastructures in developing countries jeopardize the delivery of foodstuffs in time for their transformation or sell, which may render some of the products inedible or unfit for human consumption. With the usage of sensors providing freely available, reliable, and trustworthy data to the end users in the proposed system, it can be proven that the foodstuffs sold in the blockchain-based market have been grown or fattened in good environmental and overall conditions.
- 3) Lack of trust in the quality of sold products. Because of the previous issue, there is sometimes significant reluctance to buy products from SSPs in developing countries, even if they are offered according to the standards of production used where the products are sold. The available information put forward by the proposed system will help mitigate this lack of trust by adding transparent data accessible to the end users.

The system that is put forward in this manuscript uses a private Ethereum network to deploy the Smart Contracts used for trade operations. There are certain pieces of data transmitted, though, that are not related to Smart Contracts information, such as the environmental information (temperature, humidity, etc.) of the products that are sold, but they have been included due to their major usefulness for the

proposed system. The consensus algorithm chosen to validate transactions among the deployed network is PoA, which is different from the PoW consensus algorithm or the PoS one forecasted to be used by Ethereum 2.0. PoA is far better suited for environments with low computational resources like IoT developments in developing countries, as it does not require to have all the nodes obtaining and providing hash function outputs, which tends to be very energy consuming.

Overall, the system will follow a workflow that begins with the registration of the users from the web application. Once registered, they will add their products with sensor identifiers that will gather measurements periodically. When the product is available for sale, the owner will assign it. All products for sale will be available for any registered user with sufficient funds. To purchase them, users just must access the product page and buy what they like. When a product is purchased, its registered owner will change, and its identifier will be recorded in the blockchain certifying them as their legitimate owner.

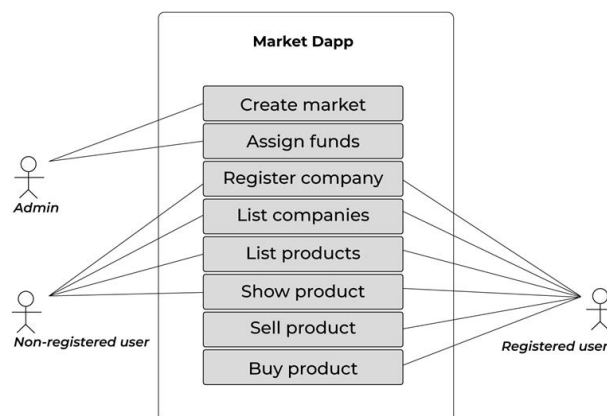


FIGURE 1. System use case diagram.

Figure 1 shows the use cases defined within the market. As it can be seen, the use cases have a straightforward mapping with the main activities that can be carried out in a trade website: *create market* (to deploy the terms used by the Smart Contract), *assign funds* (for trade operations among participants), *register company* (to include a company in the system), *list companies* (to have a list with all the companies selling products in the website), *list products* (to have a registry of all the products sold), *show product* (so as to display product features), *sell product* and *buy product*.

#### A. HIGH-LEVEL SYSTEM DESCRIPTION

Figure 2 illustrates the architecture of the system development as a high-level description. Its layered structure is made up by several subsystems that interact with each other to provide greater functionalities:

- 1) **Client:** it contains all the functionalities required for the client interaction with the website. On the one hand, a web browser used for navigation through the website is required to access the Graphical User Interface.



On the other hand, a wallet extension that allocates available funds is used as well.

- 2) **Presentation:** it represents the web visual interface, with a web server containing all its required presentation features. Its function is to facilitate the use of the application to the users. It works as the frontend of the system.
- 3) **Business:** its function is to establish communications between the frontend and the Smart Contract via an application server, providing the necessary resources to the website.
- 4) **Blockchain:** the core on which the application is built. It works as a network that manages the interactions between users and the contract (that is to say, the transactions taking place in the system). At the same time, it fulfils the function of the data layer by storing the Smart Contract state that can be consulted at any time. Thus, it contains the Smart Contract with the application logic that defines the functions required to ensure the correct operation of the market system.
- 5) **Sensor:** an element or group of elements external to the network. They are responsible for sending product data to the Smart Contract. By having information from sensors, accurate data is obtained from the environment where the foodstuffs to be sold are being taken care of. In this way, a large record of data can be obtained with regards of the good. What is more, optimal conditions of the products sold in the market can be proven, thus creating a way to guarantee high quality in the production sold by the SSPs.

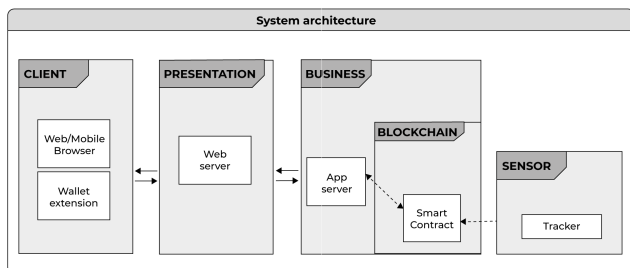


FIGURE 2. High-level system architecture.

As mentioned, the distributed system associated to this marketplace is based on Smart Contracts [38]. They can be defined as programs that become stored in a decentralized network (that is to say, the blockchain) and contain all the necessary business logic to be executed in data transfers. Even though Smart Contracts guarantee the interchange validity between products and funds, unfortunately for the time being Smart Contracts usually not regarded as a binding legal, conventional one in most of the cases (regardless of this, a workaround is put forward later). A Smart Contract consists of a) the logic, and b) a state based on the blockchain stored data, which can be edited and always consulted. Figure 3 showcases the structure of the data used organized in three basic entities:

- **Company:** refers to the agents that will make use of the application by selling products, purchasing them, or both. It is uniquely identified by an address and contains basic data related to identification and funds. A company can either be owner or buyer of none or multiple products.

- **Data:** refers to the monitored data in a particular time-frame associated to a single product, such as temperature, atmospheric humidity, carbon dioxide level, soil moisture, Ph level and geographical location. Data are represented by a string of characters.

- **Product:** refers to the element traded in the market. It is uniquely identified by a string that results from the calculation of a hash function. It also contains basic data given by the owner at the time of its creation, along with the data provided by the sensor in charge of monitoring it. It can have N associated structures, a single owner, and either one or no buyer. It has three different statuses: tracking, on sale and sold.

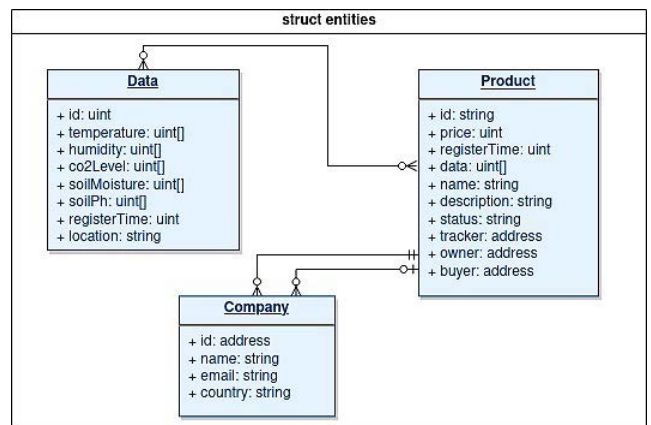


FIGURE 3. Smart Contract entities.

Every entity is stored in key/value maps, where the value will be an element of the previously defined structures (company, data, product) and the key its corresponding identifier. The methods required for the application logic will be the following:

- *createCompany* to register a company.
- *createProduct* to create and register a product.
- *addData* used by the sensor to add information to a product.
- *authorizeSale* to allow the owner to sell a product.
- *Payment* to allow a company to buy a product.
- *getProductData* is an auxiliary function to retrieve the data associated to a product added during implementation activities. This latter function is required due to a limitation in the Smart Contract programming language, which is unable to retrieve multiple nested structures. The implementation of the Smart Contract, as well as all the other pieces of work, are available in [39]. Such Smart Contract relies on the description via parameters of *Products* (with their information about name, identifiers, and status, among other features), *Companies* (with their identifiers, names, email,

etc.) and the addition of environmental *Data* (humidity, soil moisture, etc.) related to those products. Functions related to the addition of companies, products, their data and how sales are authorized are included as well. The most prominent of these functionalities are the main focus of the next subsection.

To deploy the Smart Contract defined in this system, a blockchain able to manage their associated technology is essential to do so. There are many blockchain platforms where Smart Contracts are defined, such as HyperLedger Fabric, NEO or Ethereum. Once the blockchain platform has been selected, the type of network must be decided (public, private, consortium or a hybrid between them), each having its advantages and drawbacks. The consensus algorithm (namely, the protocol in charge of ensuring the validity of the blocks generated) must be defined as well. Once these decisions have been made, the network architecture, the node quantity, its type, and hardware parameters must be set.

When the network is established, the Smart Contract can be hosted in the blockchain; it will become accessible through an address. There are two types of interaction with the Smart Contract. The first is a state check: it will be carried through calls, which are costless since they do not modify the data of the blockchain. However, it might be desirable to edit or add additional information to the data being transferred as inputs/outputs of Smart Contract codified functions. In this case, transactions must be used, which have a computational cost carried by the miner nodes. Therefore, a fee will be charged to the issuer of the transaction to pay the miners.

Since direct interaction with the Smart Contract requires knowledge that is not expected from end users to have (and in any case, the system put forward here must adapt to users and not vice versa), a frontend will be necessary to make users able to interact with the blockchain network through a website. Therefore, the resulting website will showcase the stored data in lists and tables and the online forms used for the registration of both companies and products with their corresponding buying and selling functionalities. At the same time, a backend with an API will be necessary to send, receive, process, and manage the transmitted data between the frontend and the Smart Contract.

In this context, sensors are responsible for collecting the product data, such as temperature, humidity, or soil Ph. These values are sent to the Smart Contract, where they will be processed to enrich the data from the stored product. Depending on the transmitting capabilities of the sensor, data will be sent to the Smart Contract directly, or even periodically if required. Lastly, the end users will be required to have a wallet with fundings belonging to the network to be used. A web extension to manage wallets and payments will guarantee a simpler user experience.

## B. SYSTEM BEHAVIOR AND PROCEDURES

Once a frontend is made available with products, company listings, data tables and statistics, the visual experience is made simpler to the users of the application. The data present in the frontend will be supplied by the backend calls to the

previously defined functions of the Smart Contract. Those functionalities that require the user's intervention, therefore regarded as transactions for the blockchain, are defined as the following:

- *Register company* (as depicted in Figure 5 with a sequence diagram): users or companies must be previously registered in the system to be able to use the application beyond data visualization. To do so, the user must access the "Register company" tab where they will fill in a form with the company's name, an email, and its billing address. Before sending the form, the user must accept the platform Terms and Services agreement. Afterwards, a tab with the payment information (operation costs) will be opened; the user must accept it for the transaction to be registered. This will store the company information onto the blockchain, uniquely identified by the payment's address. Only one address will be associated to one company. Issues might happen during the company registration procedure: a) the company could already be registered, b) it could have insufficient funds to enter the market or c) there might be missing information in the application form. Depending on the stage of these issues, an error message will be sent back to inform about the problem that arose during the company registration procedure.

- *Register product*: users registered in the blockchain can register products through the "Register product" tab on the website. A form must be filled in with the product's name and description, as well as the address of the sensor in charge of its monitoring. Once sent, a payment (operation cost) must be done. A single identifier will be assigned to the product, which will be the 'owner' field assigned to the payment address along with a timestamp to record its registered date and time. Initially, the product will have the "Tracking" status. As it happened before, issues might take place while registering a product. The most typical are a) the product to be registered cannot be linked to a company address, b) funds might be insufficient after trading

operations to register the product or c) information could be missing from the form required to be filled. As it happened before, error messages will be sent in the suitable moments where these problems are found out. All these steps have been displayed in the sequence diagram of Figure 6.

- *Product tracking*: as shown in Figure 4, a product in "Tracking" status will be monitored by a sensor or groups of sensors until it is put on sale. These sensors will periodically send information to the Smart Contract with the temperature, atmospheric humidity, carbon dioxide level, soil Ph and soil moisture, geographical location, and measurement timestamp values. The Smart Contract's function must manage the errors that may occur because of an incorrect product identifier or if the product has a different sensor assigned.

- *Authorize sale*: once the product is available for sale, the owner must shut down the sensors and access the product web page to assign it a sale price. This will change the status of the product to "On Sale" and the price will then be registered (Figure 7). The company whose address is stored in the "owner" field will be the only one able to use this

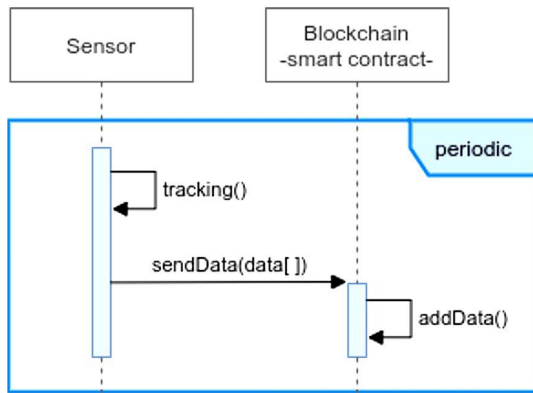


FIGURE 4. Product tracking procedure.

function. Typical issues that could happen in this procedure are that a) the buyer has insufficient funds to perform such an operation or b) the price of the good intended to be bought is wrong or missing. As in previous cases, error messages will be sent when required to alert about any problem happening.

•**Buy product:** any product with the “On Sale” status can be purchased by any previously registered user except by the current owner. To do so, the product page will display its price and a button to process the order (Figure 8). In this case, the transaction price will be the established price for the product plus the operational cost. The previous owner will receive the former price. The buyer will then become the new owner and the purchase will be registered in the blockchain. Should there be insufficient funds to perform such operation, an error message will be sent back to the client trying to buy the product. Note that despite the similarities among the different functionalities described here, their behavior has been described in the previous UML figures in an independent manner in case they are ported to other related developments in the future.

### C. CYBERSECURITY FEATURES

As with other developments that involve blockchain, there are several cybersecurity features that must be considered as well. The system that is put forward here makes use of several hardware entities that, as with the other features related to security, have been included as part of a security threat analysis depicted in Figure 9. Specifically, the following hardware elements are present:

- 1) The buyers computer used to connect to the server where the web market is located.
- 2) The server itself, used to a) store the several web pages used by the web market, b) receive the information from the product sensors and validate it via PoA and c) take part as a node in the Ethereum private network where Smart Contracts are deployed, and data are validated.
- 3) The sensors that collect information from the products to be sold, which take part in the blockchain deployment put forward in the system and provide critical data

about the conditions on how foodstuffs were produced during their lifetime.

- 4) The other nodes from the private Ethereum network, which have been included as the ones used for data interchanges related to Smart Contract executions and information about transactions.

All these hardware entities can be eventually threatened by security attacks, as information sent from/to these devices in a bidirectional manner (except for the sensors used to collect environmental data, that are conceived just to push information towards the location of the web page). Besides, there are three processes related to communications in the system that are taking place:

- 1) Requests based on web service features being interchanged between the buyer’s computer and the server where the web market site is located.
- 2) The usage of the PoA consensus algorithm to make the required validations for the blockchain-based deployment of the system.
- 3) The operations performed by the server within the Ethereum private network as a node that has deployed Smart Contracts on it.

Lastly, there are three boundaries of different nature set within different parts of the system, which have also been represented in Figure 9. To begin with, a boundary is set between the buyers’ computer and the server with the web market files, which is equivalent to the web service-based requests and responses performed to retrieve data from the market. A second border is set in the blockchain used by the sensors, which is built specifically for the system and is sending information towards the web site. Lastly, there is a third border separating the Ethereum private network (that the system server belongs to, as it is used to deploy the Smart Contracts) from the other parts of the system.

Overall, it has been estimated that the elements that could be more prone to cyber-attacks are the three kinds of hardware components used to interchange data throughout the proposed system: a) the *buyer’s computer* used to interact with the market that is being displayed to the potential customers, b) the *sensors* used to monitor and collect information about how the raw materials for the foodstuff elaboration are treated and c) the *web server* used to deploy the web market that has been stored in it. Additionally, the Ethereum network could also be affected by security attacks. The decentralization of the interchanged data is another aspect to consider as well. It is not possible to have a 100% guaranteed secured system, even if it was centralized and worked offline and out of any kind of network or information sharing technology (human operatives or physical location of the hardware could be prone to security attacks as well). Fortunately, the proposed system relies on technologies like blockchain, as well as the cryptographic facilities blockchain is built upon, that increase the security and authenticity of the information made available. A paradigmatic example of this would be *Man-In-The-Middle* (MITM). This form of cyberattack aims to obtain information from a private data interchange between two parties and/or

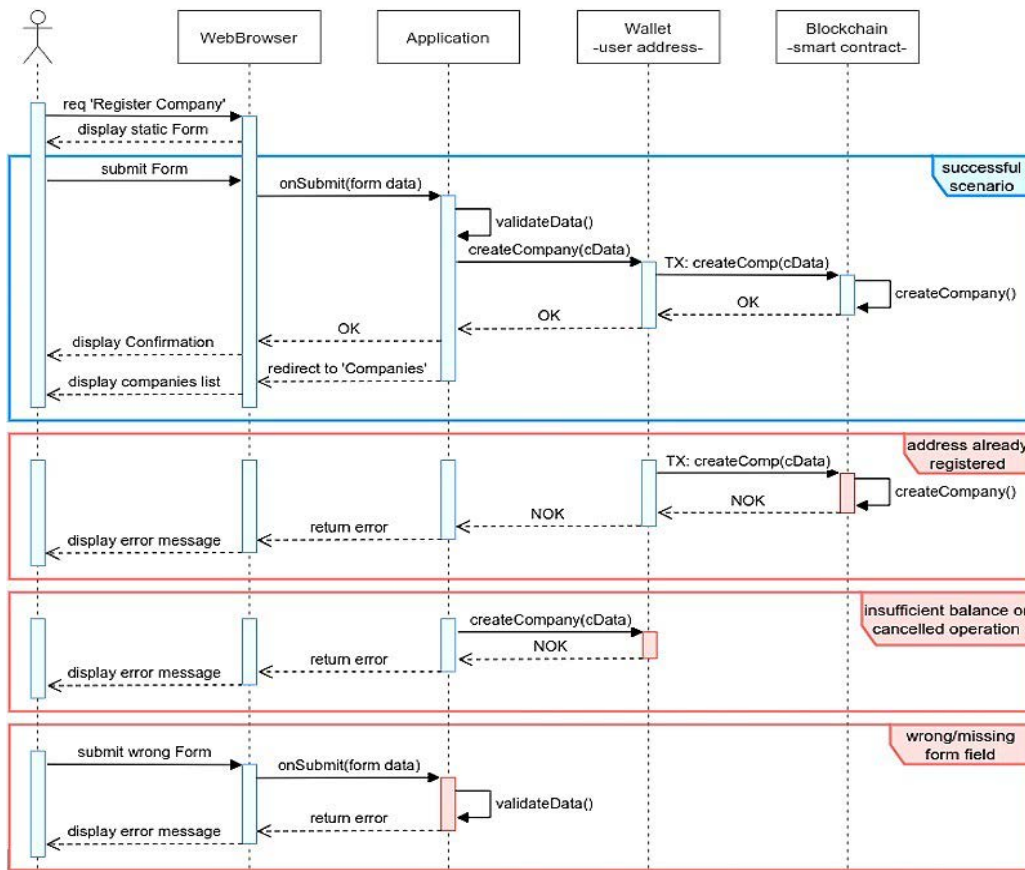


FIGURE 5. Register company procedure.

distribute and share malicious content while it is inserted in the conversation, even if such parties are IoT or CPSs components such as vehicles [40]. MITM-based cyberattacks in the proposed system pose a minimal threat due to several aspects:

- 1) An authentication system based on registration and access credentials for SSPs can be enabled in the system as a countermeasure against malicious agents attempting to break into the system.
- 2) Once the SSPs are correctly registered, important information about their activities (i.e., their identities, account balance) does not need to be transmitted, except for wallet identifiers that become public anyway in the Smart Contracts that are available to the blockchain users. Most data that will be shared among the system will be related to the primary sector goods and foodstuffs that the SSPs are taking care of, as well as their environmental and border conditions; this data will also be made public for buyers that want information about the products they are interested in.
- 3) Information transmitted through the proposed system is very likely to come from different places at the same or almost the same time (the infrastructure that each SSP uses to transmit and store data in the blockchain shared

by all the participants in the system), so the MITM attack would need to eavesdrop a significant number of the physical communication links to effectively alter the information transmitted (as it is required by all the blockchain nodes that pieces of information are the same regardless of their location), which depending on the number of users could become too costly and impractical.

- 4) Consensus algorithms in blockchain work in a way that it is required that many, the majority or even all the nodes participating in a blockchain agree to validate specific information as the valid one. This is no different for the case of the PoA algorithm used in the proposed system, where the nodes taking part of the Ethereum private network have their reputation attached to their blockchain identifier (not their actual identity) and are therefore motivated to keep a good one. If after validating a piece of data a party forges it in any way, it would become obvious what blockchain identifier did it, the data tampering could be mitigated, and the malicious user expelled from the system. Furthermore, if the Ethereum public network was to be used instead, it should be taken into consideration that the deployed Smart Contract not only makes use of

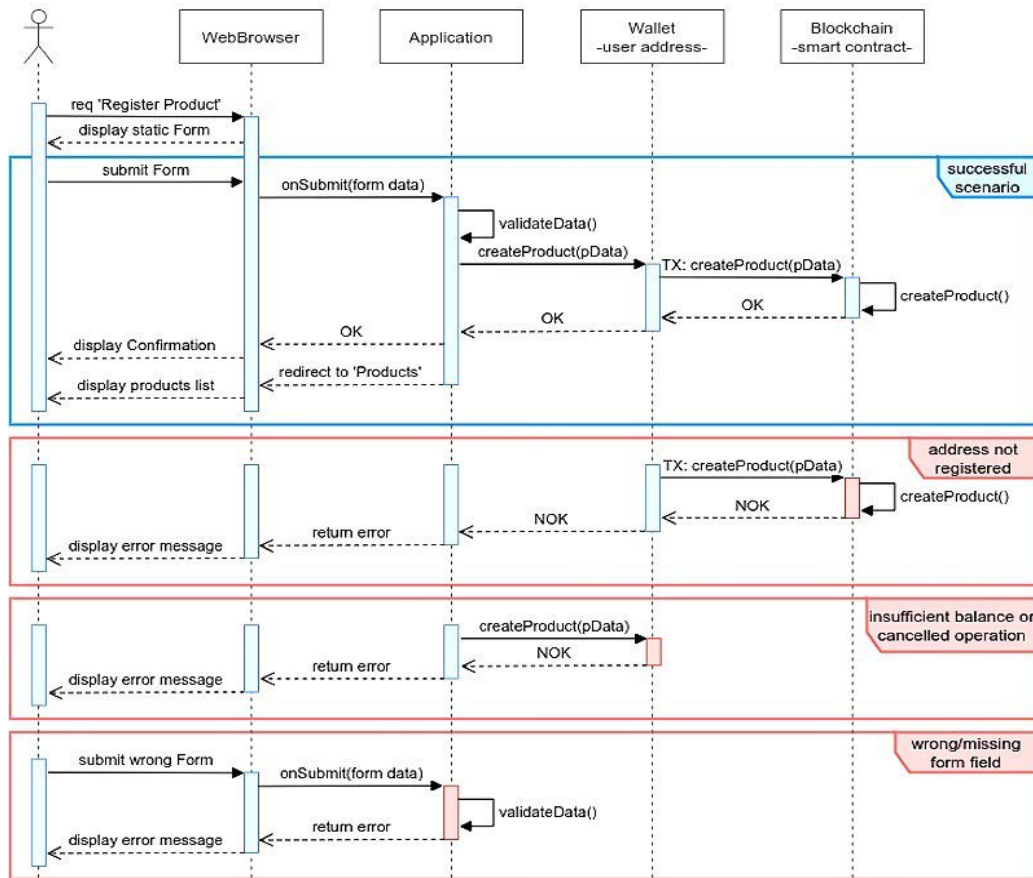


FIGURE 6. Register product procedure.

cryptographical libraries (as explained in the following point) but also would make use of a distributed system (the Ethereum network itself) that currently uses PoW as the consensus algorithm, which adds up to the proposed system characteristics. Indeed, performing a 51% attack [41] would demand that the attacker has more computational power than most of the validating nodes of the attacked blockchain. As it was explained before, this becomes quickly unfeasible if a certain number of users is reached. The fact that PoW is likely to be phased out in favor of PoS [42] in the Ethereum public network does not change the essentials of this expected performance.

- 5) Currently, the underlying Ethereum private network used in the implementation of the proposed system uses ECDSA (Elliptic Curve Digital Signature Algorithm) as asymmetrical cryptographical libraries used to digitally sign information pieces. Among other security procedures, this implies the usage of hash outputs that daisy-chain with each other (in blockchain, a current block hash is obtaining from both their data and the previous block hash). As far as ECDSA is concerned, the fact that Elliptic Curve Cryptography (ECC) is used makes 283-bits public keys comparable to 3072-bit

public keys for the RSA algorithm should come as one additional major countermeasure against data tampering in the system [43]. The Keccak-256 function [44] is also used by the Ethereum network where the Smart Contracts of the proposed system are deployed for data hashing.

- 6) Security can also be incorporated in other parts of the developed system that lie below the blockchain-based data. HTTP Secure (HTTPS) can be used for data transmission among the application level, which in turn would employ Transport Layer Security (TLS) for secure information transfer below purely data-based layers, thus providing further security countermeasures to the proposed system that would expand beyond what blockchain can offer in the system.
- 7) As far as other attacks are concerned, the countermeasures provided by the system prove to have similar effectiveness, as they rely on the same keys aspects of the system that have been describe before. For example, the Denial of Service (DoS) attack can be countered with security infrastructure of proven worthiness (firewalls that add rules based on IP addresses and ports, DoS Defense Systems or DDSs for protocol-based and data rate-based attacks, etc.). In addition to DoS, its

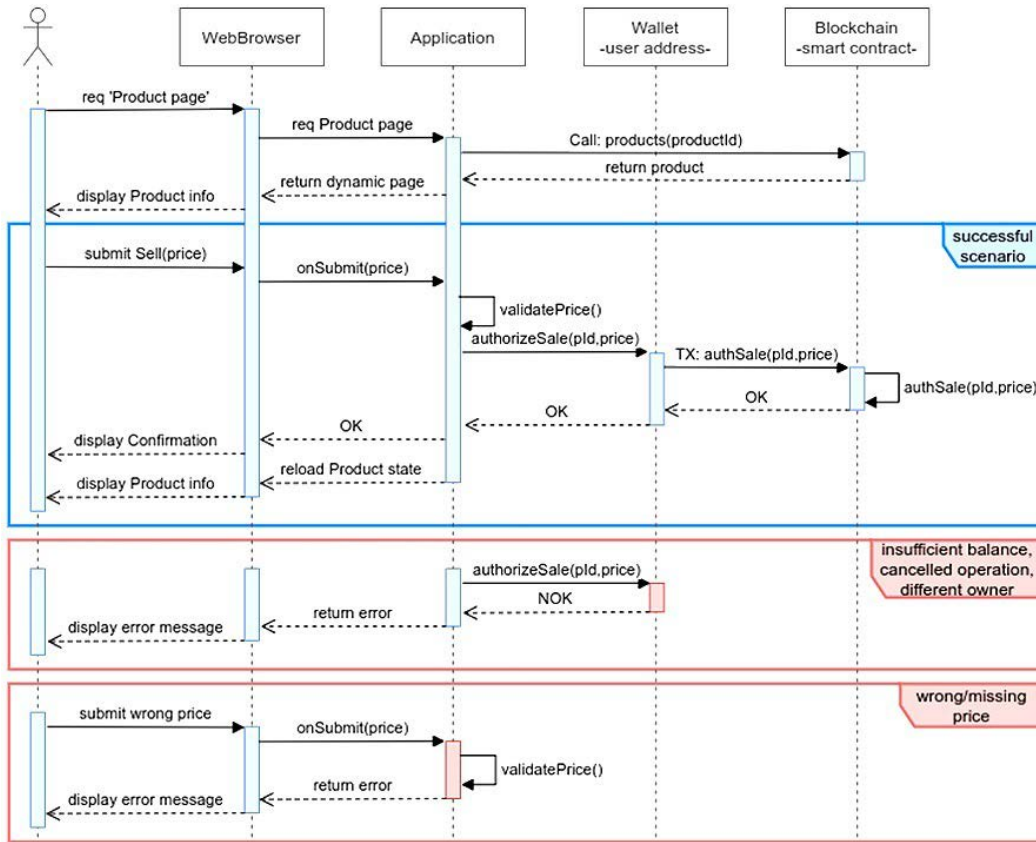


FIGURE 7. Authorize sale procedure.

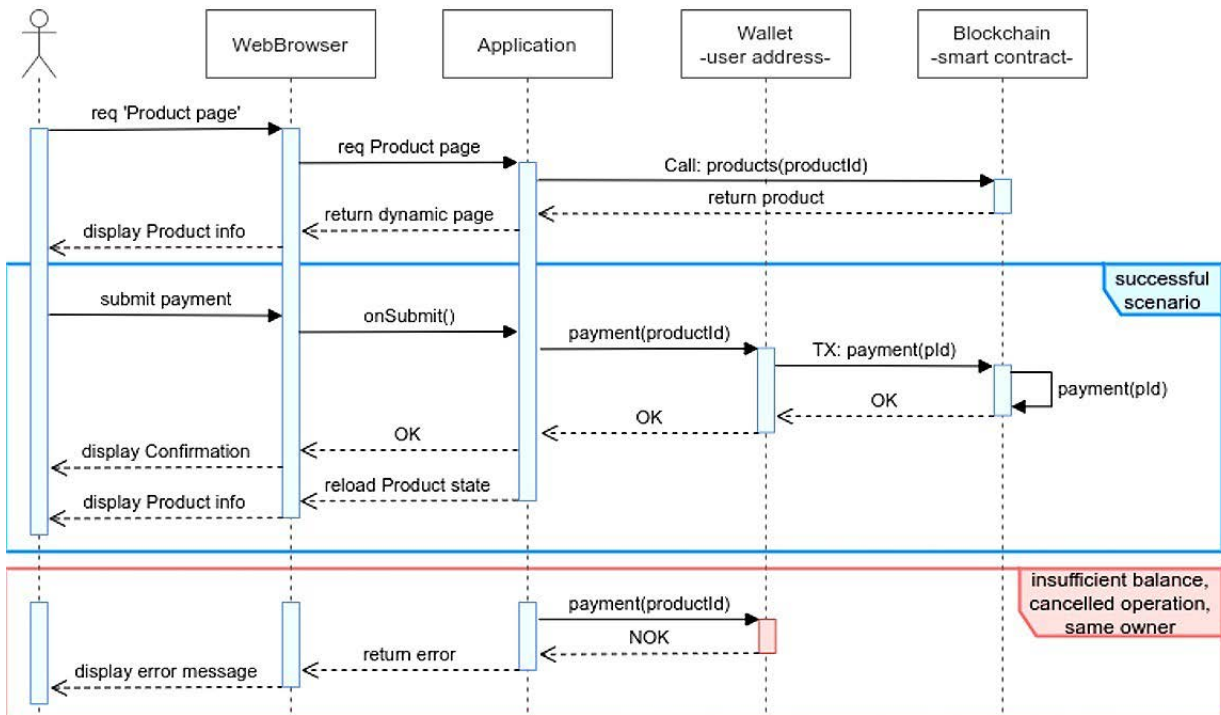


FIGURE 8. Buy product procedure.

distributed alternative (Distributed Denial of Service or DDoS), which could in theory use against the sensors

providing data in the system, would have minimal to no chances of success, due to the fact that the data sent

to the server is sent at the same pace for every node and they are verified and shared among the users of the system via PoA consensus algorithm, which makes it easy to exclude any malicious device acting as a legitimate sensor.

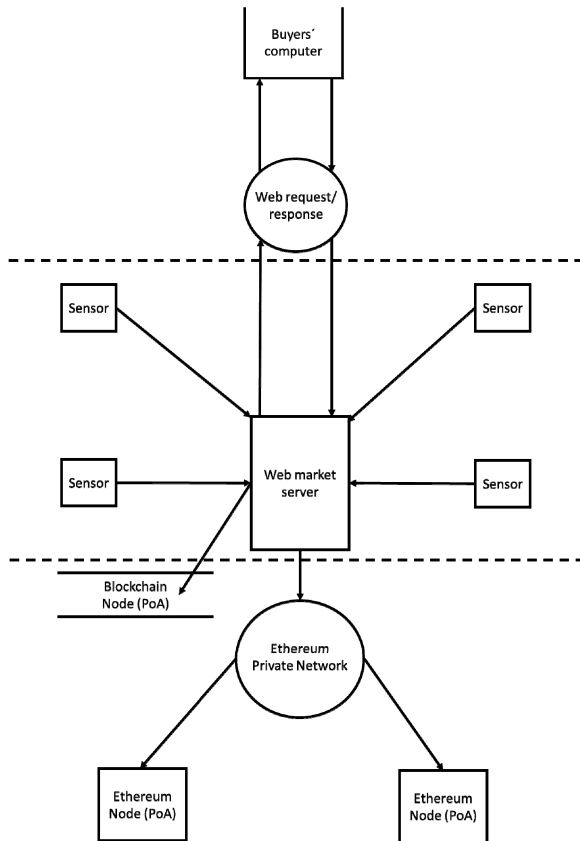


FIGURE 9. Security threat analysis for the proposed system.

- 8) Additionally, spoofing attacks face the same challenges mentioned in previous point 4 that makes them unlikely to be successful: any party attempting to add tampered data to the blockchain would require to validate the fraudulent transaction via PoA; this attempt would demand a large proportion of the computational resources within the system that a single node is unlikely to have. Therefore, the data forging would be discovered, and the spurious node could be expelled from the system.

When all is said and done, the overall security measures described would be deployed in a layered manner, as depicted in Figure 10. Note that the security measures or protocols included in the figure as bold and italic characters define what layers are providing security for the end user.

**IV. SYSTEM IMPLEMENTATION AND TESTING**

As previously mentioned, the implementation of the proposed solution has been carried out in a cloud environment, as it was deemed by the authors of this manuscript that implementation and testing activities were mandatory to test that the idea

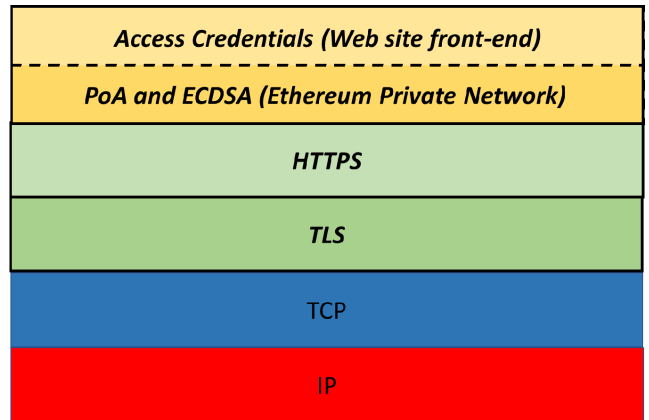


FIGURE 10. Layered locations of the security measures that can be used in this system.

of having a blockchain-based market running with acceptable performance. Microsoft Azure [45] has been chosen to provide the required infrastructure. During the development and integration process of the application in the cloud it was used as a service under a Platform-as-a-Service (PaaS) model. However, clients make use of the cloud as a service based on the Software-as-a-Service (SaaS) model. This is because the application is provided directly without having to manipulate any resources in the cloud nor having to access Azure services.

**A. SYSTEM IMPLEMENTATION**

Since a blockchain network is required to deploy the Smart Contract, the network provided by Ethereum has been regarded as the most suitable one [46]. Due to the large amount of documentation available for Ethereum, this technology makes easy using existing networks and creating new ones.

Because of the greater control in the customization parameters, the authors of this manuscript have opted for a private blockchain network to find the optimal ones for the proposed solution. However, a consensus network could be used too; it must be born in mind that the latter differs from the private in the authority owner of the nodes involved in the mining and signing of blocks. In this case, public entities with reliance and trust such as universities or government entities.

Ethereum nodes conform the network though constant communication among them at the data level. The network architecture that was deployed consisted of three different nodes. Two are hosted on two Virtual Machines (VMs) in the Azure cloud and another one on a local computer. Its creation and network deployment follows these steps:

1. Firstly, the cloud resources are deployed: two VMs with OS Ubuntu 16.04 are deployed. Each of them has a different security resource group to have a decentralized network. Both machines have static and public IP addresses. An implementation based on the minimum requirements set by the Ethereum protocol is necessary to host the nodes. Its specific settings are the following:

a. Node-1: It is in a VM in the B2s Azure Cloud with 2 vCPUs, 4GBs of RAM and a Solid-State Drive (SSD). Node-1 is hosted in the Western Europe region. It is a full node responsible for mining functions.

b. Node-2: It is in a VM in the B2s Azure Cloud with 2 vCPUs, 4GBs of RAM and an SSD. It is hosted in the Northwestern America/ Northwest America region. It is a full node in charge of the mining functions.

c. Node-3: It is in a local machine in Madrid, Spain. It is a light node; it validates the new blocks created by the mining nodes of the network.

2. A workspace is created in each of the nodes and Go Ethereum [47], also known as 'Geth' -an implementation of the Ethereum protocol that enables us to create customizable Ethereum Virtual Machines (EVMs) nodes- is installed.

3. The mining nodes require an account or wallet to operate on the blockchain. Two are created through Geth, and they are assigned a private password to unlock them every time their services are needed, such as start/stop the mining operations, send transactions, etc.

4. A common first block known as Genesis Block is necessary to create a blockchain or to join an already existing one. The new blocks created in the blockchain will be concatenated from this block. This block will contain the key parameters of the network's functioning. The Puppeth tool in Geth [48] will allow the creation of the file interactively from the terminal. The Genesis block that has been created for deployment purposes is depicted below. It uses 778 bytes of storage, which also gives an idea of what size the other blocks that will be linked to the blockchain are (despite the differences in the figures and parameters that they include with regards to the transactions that have been included). Each block is regarded as compact enough not to use too many resources, but large enough to provide information details about transactions and the block itself. {

```
"config": {
  "chainId": 13542,
  "homesteadBlock": 0,
  "eip150Block": 0,
  "eip150Hash": "0x00",
  "eip155Block": 0,
  "eip158Block": 0,
  "byzantiumBlock": 0,
  "constantinopleBlock": 0,
  "petersburgBlock": 0,
  "istanbulBlock": 0,
  "clique": {
    "period": 5,
    "epoch": 30000
  }
},
"nonce": "0x0",
"timestamp": "0x60803235",
"extraData": "0x00",
"gasLimit": "0x47.760",
"difficulty": "0x1",
"mixHash": "0x00",
"coinbase": "0x00",
"alloc": {
```

```
"ee219800286f77e4ccd388b04a1b63454e20330d":
{
  "balance":
  "0x1500000000000000000000000000000000000000000000000000000000000000",
  },
"number": "0x0",
"gasUsed": "0x0",
"parentHash": "0x00"
}
```

Other aspects to consider are as follows:

a. PoA is the chosen consensus algorithm since the nodes are managed by a trusted entity. It reduces the large computational load and the energy required by other algorithms like PoW in the creation of new blocks.

b. The time used for block creation has been limited to 5 seconds. By doing so, we achieve a fast response blockchain that can support many transactions without reaching saturation.

c. Afterwards, the wallet addresses used for block generation in the validation procedures are added. Initial funds are also added to the wallets for them to be able to operate.

d. We assign the *chainId*, a numeric identifier required to connect to the blockchain through peering. The assigned value must be exclusive to our blockchain.

e. The JavaScript Object Notation (JSON) file generated is copied in each of the nodes and initialized with Geth.

5. Once the nodes are established, the communications between them and with the rest of the components interacting with the network must be enabled, such as the backend and the sensor. To do so, the firewall policies of the machines (virtual and local) are modified by enabling the http:80 and rpc:30000 ports and limiting their access to the IP addresses of the system.

6. Node access is made through the EVM console. It can be done in two possible ways: a) either locally through the machine where the node is hosted using Secure Shell (SSH) if the machine is remote or b) using the RPC (Remote Procedure Call) with Geth from one of the already enabled IP addresses. Once the node console is accessed, the wallet assigned to the node is unlocked and the mining process is started, thus enabling the creation of new blocks and transaction processing in the network. The overall structure of the cloud-based components, and how the different technologies have been used, are depicted in Figure 12. Once the blockchain is established and initialized, the Smart Contract can be deployed. It has been written in Solidity. This is an object-oriented, high-level language resembling JavaScript aimed at creating Smart Contracts in Ethereum [49]. The code developed for the system is compiled in two files. The first file is based on Bytecode, a low-level programming code that is understandable by an EVM. The second file is Application Binary Interface (ABI), a JSON-written interface that enables us to interact with the contract by using calls and transactions with the backend. Since both files are required by the system, they must be saved inside the project files.



A JavaScript-written script was codified to deploy the contract. The script uploads the contract in the blockchain by making use of a transaction. The Bytecode is sent via the ABI, processed, and hosted in an exclusive and static blockchain address. This transaction requires both a blockchain wallet with sufficient funding to assume the transaction costs, and Web3js, a library collection that allows remote interaction with a node through RPC. Once the blockchain network is established and the contract is deployed, the application is fully functional. Nevertheless, a prototype for a website has been written to facilitate the tasks. As mentioned before, this component is hosted on a VM type A2v2 in the Azure cloud, along with 2 vCPUs, 4GBs of RAM and an SSD. It is hosted in the Western Europe region with a static public IP address. The front-end has been developed in Hypertext Markup Language (HTML), Cascade Style Sheet (CSS) and JavaScript. The ReactJs library [50] has simplified the web design; the backend makes use of JavaScript in a NodeJs environment [51]. As it has been depicted in Figure 11, the website has been designed with each of the use cases that are required to be run in mind, keeping a 1:1 mapping with the design activities described in the previous section.

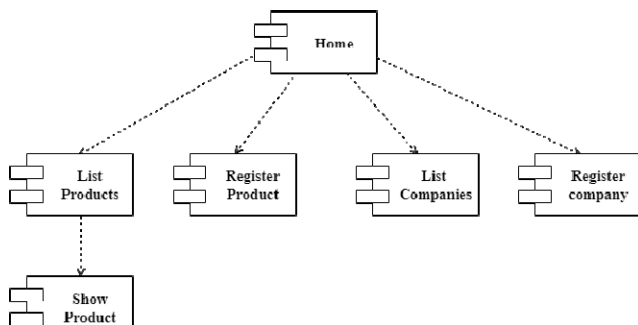


FIGURE 11. Frontend architecture.

Each of these components can be described in the following way:

- 1) Home: the default page of the application (Figure 13, top). Like it happens with web pages, it has a navigation bar at the top to access the rest of the functionalities. The system logo is displayed in the center, and the lower part displays general statistics from the application.
- 2) List Products: a table listing of the products registered in the system (Figure 13, bottom). Each of the columns represent the most representative data of the product, with its status identified with colors for an easy recognition as to whether a product is for sale or not. The last column is a button to access the product page where further information about the product can be found.
- 3) Show Product: a specific page for each product with detailed information. In the case of having monitoring data, information, and statistics of the product monitored will appear together with a drop-down list with each data input provided by the sensor and a workflow

on top (Figure 14, top). When the product is in “tracking” status, it will have a box for the owner to assign a selling price to the product. Before the sale takes place (and therefore, right before the Smart Contract is enforced), the terms and conditions for concluding the legally binding Smart Contract can be shown throughout the website. In this way, and although price and features of the product to be purchased are already known at this point, the terms of the purchase can be further clarified. This terms and conditions notice would make the Smart Contract easier to enforce in legal scenarios where there are constraints for them to be accepted, since they add an element of notice (end users can find the agreements, read them and accept or decline them), provided that they are shown under fair conditions [52]. These conditions would be: a) *Clear and conspicuous notice of terms* (the terms are available at a conspicuous location or, in case of the proposed system, when giving the final agreement to the Smart Contract), b) *active acceptance* (users must explicitly agree on the terms put forward to them), c) *controlled access* (no access will be given to the available products unless the clients have agreed to the terms and conditions before), d) *periodic reviews* (notification of changes in the Smart Contract terms, in case legislation makes it necessary) and e) *location* (agreements easy to find in the website). Figure 15 shows an example of how these terms and conditions can be depicted in a way that are representing the information provided by the Smart Contract. Usually, they must be as detailed and accurate as possible; that is why Figure 15 has a large number of different contents to consider (Agreement to Terms, Intellectual Property Rights, etc.).

- 4) Once this price is assigned, the workflow will change its status to “on sale” and a “Buy” button will appear with the price of the product. Finally, when the product is purchased, it will change to the final status, “Sold” (Figure 14, bottom).
- 5) Register Product: a page that contains a small form with the necessary text fields to register a new product in the system.
- 6) List Companies: the list of companies and their blockchain addresses that are part of the system and can make use of the marketplace functionalities.
- 7) Register Company: a page with the form to register in the system. It is necessary to accept the terms of service.

Since the data layer of the set architecture is based on the state of the Smart Contract, the web requires constant communication with the contract to display updated data. This communication will be done through contract invocations via data calls, dependent on the network’s response time, along with its subsequent processing and display of the data times. With NextJs (the NodeJs framework), any necessary call will be made at the server-side before the user browses throughout the web pages. By doing this, it is possible to optimize data fetching tasks and reducing loading times. The

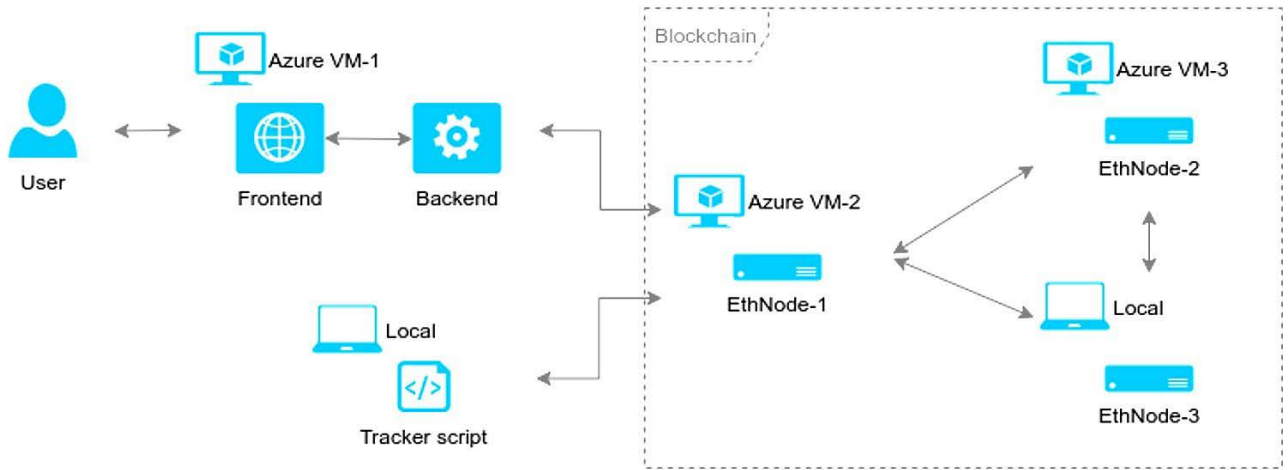


FIGURE 12. Overall cloud-based architecture.

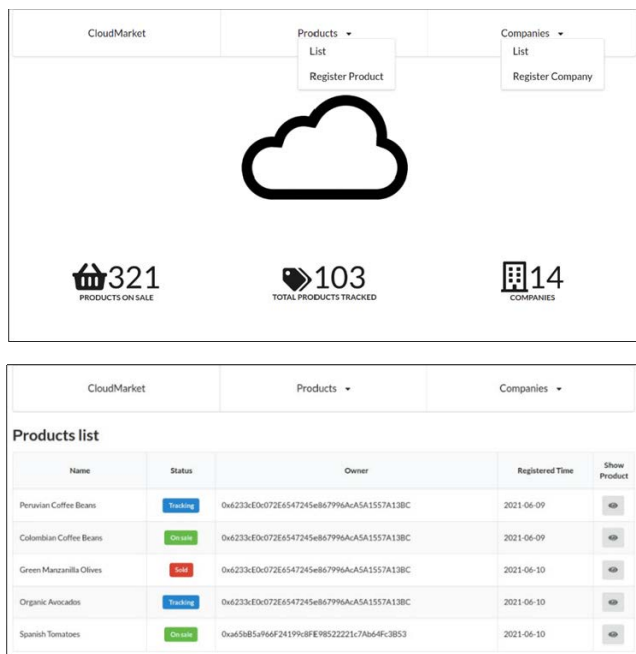


FIGURE 13. Default web page (top), product list (bottom).

framework also facilitates the web routing tasks, thus achieving an overall better user experience. We use the Web3js library to manage the calls and transactions in the backend. By means of RPC, the library makes use of the ABI file of the Smart Contract to communicate with it in its assigned address.

As mentioned above, transactions modify the contract state and therefore require a payment of the operation cost. To simplify this task, Metamask is used as the tool to deal with those transactions. The backend is responsible of managing these operations, whereas the user will confirm or reject payments through a pop-up window.

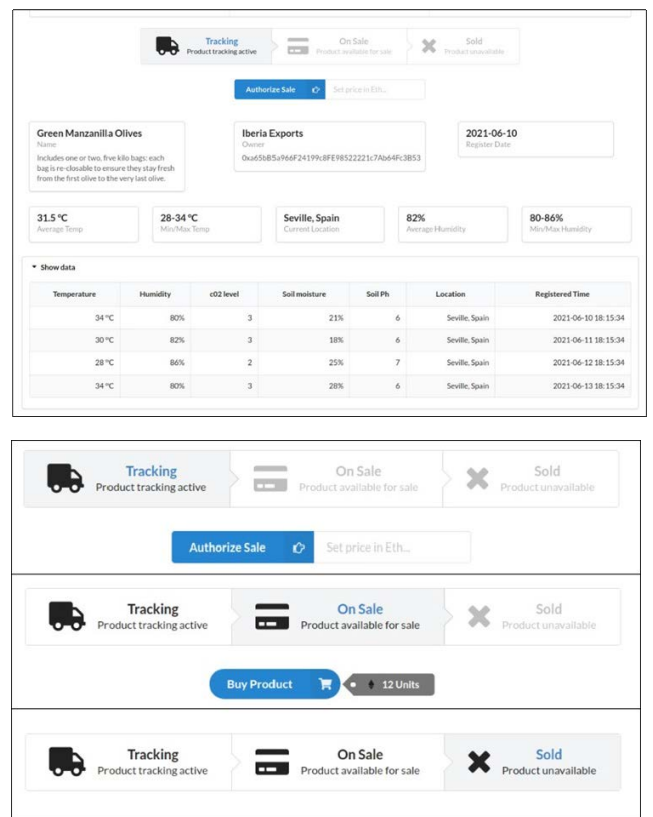


FIGURE 14. Product default web page (top) and workflow progress (bottom).

Finally, the expected functionality of the sensors in an environment that would go beyond the laboratory has been simulated through a script written in Python. This script periodically sends randomized data (within acceptable ranges) directly to the Smart Contract. While data sources would be very different from what is defined in this example, how data are transferred or stored are done in the same way, so it can

## Terms and conditions

These terms and conditions ("Agreement") set forth the general terms and conditions of your use of the [bmarket.com](http://bmarket.com) website ("Website" or "Service") and any of its related products and services (collectively, "Services"). This Agreement is legally binding between you ("User", "you" or "your") and this Website operator ("Operator", "we", "us" or "our"). If you are entering into this agreement on behalf of a business or other legal entity, you represent that you have the authority to bind such entity to this agreement, in which case the terms "User", "you" or "your" shall refer to such entity. If you do not have such authority, or if you do not agree with the terms of this agreement, you must not accept this agreement and may not access and use the Website and Services. By accessing and using the Website and Services, you acknowledge that you have read, understood, and agree to be bound by the terms of this Agreement. You acknowledge that this Agreement is a contract between you and the Operator, even though it is electronic and is not physically signed by you, and it governs your use of the Website and Services. This terms and conditions policy was created with the help of the [terms and conditions generator](#).

### Accounts and membership

You must be at least 18 years of age to use the Website and Services. By using the Website and Services and by agreeing to this Agreement you warrant and represent that you are at least 18 years of age. If you create an account on the Website, you are responsible for maintaining the security of your account and you are fully responsible for all activities that occur under the account and any other actions taken in connection with it. We may, but have no obligation to, monitor and review new accounts before you may sign in and start using the Services. Providing false contact information of any kind may result in the termination of your account. You must immediately notify us of any unauthorized uses of your account or any other breaches of security. We will not be liable for any acts or omissions by you, including any damages of any kind incurred as a result of such acts or omissions. We may suspend, disable, or delete your account (or any part thereof) if we determine that you have violated any provision of this Agreement or that your conduct or content would tend to damage our reputation and goodwill. If we delete your account for the foregoing reasons, you may not re-register for our Services. We may block your email address and Internet protocol address to prevent further registration.

### User content

We do not own any data, information or material (collectively, "Content") that you submit on the Website in the course of using the Service. You shall have sole responsibility for the accuracy, quality, integrity, legality, reliability, appropriateness, and intellectual property ownership or right to use of all submitted Content. We may, but have no obligation to, monitor and review the Content on the Website submitted or created using our Services by you. You grant us permission to access, copy, distribute, store, transmit, reformat, display and perform the Content of your user account solely as required for the purpose of providing the Services to you. Unless specifically permitted by you, your use of the Website and Services does not grant us the license to use, reproduce, adapt, modify, publish or distribute the Content created by you or stored in your user account for commercial, marketing or any similar purpose.

### Backups

We are not responsible for the Content residing on the Website. In no event shall we be held liable for any loss of any Content. It is your sole responsibility to maintain appropriate backup of your Content. Notwithstanding the foregoing, on some occasions and in certain circumstances, with absolutely no obligation, we may be able to restore some or all of your data that has been deleted as of a certain date and time when we may have backed up data for our own purposes. We make no guarantee that the data you need will be available.

### Links to other resources

Although the Website and Services may link to other resources (such as websites, mobile applications, etc.), we are not, directly or indirectly, implying any approval, association, sponsorship, endorsement, or affiliation with any linked resource, unless specifically stated herein. We are not responsible for examining or evaluating, and we do not warrant the offerings of, any businesses or individuals or the content of their resources. We do not assume any responsibility or liability for the actions, products, services, and content of any other third parties. You should carefully review the legal statements and other conditions of use of any resource which you access through a link on the Website. Your linking to any other off-site resources is at your own risk.

**FIGURE 15.** Terms and conditions for legally binding contract.

be proven that the distributed, blockchain-based marketplace for developing countries shown here can work successfully.

Once the solution has been designed and implemented it must be tested. Since every subsystem has a different function, each of them will be tested in a specific manner. Therefore, tests based on functionality, security and performance are necessary.

## B. SYSTEM PERFORMANCE

The system has been developed with a Test-Driven Development-based methodology. Thus, specific tests have been conceived before writing the code that will pass them. This verifies that the Smart Contract methods operate correctly. A test collection for the backend to ensure a proper communication with the blockchain and the Smart Contract has been carried out as well. Moreover, the collection certifies that calls and transactions are functioning correctly, and potential errors managed. We have also carried out another set of tests aimed to evaluate the security of the network and communications, both internal and external, related to the cloud environment. The key elements to test in a Smart Contract are the cost and time that its deployment take, as well as the use of its codified functions through transactions in the blockchain.

The timeframe when these actions are confirmed is known as the latency of the blockchain network. The analysis of these values has been performed unitarily, that is, invoking each method only once. Each block in the network has an 8,000,000 Gas limit (as in the Ethereum main network), so it is greater than the cost of each method in Gas. A new block is created each 5 seconds in the blockchain, so the estimated time to process a transaction will be between 0 and 5 seconds. This timespan will be kept as long as the blockchain is not saturated of transactions. If there are more transactions

queued to be processed than the capacity of a single block, the transaction will be queued and, depending on the status of congestion or saturation, it can end up being processed after multiple blocks. It must be noted that Gas values obtained per transaction are subject to the amount of data sent. It is shown in Table 2 how the processes that require sending multiple fields of information are indeed more expensive in gas cost, while those that require only one field have a much lower cost.

**TABLE 2.** Smart Contract deployment and method costs.

Operation	Gas cost
Deployment	1,751,552
CreateCompany	137,001
CreateProduct	174,987
AddData	178,484
AuthorizeSale	71,521

Once it has been ensured that the costs and latency of the Smart Contract methods are within the established parameters and observed a correct and fast functionality, a stress test can be performed. In this test, it is evaluated how the system performs against a massive use of simultaneous users. To simulate the latter situation, transactions are repeatedly sent using the *createProduct* function. This is the only error-free method that allows transactions with identical data since the rest of functions contain requirements that do not allow the repetition of certain fields. This makes possible that the costs of each transaction are identical. Before carrying out testing activities, an estimation of latency for the *createProduct* transaction repeated 100 times with a cost of 174,987 Gas

was calculated by using the following operations:

$$\begin{aligned}
 \text{transactions per block} &= \frac{\text{block gas limit}}{\text{transaction gas cost}} \\
 &= \frac{8,000,000}{174,987} = 45.71 \rightarrow 45 \quad (1) \\
 \text{number of blocks} &= \frac{\text{number of transactions}}{\text{transactions per block}} \\
 &= \frac{100}{45} = 2.22 \rightarrow 3 \text{ or } 4 \text{ blocks} \quad (2)
 \end{aligned}$$

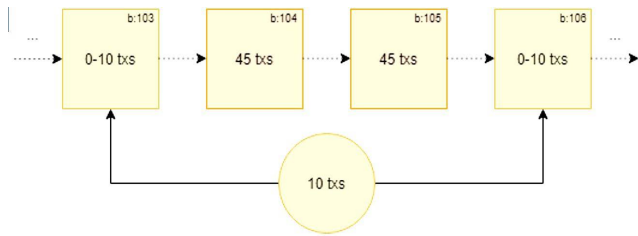


FIGURE 16. Distribution of transactions.

It can be seen in (1) how the transactions included in each block are the result of dividing the gas limit established for blocks in the network (set at 8,000,000 as explained before) by the gas cost of each *createProduct* transaction (as shown in Table 2 ), so that once the gas limit for the block is depleted no more transactions will be included in the block. The number obtained in (1) has been rounded down to an integer because transactions without enough gas to be wholly completed cannot be added to the block and will require another one. Related to this, the number of blocks required for the transaction that takes place 100 times is shown in (2). Since it would imply that 100 transactions must be saved in blocks with a storage capacity of 45 transactions per block, 2.22 blocks would be needed. However, as far as the number of blocks is concerned, the transactions that are “left over” after filling the rest of the blocks can be found either in a single additional block or two blocks, following the pattern in Figure 16 (any number higher than 2 blocks will require at least one additional block, as blocks are numbered as integer entities).

It can be observed in Table 3 that, despite some (minimal) variation with respect to the estimated latencies, the value correlation follows an almost constant trend, matching the expected interval. This proves that, even in a state of saturation, the network is not only resilient enough to withstand all transactions, but also it operates as expected even with up to 100,000 simultaneous users. At this point, though, performance severely worsens and becomes evident that for larger amounts of users, more capable resources would have to be deployed.

Lastly, the website’s performance must be evaluated both in its frontend and backend. The latency that results in accessing and navigating among the web site pages shown to the end users is the key factor. The time it takes to confirm an action

TABLE 3. Stress tests results.

Users	Estimated latency	Resulting latency
1	0-5s	4.41 s
10	0-5 s	4.33 s
100	15-20 s	16.60 s
1,000	115-120 s	118.42 s
10,000	1115-1120 s	1,119.16 s
100,000	11115-11120 s	11,117.32 s

such as registering is limited by the latency of the previously tested blockchain, so that is not taken in consideration for these testing operations.

From the data obtained in the Table 4 it can be determined that the web has a decent response time, partially aided by a large volume of cached data that helps in the responses time of the pages previously visited.

TABLE 4. Web performance values.

Page	Load times	Memory resources
Home	1.82 s	16.8 Mb
List products	2.87 s	16.7 Mb
Show product	3.62 s	32.8 Mb
Register product	3.12 s	22.3 Mb
List companies	1.89 s	16.6 Mb
Register company	3.78 s	22.8 Mb

While the deployed Smart Contract should have no memory shortage problems, there is a physical limit based on the memory of the node itself. In this case, a 280 GiB SSD has been used (also, storage used by the operating system and programs must be considered as well). In the case of the Ethereum main network, the entire blockchain, including all deployed smart contracts, uses a total of 1.184 Terabytes under Geth (the Ethereum client written in GO, or 574 Gigabytes (under Parity, which is written in Rust) at the moment of writing this manuscript, as it can be seen in [53]. Therefore, it is most likely that there will not be any problems of storage in the short or medium term. In any case, it should be noted that if at some point in the future the nodes reach their data storage limit, storage space can be added dynamically for the cloud facilities where the Smart Contract is running. This is one of the most significant advantages of making a deployment in the cloud.

C. SYSTEM TESTING RESULTS ANALYSIS AND DISCUSSION

Bearing in mind the results obtained, it can be considered that the conceived system is robust enough and fulfills its aimed functionality (offering a secure, efficient, and dynamic solution along with a good interaction with the user). The system can work efficiently providing service for up to 100,000 users performing simultaneous operations. While performance decreases as new users are added to the system,

it is still able to carry out the expected trade operations and close interchanges at a satisfactory enough pace for end users. It must be taken into consideration that the performance of the testbed shown in this manuscript has been deployed with three nodes enabled with medium to low computational power (4GBs in both cases). Although an impact in performance is shown, it also proves that this kind of devices can be used to power trade operations, which in turn make possible the usage of cheaper devices in these systems, which is a crucial feature to consider for developing countries. However, to maintain satisfactory performance figures, additional pieces of hardware must be used to keep up with the inflow of new users.

Fortunately, it is a system with a great and simple scalability where new nodes can be added without their geographical location being of any importance. However, the latency of the blockchain used to store data might become an issue when it is saturated with transactions. There are many possible solutions to speed up the processing rate: reducing the generation time of new blocks, increasing the capacity of each block, creating a database to manage the pending transactions, etc. Nevertheless, blockchain makes up for this limitation in scalability by providing a high security level in the exchange of goods on the web.

## V. CONCLUSION AND FUTURE WORKS

After testing and evaluating the different parts of the system, we can assert that the developed application meets the needs of any end user that demands this kind of application. It establishes a market system that is secure, transparent, immutable, and distributed due to its decentralization. The combination of a web-based frontend and a blockchain backend has allowed us to develop a system that enables the exchange of goods between users in a secure, efficient and user-friendly way. Although the use of Smart Contracts might create challenges due to their perceived rigidity when it comes to correcting possible bugs in the code, the advantages provided far outweigh these possible limitations. Cloud computing technology also provides a way of avoiding the difficulties linked to the maintenance of local servers, while enabling total control over the consumption and scalability of the system. Furthermore, the system conceived here uses blockchain to its advantage to provide security that relies on its very design, thus offering a significant degree of security against several major attacks like Man in the Middle, forged data transfers or eavesdropping.

In short, this project lays the foundations for the symbiosis that already exists between the technologies used. Each of the technologies brings its advantages in addition to solving many of the disadvantages of the other. As mentioned before, the implementation works that the authors of this manuscript have also been made available for the research community in [39].

As for future works, the next steps to follow in the development of the system lie in the addition of new functionalities to the system. The most interesting functionalities would be:

- 1) Addition of a transaction explorer page on the blockchain that updates dynamically and a system for registering and managing application users, hence defining roles and being able to restrict functionalities to unregistered users.
- 2) Improvements to the depicted website, which should first focus on improving the effectiveness and efficiency of the website in terms of speed as well as a visual refactor given its current simplicity.
- 3) Currently, the information associated with products and companies is simple enough. Yet the addition of new fields such as images of the products would be essential to increase the confidence of customers and it would help to improve the visual appearance of the website. This improvement would require some changes in the code of the website and the Smart Contract.
- 4) Integration of other goods or services that are related to utilities (water, electricity) if the existing infrastructure makes possible their transfer. Research works have been done [54] that prove it is possible to create Smart Contracts for energy transfers through the power grid.
- 5) The general features of the sensors to be used must be considered. While sensors capable of taking accurate measures are preferable, it is of major importance to guarantee that the data that will be gathered from them have not been tampered with, otherwise the blockchain deployment will be rendered useless. To prevent this issue, tamper-proof sensors can be used to measure data. Such pieces of hardware already exist and are of widespread use [55], [56]. Another option is preparing the sensors to have physical anti-tamper enhancements, such as encasements with tamper proof screws [57]. This latter procedure will require additional work and costs to be put in the sensors, but it can be beneficial for installation or data audits.
- 6) The proposed system could also benefit from further enhancements that will go beyond what is been proposed in this manuscript. For example, instantaneous traceability of the information regarding foodstuffs production could be provided by equipping sensors with REST interfaces that provided data any given moment that they are requested to do so. Besides the sensor enhancements put forward before, another aspect to consider is the replacement of sensors, either due to battery exhaustion or unexpected malfunctions. Battery and wireless signal levels must be provided as part of the information offered by raw materials for foodstuffs obtention. In this way, some of the potential limitations in the proposed system can be addressed.

- 7) Legislative improvements can be considered in the implemented solution as well. As explained in the System implementation and Testing section, Smart Contracts may receive over time an equivalent legal status to the one that regular written contracts have, so additions could be done in the Graphical User Interface that will allow system users to sign up Smart Contracts. This would be done by adding the terms and conditions for legally binding contract, if they are stated in a very clear manner whenever a purchase order is going to be executed, so that Smart Contract information could be filled up before deploying it to the blockchain.
- 8) There are some other synergies that could also be considered in the system. For example, since blockchains are built upon a consensus algorithm defined by the participants of the system and can show data to every need in a transparent manner, they can enable collective intelligence to participate in the system, so that the consensus algorithm (which can be regarded as social procedures from a different perspective) will become more educated and focused on end users' feedback and behavior [58].

Finally, it must be kept in mind that blockchain is a technology that, compared to other ones (Internet, the web), is still in its early stages. Therefore, monitoring its development is necessary to find out new possible applications and functionalities for it.

## ACKNOWLEDGMENT

These research activities have been done with the authors' involvement in the Sustainability-Aware IoT Systems Driven by Social Communities (SIoTCom) Project [59], where the creation of a blockchain-based and distributed architecture that benefits from consensus derived from collective intelligence plays a significant role. Additionally, the authors of this paper would like to thank the Non-Governmental Organization KUBUKA-Más Por Ellos [60] for the support and the information regarding how best to conceive a system like the one put forward in this manuscript. Thanks to their feedback, the usefulness of this system can be maximized while keeping a realistic perspective.

## REFERENCES

- [1] L. Mutegei, T. Wanyoike, J. Sevilla, J. Olukuru, T. Mberi, and T. Weru, "Unlocking the supply of open government data for SDGs: A case of Kenya national bureau of statistics (KNBS)," in *Proc. IST-Africa Week Conf. (IST-Africa)*, May 2017, pp. 1–11, doi: [10.23919/ISTAFRICA.2017.8102304](https://doi.org/10.23919/ISTAFRICA.2017.8102304).
- [2] W. Li, H. El-Askary, V. Lakshmi, T. Piechota, and D. Struppa, "Earth observation and cloud computing in support of two sustainable development goals for the river Nile watershed countries," *Remote Sens.*, vol. 12, no. 9, p. 1391, Apr. 2020, doi: [10.3390/rs12091391](https://doi.org/10.3390/rs12091391).
- [3] J. I. Sudusinghe, R. P. Jayaratne, and A. S. Kumara, "UN SDGs shaping sustainable supply chains: The case of apparel manufacturers in developing countries," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Informat. (SOLI)*, Jul. 2018, pp. 102–107, doi: [10.1109/SOLI.2018.8476697](https://doi.org/10.1109/SOLI.2018.8476697).
- [4] T. Yaméogo, A. Bossa, B. Torou, J.-L. Fusillier, D. Da, Y. Yira, G. Serpantié, F. Somé, and M. Dama-Balima, "Socio-economic factors influencing small-scale farmers' market participation: Case of Rice producers in dano," *Sustainability*, vol. 10, no. 12, p. 4354, Nov. 2018, doi: [10.3390/su10124354](https://doi.org/10.3390/su10124354).
- [5] Microsoft Corporation. (2020). *Smart Farming Innovations for Small-Scale Producers*. [Online]. Available: <https://gcgh.grandchallenges.org/challenge/smart-farming-innovations-small-scale-producers>
- [6] United Nations. (2017). *United Nations General Assembly. Resolution Adopted by the General Assembly on 6 July 2017: Work of the Statistical Commission pertaining to the 2030 Agenda for Sustainable Development*. [Online]. Available: [http://ggim.un.org/documents/a\\_res\\_71\\_313.pdf](http://ggim.un.org/documents/a_res_71_313.pdf)
- [7] P. Katila, C. C. Pierce, J. W. De, G. Galloway, P. Pacheco, and G. Winkel, Eds., *Sustainable Development Goals: Their Impacts on Forests and People*. Cambridge, U.K.: Cambridge Univ. Press, 2019, doi: [10.1017/9781108765015](https://doi.org/10.1017/9781108765015).
- [8] S. Yu, K. Lv, Z. Shao, Y. Guo, J. Zou, and B. Zhang, "A high performance blockchain platform for intelligent devices," in *Proc. 1st IEEE Int. Conf. Hot Inf.-Centric Netw. (HotICN)*, Aug. 2018, pp. 260–261, doi: [10.1109/HOTICN.2018.8606017](https://doi.org/10.1109/HOTICN.2018.8606017).
- [9] S. Cakic, A. Ismailisufi, T. Popovic, S. Krco, N. Gligoric, S. Kupresanin, and V. Maras, "Digital transformation and transparency in wine supply chain using OCR and DLT," in *Proc. 25th Int. Conf. Inf. Technol. (IT)*, Feb. 2021, pp. 1–5, doi: [10.1109/IT51528.2021.9390117](https://doi.org/10.1109/IT51528.2021.9390117).
- [10] Q. Yang and H. Wang, "Blockchain-empowered socially optimal transactive energy system: Framework and implementation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3122–3132, May 2021, doi: [10.1109/TII.2020.3027577](https://doi.org/10.1109/TII.2020.3027577).
- [11] A. Seitz, D. Henze, D. Miehle, B. Bruegge, J. Nickles, and M. Sauer, "Fog computing as enabler for blockchain-based IIoT app marketplaces—A case study," in *Proc. 5th Int. Conf. Internet Things: Syst., Manage. Secur.*, Oct. 2018, pp. 182–188, doi: [10.1109/IoTSM.2018.8554484](https://doi.org/10.1109/IoTSM.2018.8554484).
- [12] J. Li, A. Grintsvayg, J. Kauffman, and C. Fleming, "LBRY: A blockchain-based decentralized digital content marketplace," in *Proc. IEEE Int. Conf. Decentralized Appl. Infrastruct. (DAPPS)*, Aug. 2020, pp. 42–51, doi: [10.1109/DAPPS49028.2020.00005](https://doi.org/10.1109/DAPPS49028.2020.00005).
- [13] A. Grintsvayg and J. Kauffman. *LBRY: A Decentralized Digital Content Marketplace*. Accessed: Aug. 30, 2020. [Online]. Available: [https://lbry.tech/spec?\\_ga=2.205781972.673969195.1630307882-618346550.1629970238](https://lbry.tech/spec?_ga=2.205781972.673969195.1630307882-618346550.1629970238)
- [14] J. Martins, M. Parente, M. Amorim-Lopes, L. Amaral, G. Figueira, P. Rocha, and P. Amorim, "Fostering customer bargaining and E-procurement through a decentralised marketplace on the blockchain," *IEEE Trans. Eng. Manag.*, vol. 69, no. 3, pp. 810–824, Jun. 2022, doi: [10.1109/TEM.2020.3021242](https://doi.org/10.1109/TEM.2020.3021242).
- [15] P. W. Khan, Y.-C. Byun, and N. Park, "IoT-blockchain enabled optimized provenance system for food industry 4.0 using advanced deep learning," *Sensors*, vol. 20, no. 10, p. 2990, May 2020, doi: [10.3390/s20102990](https://doi.org/10.3390/s20102990).
- [16] J.-S. Park, T.-Y. Youn, H.-B. Kim, K.-H. Rhee, and S.-U. Shin, "Smart contract-based review system for an IoT data marketplace," *Sensors*, vol. 18, no. 10, p. 3577, Oct. 2018, doi: [10.3390/s18103577](https://doi.org/10.3390/s18103577).
- [17] V. P. Ranganathan, R. Dantu, A. Paul, P. Mears, and K. Morozov, "A decentralized marketplace application on the Ethereum blockchain," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2018, pp. 90–97, doi: [10.1109/CIC.2018.00023](https://doi.org/10.1109/CIC.2018.00023).
- [18] MetaMask. *MetaMask—A Crypto Wallet & Gateway to Blockchain Apps*. Accessed: Aug. 30, 2020. [Online]. Available: <https://metamask.io/>
- [19] IPFS. *What is IPFS*. Accessed: Aug. 30, 2020. [Online]. Available: <https://docs.ipfs.io/concepts/what-is-ipfs/>
- [20] S. Musso, G. Perboli, M. Rosano, and A. Manfredi, "A decentralized marketplace for M2M economy for smart cities," in *Proc. IEEE 28th Int. Conf. Enabling Technol., Infrastruct. Collaborative Enterprises (WET-ICE)*, Jun. 2019, pp. 27–30, doi: [10.1109/WETICE.2019.00014](https://doi.org/10.1109/WETICE.2019.00014).
- [21] IOTA Website. *IOTA*. Accessed: Jul. 28, 2022. [Online]. Available: <http://wiki.iota.org/learn/about-iota/an-introduction-to-iota>
- [22] P. Tzianos, G. Pipelidis, and N. Tsiamitros, "Hermes: An open and transparent marketplace for IoT sensor data over distributed ledgers," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 167–170, doi: [10.1109/BLOC.2019.8751331](https://doi.org/10.1109/BLOC.2019.8751331).
- [23] B.-G. Jeong, T.-Y. Youn, N.-S. Jho, and S. U. Shin, "Blockchain-based data sharing and trading model for the connected car," *Sensors*, vol. 20, no. 11, p. 3141, Jun. 2020, doi: [10.3390/s20113141](https://doi.org/10.3390/s20113141).

- [24] M. Pincheira, M. Vecchio, and R. Giaffreda, "Rationale and practical assessment of a fully distributed blockchain-based marketplace of fog/edge computing resources," in *Proc. 7th Int. Conf. Softw. Defined Syst. (SDS)*, Apr. 2020, pp. 165–170, doi: [10.1109/SDS49854.2020.9143892](https://doi.org/10.1109/SDS49854.2020.9143892).
- [25] L. Cocco, K. Mannaro, R. Tonelli, L. Mariani, M. B. Lodi, A. Melis, M. Simone, and A. Fanti, "A blockchain-based traceability system in agri-food SME: Case study of a traditional bakery," *IEEE Access*, vol. 9, pp. 62899–62915, 2021, doi: [10.1109/ACCESS.2021.3074874](https://doi.org/10.1109/ACCESS.2021.3074874).
- [26] A. Shahid, A. Almogren, N. Javaid, F. A. Al-Zahrani, M. Zuair, and M. Alam, "Blockchain-based agri-food supply chain: A complete solution," *IEEE Access*, vol. 8, pp. 69230–69243, 2020, doi: [10.1109/ACCESS.2020.2986257](https://doi.org/10.1109/ACCESS.2020.2986257).
- [27] Y. P. Tsang, K. L. Choy, C. H. Wu, G. T. S. Ho, and H. Y. Lam, "Blockchain-driven IoT for food traceability with an integrated consensus mechanism," *IEEE Access*, vol. 7, pp. 129000–129017, 2019, doi: [10.1109/ACCESS.2019.2940227](https://doi.org/10.1109/ACCESS.2019.2940227).
- [28] H. Chen, Z. Chen, F. Lin, and P. Zhuang, "Effective management for blockchain-based agri-food supply chains using deep reinforcement learning," *IEEE Access*, vol. 9, pp. 36008–36018, 2021, doi: [10.1109/ACCESS.2021.3062410](https://doi.org/10.1109/ACCESS.2021.3062410).
- [29] M. N. M. Bhutta and M. Ahmad, "Secure identification, traceability and real-time tracking of agricultural food supply during transportation using Internet of Things," *IEEE Access*, vol. 9, pp. 65660–65675, 2021, doi: [10.1109/ACCESS.2021.3076373](https://doi.org/10.1109/ACCESS.2021.3076373).
- [30] L. Yan, S. Yin-He, Y. Qian, S. Zhi-Yu, W. Chun-Zi, and L. Zi-Yun, "Method of reaching consensus on probability of food safety based on the integration of finite credible data on block chain," *IEEE Access*, vol. 9, pp. 123764–123776, 2021, doi: [10.1109/ACCESS.2021.3108178](https://doi.org/10.1109/ACCESS.2021.3108178).
- [31] Origami Network. *Origami Network—Build Your Own Marketplace Using an Ethereum Blockchain Powered Protocol*. Accessed: Jan. 26, 2022. [Online]. Available: <https://ori.network/>
- [32] Bit2Me. *What is OpenBazaar and How Does it Work?*. Accessed: Jan. 26, 2022. [Online]. Available: <https://academy.bit2me.com/en/que-es-openbazaar/>
- [33] CoinDesk. *OpenBazaar Co-Founder Explains Why Web 3's Answer to eBay Folded its Tents*. Accessed: Jan. 26, 2022. [Online]. Available: <https://www.coindesk.com/business/2021/07/15/openbazaar-co-founder-explains-why-web-3s-answer-to-ebay-folded-its-tents/>
- [34] ModulTrade. *Our Viable Product*. Accessed: Jan. 26, 2022. [Online]. Available: <https://modultrade.io/>
- [35] Y.-W. Chang, K.-P. Lin, and C.-Y. Shen, "Blockchain technology for e-Marketplace," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 429–430, doi: [10.1109/PERCOMW.2019.8730733](https://doi.org/10.1109/PERCOMW.2019.8730733).
- [36] P. Gupta, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Towards a blockchain powered IoT data marketplace," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2021, pp. 366–368, doi: [10.1109/COMSNETS51098.2021.9352865](https://doi.org/10.1109/COMSNETS51098.2021.9352865).
- [37] H. Yoo and N. Ko, "Blockchain based data marketplace system," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 1255–1257, doi: [10.1109/ICTC49870.2020.9289087](https://doi.org/10.1109/ICTC49870.2020.9289087).
- [38] J. M. Montes, C. E. Ramirez, M. C. Gutierrez, and V. M. Larios, "Smart contracts for supply chain applicable to smart cities daily operations," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, Oct. 2019, pp. 565–570, doi: [10.1109/ISC246665.2019.9071650](https://doi.org/10.1109/ISC246665.2019.9071650).
- [39] M. P. Lamela. *Smart Contract Code (File: CloudMarket.sol)*. Accessed: Aug. 30, 2020. [Online]. Available: <https://github.com/mpereira15/CloudMarket>
- [40] F. Ahmad, F. Kurugollu, A. Adnane, R. Hussain, and F. Hussain, "MARINE: Man-in-the-middle attack resistant trust model in connected vehicles," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3310–3322, Apr. 2020, doi: [10.1109/JIOT.2020.2967568](https://doi.org/10.1109/JIOT.2020.2967568).
- [41] F. A. Aponte-Novoa, A. L. S. Orozco, R. Villanueva-Polanco, and P. Wightman, "The 51% attack on blockchains: A mining behavior study," *IEEE Access*, vol. 9, pp. 140549–140564, 2021, doi: [10.1109/ACCESS.2021.3119291](https://doi.org/10.1109/ACCESS.2021.3119291).
- [42] Ethereum. *Proof-of-Work (PoW)*. Accessed: Apr. 26, 2022. [Online]. Available: <https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/>
- [43] *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*, document RFC 4492, Network Working Group. Accessed: Apr. 27, 2022. [Online]. Available: <https://tools.ietf.org/search/rfc4492>
- [44] *Ethers.js Library Official Documentation Website, Hashing Algorithms*. Accessed: Apr. 26, 2022. [Online]. Available: <https://docs.ethers.io/v5/api/utis/hashing/>
- [45] Microsoft Azure. *Microsoft Azure: Cloud Computing Services*. Accessed: Aug. 30, 2020. [Online]. Available: <https://azure.microsoft.com/en-us/>
- [46] Ethereum. *What is Ethereum? The Foundation for Our Digital Future*. Accessed: Aug. 30, 2020. [Online]. Available: <https://ethereum.org/en/what-is-ethereum/>
- [47] Geth. *Go Ethereum: Official Go implementation of the Ethereum Protocol*. Accessed: Aug. 30, 2020. [Online]. Available: <https://geth.ethereum.org/>
- [48] Puppeth. *Puppeth: Ethereum Private Network Manager (Secondary Repos)*. Accessed: Aug. 30, 2020. [Online]. Available: <https://github.com/puppeth>
- [49] Solidity. *Solidity—Solidity 0.8.7 Documentation*. Accessed: Aug. 30, 2020. [Online]. Available: <https://docs.soliditylang.org/en/v0.8.7/>
- [50] ReactJs. *React: A JavaScript Library for Building User Interfaces*. Accessed: Aug. 30, 2020. [Online]. Available: <https://reactjs.org/>
- [51] NodeJs. *Node.js is a JavaScript Runtime Built on Chrome's V8 JavaScript Engine*. Accessed: Aug. 30, 2020. [Online]. Available: <https://nodejs.org/en/>
- [52] J. Mackie. *What's a Legally Binding Agreement*. Accessed: Apr. 26, 2022. [Online]. Available: <https://www.termsfeed.com/blog/legally-binding-agreement/>
- [53] Etherscan. *Ethereum Full Node Sync (Default) Chart*. Accessed: Jan. 24, 2022. [Online]. Available: <https://etherscan.io/chartsync/chaindefault>
- [54] P. M. Royo, J. Rodríguez-Molina, J. Garbajosa, and P. Castillejo, "Towards blockchain-based Internet of Things systems for energy smart contracts with constrained hardware devices and cloud infrastructure," *IEEE Access*, vol. 9, pp. 77742–77757, 2021, doi: [10.1109/ACCESS.2021.3081932](https://doi.org/10.1109/ACCESS.2021.3081932).
- [55] HSI Sensing. *Sentinel Retro—PRX+12215 Intrusion Detection and Prevention Solutions*. Accessed: Jan. 20, 2022. [Online]. Available: <https://hsisensing.com/sentinel/>
- [56] Bravo Controls. *Tamper Proof Room Temperature Sensor*. Accessed: Jan. 20, 2022. [Online]. Available: <https://bravocontrols.com/shop/tamper-proof-room-temperature-sensor/>
- [57] Fastenright. *What is a Tamper Proof Screw?*. Accessed: Jan. 20, 2022. [Online]. Available: <https://www.fastenright.com/blog/what-is-a-tamper-proof-screw>
- [58] A. Baronchelli, "Collective intelligence and the blockchain: Technology, communities and social experiments," 2021, *arXiv:2107.05527*. Accessed: Jan. 21, 2022.
- [59] *Sustainability-Aware IoT Systems Driven by Social Communities (StoT-Com)*. Accessed: Jan. 21, 2022. [Online]. Available: [https://investigacion.ugr.es/sites/vic/investigacion/public/documentos/proyectos/ministerio/2020/resoluciones/provisional/PRP\\_PID2020.pdf](https://investigacion.ugr.es/sites/vic/investigacion/public/documentos/proyectos/ministerio/2020/resoluciones/provisional/PRP_PID2020.pdf)
- [60] *Kubuka-Más Por Ellos Website*. Accessed: Aug. 30, 2020. [Online]. Available: <https://kubuka.org/>



**MANUEL PEREIRA LAMELA** received the bachelor's degree (Hons.) from the Technical University of Madrid, in July 2021, with the thesis titled "Deployment of Blockchain Solution in Cloud Facilities Oriented to the Achievement of Sustainable Development Goals." He is currently a Telematics Engineer specialized and experienced in distributed systems, cybersecurity, and distributed ledger technologies. His main research interests include cybersecurity, blockchain, distributed systems, sustainable development goals, and cyber-physical systems for social good.



**JESÚS RODRÍGUEZ-MOLINA** received the Ph.D. degree (Hons.), in 2017, with the thesis titled “Contribution to the Design, Implementation and Standardization of Semantic Middleware Architectures for the Smart Grid.” He is currently an Assistant Professor at the Technical University of Madrid. He has performed research activities at ETH Zürich, Switzerland; SINTEF, Norway; NREL, CU Boulder, Colorado; and TU Wien, Vienna. His research interests include distributed

and cyber-physical systems, blockchain, autonomous vehicles, the smart grid, and middleware. He has recently started getting familiar with deep learning.



**JUAN GARBAJOSA** (Senior Member, IEEE) joined the Universidad Politécnica de Madrid (Technical University of Madrid, UPM), Spain, as a full-time Professor, in 1997, where he is currently a Full Professor at the Computer Systems School (ETS de Sistemas Informáticos). He is the Deputy Vice-Rector for quality systems and competitiveness. Before that, he has spent 16 years in industry and government in different engineering and management positions, ten of which at a

start-up at the time, now a large multinational company. His current research interests include cyber physical systems architecture, agile and innovation, and more recently collective intelligence.

...



**MARGARITA MARTÍNEZ-NÚÑEZ** is currently a Professor at the Department of Organization Engineering, Business Administration and Statistics, Technical University of Madrid. She has participated in several Journal Citation Reports publication with high impact and co-leads the European-level initiative for higher education called European Engineering Learning Innovation and Science Alliance (EELISA). Her research interests include digital transformation, smart grid,

business and management, business models applied to sustainable development goals, and environmental economics.