IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 June 2024, accepted 31 July 2024, date of publication 6 August 2024, date of current version 16 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3439412

RESEARCH ARTICLE

Smart Contracts for Managing the Agricultural Supply Chain: A Practical Case Study

EMAN-YASER DARAGHMI[1], SHADIA JAYOUSI[2], YOUSEF-AWWAD DARAGHMI[3,4], RAED S. M. DARAGHMA[4], AND HACÈNE FOUCHAL[5]

[1]Department of Computer Science, Palestine Technical University—Kadoorie, Tulkarm 9993400, Palestine
[2]Faculty of Graduate Studies, Palestine Technical University—Kadoorie, Tulkarm 9993400, Palestine
[3]Department of Computer Systems Engineering, Palestine Technical University—Kadoorie, Tulkarm 9993400, Palestine
[4]Department of Communication Engineering, Palestine Technical University—Kadoorie, Tulkarm 9993400, Palestine
[5]Lab-I*, Université de Reims Champagne-Ardenne, 51100 Reims, France

Corresponding author: Eman-Yaser Daraghmi (e.daraghmi@ptuk.edu.ps)

This work was supported in part by Palestine Technical University—Kadoorie and Al Maqdisi Program coordinated by the French Ministry for Europe and Foreign Affairs (MEAE) and the French Ministry for Higher Education, Research and Innovation (MESRI); and in part by the Consulate General of France in Jerusalem.

ABSTRACT The agriculture sector stands as one of the most significant sectors sustaining 70 percent of the world's population. In this sector, the supply consists of a series of interconnected stages, spanning from farming through production to the final delivery of goods to the end customer. A lack of transparency within the supply chain presents the largest gap between suppliers and retailers, such as not ensuring the true value of products or services. This research introduces AgroChain, a Blockchain-based system that is designed to support the Agricultural Supply Chain (ASC) process. For scalability purposes, the proposed AgroChain solution comply with a process model that separates the registry of agricultural records from the record itself. The development of the AgroChain prototype along with its smart contracts that establish a transparent, yet secure, environment within the ASC under Quorum which is Ethereum oriented network is illustrated in this research. This research demonstrates that Blockchain networks offer a solution to manage the ASC ensuring traceability, privacy and integrity. Moreover, the research indicates that there is a pressing need to promote the standardization of ASC smart contracts, incorporating secure and straightforward process. Smart contracts have to ideally be implemented in consortium environments, enabling reliable validation of transactions by independent third parties without necessitating access to their content. Moreover, investigating the accountability for illegal activities become challenging.

INDEX TERMS Agriculture supply chain, blockchain, smart contract, traceability.

I. INTRODUCTION

Recently, numerous governments worldwide have placed emphasis on developing sustainable agriculture in order to improve the quality of the farmers' life and sustainability raising their earning. There is no doubt that the agricultural sector is considered as one of the largest sectors affecting the economy. Based on data compiled by the Center for Economic Policy Research (CEPR), the agricultural sector engages 13.4 percent of the population in official employment, while informally involving approximately 90 percent of the workforce. However, farmers face a myriad of challenges, ranging from unpredictable weather conditions like droughts and frosts to market volatility and price fluctuations.

Regarding the agricultural sector, the Agricultural Supply Chain (ASC) consists of a series of interconnected stages, spanning from farming through production to the final delivery of goods to the end customer. The absence of communication and information tools among stakeholders in the ASC along with a lack of transparency presents the largest gap between suppliers and retailers, such as ensuring the true value of products or services and hindering effective coordination and collaboration. Traceability systems are digital techniques and algorithms widely used to monitor trace,

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

and manage the production, distribution and sales of products including agricultural crops. The crops go through what called ASC which is provides a simple, useful and abstract way of understanding how the crops produced, bought and sold. ASC traceability systems can ensure the quality of the chain this would enhance the reputation of exporters and the capacity to access new higher-value markets. These systems allow the ministry of national economy to monitor the prices of products to prevent fraud, embezzlement and prices manipulation. Various strategies have been employed to address the aforementioned challenges. Every approach carries its unique array of pros and cons, designed to meet various objectives within the domain of sustainable agriculture. For instance, the techniques outlined by Bai et al. [1] are geared towards enhancing transparency within the supply chain, whereas the methodologies discussed by Wang and Saurabh aim to bolster sales volumes and ensure better trade compliance.

In general, the farmer is responsible for harvesting the crops. The crops will be sold to the traders via ''wholesale and retail markets'' or what called ''commission markets'' which manage the trading process via auction. Commission markets assume the responsibility of generating comprehensive records containing all pertinent data related to the crops that have been sold. These records are subsequently stored within the databases of these commission markets, serving diverse purposes. These markets bear the duty of overseeing the ongoing maintenance and management of these records. Other individuals and entities also play pivotal roles in supporting the continuity of this chain. Financial institutions extend loans, governments establish regulations and policies, and agricultural research bodies devise strategies to enhance farmers' effective involvement in value chains. Additionally, radio stations, television channels, and media outlets play a significant supporting role by disseminating crucial information to farmers regarding market prices. Farmers possess the option of vending their crops through various commission markets, with corresponding records being stored within distinct databases. Consequently, farmers, who are armed with access privileges, must navigate multiple sources (commission markets) to retrieve their respective records. In a similar fashion, entities endowed with access rights can request records of a particular farmer from different providers when deemed necessary. This intricate scenario contributes to a deficiency in data coordination. Simply put, records are fragmented and isolated, lacking cohesion. This fragmentation and isolation predicament engenders a lack of oversight and control, thereby fostering disparities and price differentials. The requirement for multiple avenues to access records has accentuated challenges related to interoperability among stakeholders, further erecting barriers to seamless data sharing. Furthermore, the relentless evolution of technology has spawned sophisticated methodologies designed to breach digital privacy and security. Tragically, records emerge as prime targets for activities such as information theft, fraud, and manipulation of prices, given that they encompass confidential and sensitive information.

Blockchain technology has emerged as a versatile solution for a myriad of applications aimed at safeguarding and preserving data integrity through its chain structure. Initially rooted in cryptocurrency, blockchain has evolved into a Distributed Transaction Ledger (DTL), characterized by its distributed and digital ledger. This evolution has positioned blockchain as a transformative technology with far-reaching implications for enterprise applications in the foreseeable future. Utilizing Blockchain technology holds the potential to enhance the ASC and streamline the data management life cycle. The adoption of blockchain networks offers numerous benefits, including decentralization, fault tolerance, transparency and independent verification. A node which is a current copy of the entire chain is maintained by participants to ensure resilience and redundancy. In the event of a node failure, the network remains autonomous and sustainable as other nodes seamlessly continue their tasks. Decentralization is a fundamental feature of blockchain networks, where writing and reading in the chain are distributed among participants, eliminating the need for centralized control. This decentralized nature guarantees that no single participant can exert complete control over the chain of blocks. Moreover, participants are able to verify the integrity of transactions without the involvement of intermediaries, enhancing trust and transparency within the network. Furthermore, Blockchain networks offer robust security controls for data access and traceability, providing a secure environment for sensitive information. With these inherent features, Blockchain technology presents a promising avenue for improving the efficiency, security, and transparency of data management processes in various domains, including agriculture.

This research presents a Blockchain-based system, namely AgroChain, that is developed to demonstrate the developed smart contracts within a distributed architecture; thus, supporting the existing ASC. The specific objectives of this research are summarized as follows:

- We propose the deign principles for three technological architectures (i.e., centralized, distributed and multi-Blockchain) that are designed for the AgroChain system along with the ASC processes.
- We develop the AgroChain prototype that manages the ASC process. The developed prototype is built upon the designed distributed architecture following the ASC process. AgroChain satisfies the design principles' requirements. Additionally, the proposed AgroChain solution is consistent with common Blockchain programming languages.
- We employ the Zero-KnowledgeProof (ZKP) protocol [2] to implement smart contracts under Quorum for the proposed AgroChain prototype. The ZKP protocol enables consensus on blocks and transactions while ensuring data privacy. Consequently, the Blockchain network is perceived as trustworthy by entities, as their data persist protected during transmission without being accessed by third parties.

The Blockchain technology is outlined in Section II, along with its integration with the ASC and a review of existing literature relevant to the topic. Section III introduces our proposed process model for the ASC. In Section IV, we detail our technological solution, which encompasses various architectures for the processes of the ASC: centralized, distributed, and multi-blockchain. The implemented case study with smart contracts over Quorum is detailed in Section V. Section VI delves into the limitations of our study and suggests future directions. Lastly, conclusion and future work are presented in Section VII.

II. LITERATURE REVIEW

A. BACKGROUND

1) BLOCKCHAIN EVOLUTION

The evolution of Blockchain technology spans from its inception in cryptocurrencies to its application across various sectors, including real estate, pharmaceuticals, supply chains, energy, finance, the Internet of Things (IoT), academia, healthcare, digital identity, government, arts, media, justice, and insurance [3], [4], [5], [6]. The introduction of Bitcoin in 2008 by an anonymous individual under the pseudonym Satoshi Nakamoto marked the genesis of blockchain as a decentralized payment system, featuring its native token, Bitcoin. Initially, Bitcoin's blockchain operated on a Proof of Work (PoW) consensus mechanism, establishing a public and global network primarily dedicated to cryptocurrency transactions [7]. However, its structure lacked programmable capabilities, limiting its utility for broader applications.

The emergence of Ethereum, conceptualized by Vitalik Buterin, revolutionized blockchain technology by introducing smart contracts—a concept enabling programmable functionalities within blockchain networks [8]. Ethereum was envisioned by Buterin as a platform for "Programmable Money," with a Turing-complete language integrated into its scripting system to facilitate enhanced protocols. Gavin Wood significantly contributed to Ethereum's development, transforming it into a versatile computing platform capable of executing smart contracts through the Ethereum Virtual Machine (EVM).

Transactions altering the network state on Ethereum consume GAS, a unit denominated in Ether, Ethereum's native currency, which regulates resource consumption by contracts. While Ethereum offers faster transaction processing compared to Bitcoin, the deployment and invocation of smart contracts can incur substantial costs [9], impacting their feasibility within the cryptocurrency market.

Despite cost considerations, smart contracts offer numerous advantages such as reducing administrative overheads, enhancing business process efficiency, and mitigating privacy risks [10]. They present opportunities across various domains including public sector, sharing economy, distributed security/privacy systems {Citation}, IoT, data provenance and finance. Efforts are underway to optimize smart contract execution and minimize transaction costs, with studies exploring tradable permit schemes to improve information relevance, correct asymmetries among actors, reduce intermediary involvement, and enhance trading quality [11].

However, challenges persist in smart contract development, particularly concerning code security, debugging, and performance optimization [12], [13]. The challenges are further compounded by the absence of standardization, community support, example codes, and best practices. Initiatives as Alastria [14], the Spanish Blockchain consortium, seek to address these deficiencies by promoting collaboration, knowledge sharing, and standardization efforts. Our technological prototype, detailed in Section IV, tackles these challenges through secure, periodic audit procedures, standardized, authorization management, robust authentication, control logic implementation, and reliable, cost-effective warehousing solutions. By leveraging Quorum, a business-oriented version of Ethereum that provides privacy depending on the Zero-Knowledge Proof (ZKP) protocol, our solution ensures secure, permissioned transactions and the deployment of smart contracts tailored for enterprise environments.

2) BLOCKCHAIN INTEGRATION WITH THE AGRICULTURAL SUPPLY CHAIN (ASC)

The ASC [15] represents a comprehensive series of activities and stakeholders contributing collectively to the production, processing, distribution, and consumption of agricultural products. This intricate chain encompasses all stages of agricultural production, from initial crop cultivation and livestock raising to the availability of finished goods in the market. Various stakeholders, including farmers, suppliers, processors, distributors, retailers, and consumers, add value at each stage of the ASC. Ensuring efficiency, quality, and sustainability requires effective management throughout the agricultural product lifecycle, achievable through centralized or distributed approaches.

The integration of blockchain technology with the ASC [1] offers transformative enhancements in transparency, traceability, and efficiency. This complex sequence of activities, from agricultural production to product consumption, involves numerous stakeholders and processes. The distributed strategy of blockchain, which links nodes without a central control node, underpins its potential application within the ASC. Beyond its recognition in cryptocurrencies like Bitcoin, blockchain's utility extends beyond finance. Operating as an open ledger, it meticulously records transactions, ensuring participants remain interconnected through its unique peer-to-peer (P2P) distributed database communication protocol, which enables secure storage, verification, and auditing of transactions.

Blockchain's inherent resilience renders transactions tamper-resistant once incorporated into the chain, ensuring transaction security and data integrity. Utilizing blockchain for transaction validation introduces robustness and security, maintained through consensus processes. In agriculture, where stakeholders include farmers, suppliers,

processors, distributors, retailers, and consumers, incorporating blockchain can enhance transparency and accountability across the value chain. Advanced tools like sensors, drones, and artificial intelligence optimize productivity and minimize waste in precision agriculture. Addressing the complexity of the agricultural supply chain, blockchain technology offers a solution by enabling transparent and secure product tracking from farm to consumer. This blockchain-based traceability system empowers tamper-proof tracking, enhancing productivity and diminishing fraud risk. Blockchain's potential within the ASC extends beyond traceability, finding implementation in healthcare, education, finance, and government, bolstering transparency, integrity, provenance, and traceability. As agriculture aligns with technological advancements, integrating blockchain technology can usher in a new era of efficiency, trust, and sustainability It promises transparency and efficiency, yet challenges like scalability, interoperability, and governance require attention for seamless integration [3].

Furthermore, decentralized applications (dApps) are computer programs distributed throughout a blockchain or P2P network. dApps, often developed on the Ethereum (ETH) platform, serve various purposes, from gaming to social media. These applications enjoy the benefits of decentralization, including security, absence of censorship, and adaptability. However, they face potential setbacks in scalability, UI development, and code change complexities.

For instance, the simulated environment created by Shih [16], based on the Ethereum blockchain, demonstrates how the Blockchain can enhance the efficiency of producing and marketing organic vegetable. The blockchain transparently tracks the entire process involving farmers, distributors, and consumers, revealing its potential in agricultural optimization. Similarly, moroz highlights the potential of blockchain to enhance traceability and transparency in the food supply chain, as well as agricultural production and logistics efficiency. However, significant hurdles need addressing, including further R&D, industry-wide collaboration, and regulatory clarity.

3) THE POTENTIAL OF SMART CONTRACTS IN AGRICULTURAL SUPPLY CHAIN (ASC)

Within the domain of the Blockchain technology, a crucial advancement has emerged known as ''Smart Contracts'' [9]. These contracts bring a level of automation to transactions by executing predefined actions when specific conditions are met. In the Ethereum Blockchain ecosystem, an application housing such capabilities is termed a ''smart contract.'' It comprises a set of functions and associated data that reside at a designated address on the Ethereum blockchain. These ''smart contracts'' essentially serve as digital transaction executors, functioning as computer programs designed to autonomously fulfill contractual terms or agreements. What distinguishes them is their ability to function without the need for a central authority, legal framework, or external enforcer. By enabling trustworthy transactions and agreements among unrelated and anonymous parties, smart contracts revolutionize the landscape of transactional interactions. The roots of the smart contract concept trace back to 1997 when Nick Szabo introduced the notion as a way to digitally formalize and secure relationships within a network. In essence, a smart contract is an application operative within a blockchain network, executed by all participants within that network [15]. These codes regulate the transactions of Blockchain and meticulously outline the terms and conditions of agreed-upon contracts.

The underlying idea of smart contracts is to imbue software or hardware with contractual clauses, collateral requirements, bonding, and property rights. This approach significantly diminishes the likelihood of malicious breaches of contracts. Essentially, smart contracts are a mechanism that embeds contract-related functions directly into the blockchain network. They function autonomously, executing according to predefined rules without human intervention.

Presently, various blockchain-based projects have effectively integrated smart contracts, with platforms like Ethereum and Hyperledger leading the way. These smart contracts enable secure agreements and transactions between distinct, anonymous entities without necessitating the presence of a central authority or external enforcement. The Ethereum platform, for instance, empowers the creation of tailor-made smart contracts to suit the specific needs of desired systems.

The potential of smart contracts in Agricultural Value Chains (ASC) [17] holds immense promise for transforming the agricultural industry. Smart contracts can revolutionize how transactions occur within the complex web of interactions across agricultural value chains. By automating and digitizing contractual agreements between various stakeholders such as farmers, suppliers, distributors, and retailers, smart contracts can enhance transparency, efficiency, and trust throughout the value chain. These contracts can be programmed to execute specific actions automatically when predetermined conditions are met, such as triggering payments upon delivery of goods, ensuring timely and accurate compensation for farmers. Additionally, smart contracts can facilitate traceability by recording each step of the value chain on the Blockchain, providing consumers with accurate information about the origin, quality, and journey of agricultural products. This level of transparency not only promotes food safety and quality but also strengthens the relationships between stakeholders by reducing disputes and misunderstandings [1], [17], [18], [19], [20], [21], [22].

B. RELATED WORK

As the world experiences advancements in both social and technological spheres, information technology is increasingly being adopted to aid the development of farming practices. This has led to the establishment of expansive agricultural production and construction databases. However, in this evolving landscape, concerns regarding security and

effectiveness have come to the forefront. Recognizing these challenges, the integration of blockchain technology with the vast data sets of big data has emerged as a solution, enhancing the legitimacy of data information while simultaneously addressing issues related to data information abuse [23].

In the domain of agricultural expansion and development, substantial challenges arise from high costs and maintenance expenditures. Furthermore, there are issues with the efficacy of certain agricultural maintenance processes. To tackle these challenges, the integration of blockchain technology with the Internet of Things (IoT) has been explored. As the number of operating systems within IoT systems grows, centralized administration becomes impractical. Here, the collaborative effort of blockchain and IoT development holds promise. By facilitating automatic administration of agricultural equipment maintenance, these technologies can significantly reduce farm maintenance costs and facilitating the growth of large-scale unified crop networks. This development is further fueled by the ongoing evolution of internet technology and its profound influence on database structures [23].

Present-day supply chains are grappling with a multitude of issues related to data reliability. These encompass concerns about customer trust, supply chain transparency, product quality, logistical challenges, environmental impact, consumer data security, fraud, and food safety, among others. As the focus on food safety and sustainability intensifies, consumers are increasingly demanding greater transparency within agrifood supply chains. However, the complex and intricate nature of existing agrifood chains has resulted in a disconnection between consumers and producers. This gap prevents consumers from directly engaging with growers to address their concerns. This scenario underscores the need for transparency and trust within the supply chain. Simultaneously, the proliferation of various food and beverage products accompanied by diverse certification schemes has heightened the risk of fraud and adulteration. Currently, compliance data is often audited by trusted third parties and stored in centralized databases or paper records, leading to a range of data-related issues that hinder transparency and trust. Consequently, this compromises food safety, quality, and sustainability within the agrifood chain [18]. Blockchain offers a promising solution to address these challenges of transparency and trust within the supply chain. By ensuring the immutability of data, blockchain offers the potential to foster data sharing among the different stakeholders within a food value chain. This transformative capability has the power to usher in a new era of transparency and trust within complex supply chains [24].

Blockchain's utility extends across various domains, with its integration into agriculture showing significant potential. The concept of traceability, particularly in ensuring food safety within the food supply chain, is regarded as a viable mechanism. Efforts to achieve this objective have employed a range of strategies, including digitization and RFID tagging. These efforts are validated through real case studies conducted by prominent organizations like IBM, Walmart, and Tsinghua University. Notably, IBM's partnership with Walmart to trace the journey of mangoes from farm to consumer and Provenance's use of blockchain and smart tags to track tuna exemplifies the practical application of blockchain technology to enhance traceability and transparency within the agrifood supply chain [25].

Researchers [21] have further delved into the fusion of blockchain and RFID technology within the agricultural sector, elucidating their strengths and weaknesses. Such explorations have resulted in proposed frameworks and real-time traceability mechanisms based on blockchain, the Internet of Things (IoT), and HACCP (Hazard Analysis and Critical Control Points). These endeavors aim to infuse openness, transparency, security, neutrality, and trust into the agricultural and food supply chain [17], [21].

Malik et al. [20] have introduced the ''ProductChain'' blockchain framework as a means to track the origin of food products securely. Similarly, various researchers have delved into the integration of blockchain and IoT for enhanced traceability. This approach leverages IoT devices to collect and encode data, which is then fed into blockchains to establish provenance. Noteworthy examples include the development of blockchain-based systems like AgriBlockIoT, which contribute to improving traceability within agricultural and food supply chains [19], [20], [26], [27]. Meddeb's work introduces a blockchain-based traceability system for agri-food supply chains, leveraging transaction-based accounting to address the limitations of traditional ledger-based systems. This solution underscores the importance of end-to-end traceability and emphasizes transparency and trust within the agri-food sector [28]. The concept of Community Supported Agriculture (CSA) models, integrated with blockchain, further amplifies the potential of blockchain technology within the agri-food sector. These models not only address the origins of agricultural products but also distribute market risks between producers and consumers. Similarly, proposed models like the BPCM (Blockchain based Consumer-Producer) present innovative ways to counter challenges such as Sybil attacks and intermediary involvement, ultimately benefiting farmers and consumers alike [29], [30].

The integration of blockchain with the Internet of Things (IoT) has also resulted in the development of agricultural blockchain-based IoT systems, aimed at facilitating the selling and purchasing of agricultural products. This integration allows for real-time access to vital data, including soil moisture, climate data, payments, sale and demand prices, and more. Ultimately, the combination of these technologies presents opportunities to enhance transparency and efficiency within the agriculture sector [31], [32].

Nguyen et al. [29] proposed a Blockchain-based framework for developing a digital traceability solution as a transparent and reliable communication channel between actors in an agricultural value chain in order to build

sustainable agriculture in Vietnam. He contributes to propose an enterprise blockchain platform for developing a traceability software solution. This means that there will be no connection to cryptocurrency, which means that it will not be constrained by legal constraints in Vietnam. His experimental results showed that enterprise Blockchain platforms have properties that are suitable for deploying Blockchain-based applications in the agricultural sector. Blockchain is presented as a means of increasing transparency and traceability, lowering costs, and increasing efficiency. However, there are still issues with scalability, interoperability, and governance. Transparency and traceability can be improved, costs reduced, and time delays eliminated with the help of blockchain technology, while the quality of food products is ensured and fraud is protected against [29].

III. AGROCHAIN PROCESS MODEL

This section introduces a Blockchain-based system, namely AgroChain that is designed and developed to support the Agricultural Supply Chain (ASC) with the aim of preventing fraud and price manipulation while simultaneously providing privacy, interoperability, security, efficient access by all stakeholders to the agricultural records and ensuring the integrity of the agriculture process.

Many limitations had been found in previous literature when accessing digital information during the ASC process, including: duplication, alteration or modification in addition to not able to maintain data integrity. Permissions and roles related to digital records may also be a challenging task. Furthermore, the digital storage may not remain secure, confidential or sensitive over time. Digital information may subject to various regulations imposed by different countries. Therefore, the process of the ASC for digital storage is more complex than traditional one. Once examining the existing literature, the following process is proposed aiming to establish a generalized framework adoptable to various regions of the world.

A. THE AGRICULTURAL SUPPLY CHAIN PROCESS

In the ASC process, each participant has a specific role crucial to the agricultural supply chain. The Farmer role is dedicated to those responsible for cultivating, processing, and packaging agricultural items. Farmers initiate the process by providing essential details such as farm name, location, and product notes, resulting in the creation of a new item with a unique Universal Product Code (UPC), subsequently classified as 'Harvested'. Following this, farmers participate in auctions to sell their harvested agricultural products, selecting the best bid offered. The Distributor role pertains to entities seeking to purchase agricultural products through successful bids. Distributors play a vital role in procuring and shipping items to their destinations. Upon receiving the products, Retailers make them available for purchase to consumers, thereby completing the agricultural supply chain.

Moreover, the rest of the participants are government employees or officials who are responsible for overseeing and regulating prices of agricultural products within the supply chain. Incorporating that monitor role entails the quality and safety of agricultural products through various quality control measures and ensures governmental oversight of pricing mechanisms, fostering transparency and fairness in agricultural transactions while safeguarding the interests of all stakeholders involved in the supply chain.

Additionally, for the ASC process, a primary participant plays a crucial role for the smart contracts' deployment and maintenance. Consequently, the owner of the digital record will be able to authorize the participants' access. Figure 1 shows the phases of the proposed AroChain with the ASC process model:

- Cultivation: The ASC process begins with the cultivation phase, where farmers initiate the cultivation and production of agricultural products. Once harvested, product details must be recorded in the ASC.
- Distribution: This phase involves the distribution of agricultural products to various intermediaries, such as wholesalers, retailers, and export partners. Participants involved in distribution are responsible for managing product transportation and logistics.
- Inspection: The inspection phase entails assessing the quality and safety of agricultural products through various quality control measures, such as sampling, testing, and certification.
- Sale: In the sale phase, agricultural products are made available to consumers through retail outlets, farmers' markets, online platforms, or other distribution channels.
- Consumption: Finally, the consumption phase involves consumers purchasing and consuming agricultural products, completing the cycle of the ASC process.
- Record destruction: Upon requesting the destruction of the record, the destruction method will be invoked.

As shown in the Figure, AgroChain enables easy and error-proof tracking of farm products from the farmer to the consumer. Users of the AgroChain are allowed to fetch details of existing products and farms in addition to feed details of new Products and Farms. At the different stages of the chain, the concerned stakeholders can update and trace status of the shipment. Complete transaction history is logged and the transactions can be validated on Etherscan using the Transaction ID.

B. THE ASC SERVICES

The process of the ASC necessitates the definition of the services listed below for the e-agricultural data in order to ensure traceability:

- **Create the agricultural record:** The person who acquires the agricultural products signs the digital agricultural record providing all the circumstances associated with the acquisition process. Acquiring the agricultural products refers to the process of obtaining or procuring agricultural items such as crops, livestock, or other agricultural produce from the farm or source where they are grown or produced. This could involve

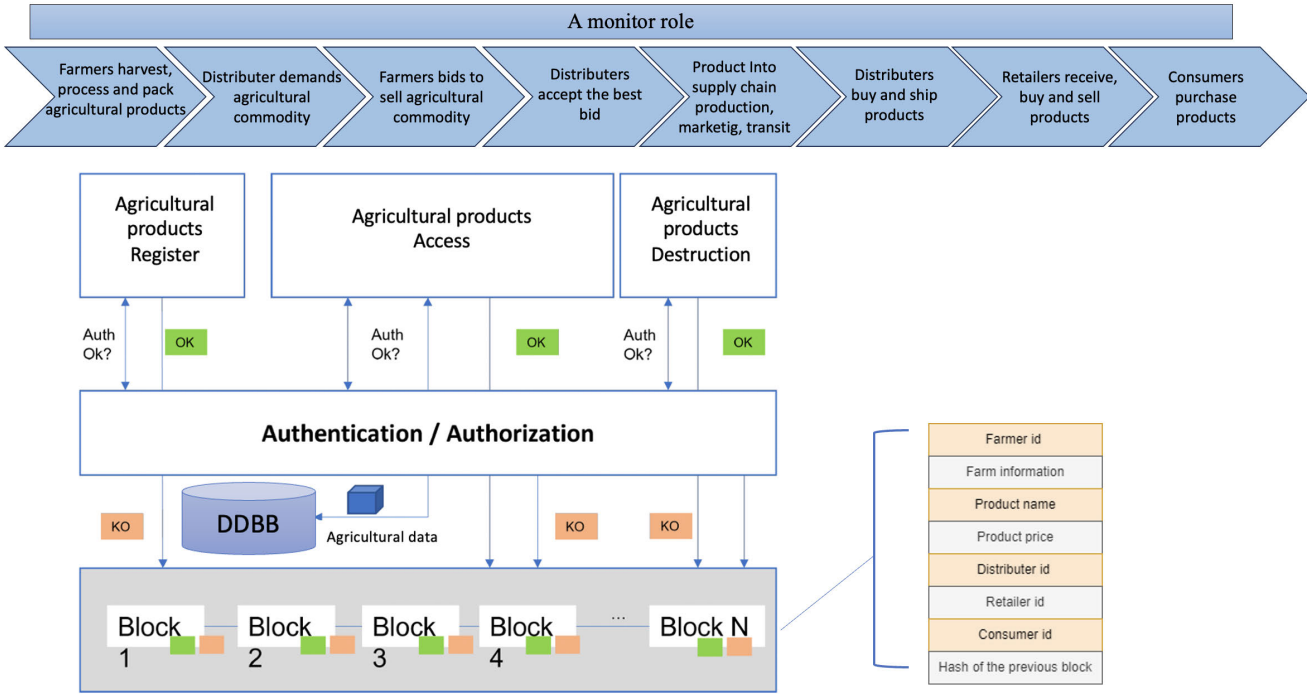

FIGURE 1. The proposed AgroChain system with ASC process.

activities such as harvesting, processing, and packing agricultural products before they are made available for sale or distribution in the supply chain. After that, the corresponding agricultural record within the Blockchain is created using these data meaning that the agricultural record must be created by the participant/user who acquires the record.

- **Invalidating the agricultural record**: Record the authorization details (who, when, how and why) for the destruction of the agricultural record, if necessary.
- **Obtaining the agricultural record**. Register all users requesting access to the agricultural record
- **Transferring the record**. The management of the agricultural record may be delegated when participants transfer that responsibility of the record to other competitors.
- **Displaying the agricultural record logs:** Report on all log entries related to a particular piece of agricultural record.

The person, who is responsible for the process of obtaining or procuring the agricultural items, such as crops, livestock, or other produce from the farm or the source where they are grown or produced, signs a record. The record includes details about where the product was obtained, when it was acquired, how they were obtained (i.e., harvested, processed, packed), and any other relevant information. If possible, the record is then stored in the compatible repository. During this process, the hash of records or their fingerprints are calculated in order to validate the records' integrity. AgroChain participants will seek to examine and review the events that have been occurred during the life of the record. Consequently, participants who have accessed the record must be registered, along with details such as where, when and how they accessed it, as well as the why they examined the record.

Data which can be logged upon access when someone accesses the record depend on the available information on the Blockchain network (i.e., IP address, account identifier, etc.). Sometimes, it may not be possible to gather certain information, such as connection source information, due to limitations or restrictions.

Upon determining the record's destruction by the competent authority, the entire circumstances have to be stored as well as invalidated in the record. This service has to be only consumed by accredited participants in a timed manner. Thus, it is important to verify whether the users of the system have the needed logs and permits, i.e., any failed attempts. All requirements outlined above have been integrated with the technological solution developed to manage the AgroChain system along with the ASC process.

IV. TECHNOLOGICAL SOLUTION

This section details the employed technological solution of the defined process model with considering the lack of homogeneity and maturity of the Blockchain in addition to the ongoing transition towards a cloud-based services by companies. Thus, ensuring the specific needs of the ASC processes will not be affected. When the hardware and the software requirements rely on the cloud services or the hybrid local-cloud infrastructure, the ability to rapidly evolve or scale

the architecture and applications will be supported by this technological solution.

Consequently, the proposed infrastructures will easily tolerate failure and remain up-to-date given that paying per consumption service will be implemented. These features are particularly prominent for the Blockchain, as rapid evolution is possible. Moreover, these changes are open to the improvements from companies, consortiums, the scientific community, etc.

A. DESIGN PRINCIPLES

Nowadays, employing a clean design principle along with a modular approach is mandatory to allow modules evolving separately without affecting the data recorded or even lost the emphasis of the ASC process. Following points show the adopted design principles for the proposed technological solution:

Smart contracts, which are detailed in the following section, are employed to provide the logic of the ASC. For reliability, a verified code like OpenZepellin library [33] must be employed as this library is developed to ensure creating safe Solidity smart contracts [8] that should be standardized and open-source to achieve the scrutiny and endorsement of the community, such as stakeholders, developers, and users who are involved in the technology and its application.

- Accessing the entire data related to the AgroChain and the ASC must only be performed via the developed smart contract. Although the Blockchain network architecture incorporates layers and interactions with contracts, the chain information should not be maintained or managed without smart contracts invoking. The architecture has to ensure both access facilitation and the independence of ASC third-parties validation, if necessary, granting them suitable permissions via the interfaces of the Blockchain.
- Smart contracts must be deployed in one or more Blockchain consortium, requiring technologically preparedness for such deployment.
- Consensus support with Zero-Knowledge Proof (ZKP): Independence from third parties must be ensured upon the participation of public or private consortiums. Licensed technology is crucial to preserve the confidentiality and the privacy without compromising the consensus of the network on the transaction's validity.
- The separation of the data custody from the agricultural record. Due to the data size, it impractical to utilize on-chain data (i.e., using the Blockchain as a confident repository). The hash of the data or its fingerprints are stored in the Blockchain registry to detect tampering, resulting architecture is more complex.
- Robust authentication and authorization methods. Ensuring maximum certainty regarding the correspondence between virtual and real identity is vital for reliability. Distributed architecture, which utilizes wallets for credential storage based on public key cryptography, guarantees custody of private keys to prevent currency loss. This mechanism suffices for authentication based on multiple factors
- Single authentication identifier: Private key ownership signifies identity. Validation of digital identity against real identity is necessary for the ASC process, necessitating defined processes for managing associated identities and private keys.
- Use of low-cost, reliable, and safe warehousing: Agricultural data must be securely stored for extended periods. While traditional solutions involve making encrypted copies stored by various entities, cloud storage offers security, reliability, and redundancy without infrastructure maintenance. However, frequent access may render it unsuitable.
- Electronic format following open standards: An open ASC format focuses on trace generation rather than imposing a standard format. Any electronic data with a fingerprint, digital digest, or footprint, such as SHA-512 or SHA-256, is acceptable. Backwards compatibility is ensured by covering MD5 and SHA-1. It is a requirement to use a minimum SHA-256-based footprint.
- External security audit compliance: The technological solution is resilient, robust and subjected to periodic audits by third parties to ensure compliance with relevant norms and regulations.

B. PROPOSED ARCHITECTURES

For implementing the AgroChain with the proposed ASC process, three architectures have been designed that ensure that it is not allowed to perform self-manipulation. The entire information related to the ASC recorded in the Blockchain network. Following points show the common elements among the proposed designed architecture.

- **Secure Database Storage (SDS):** Data protection and security is crucial, ensuring that only authorized and legitimate participants have access. Monitoring participants' activities as well as maintaining the activities logs in the registry are essential.
- **Identity Node:** Each participant in the Blockchain network has an identity node. Participants can deploy their own node privately to manage smart contracts, while other nodes validate transactions without accessing specific details.
- **Smart Contracts:** These contracts implement the management process of the AgroChain along with tracking all events within the system.
- **Blockchain Consortium:** Networks like Alastria or R3, where smart contracts are deployed to establish consensus and maintain an immutable chain of blocks, ensuring data integrity throughout the entire ASC process.
- **Identity Manager:** Responsible for managing authentication and authorization processes, ensuring that each participant has accredited access and permissions. Accounts and identity accreditations are tied to individual persons.

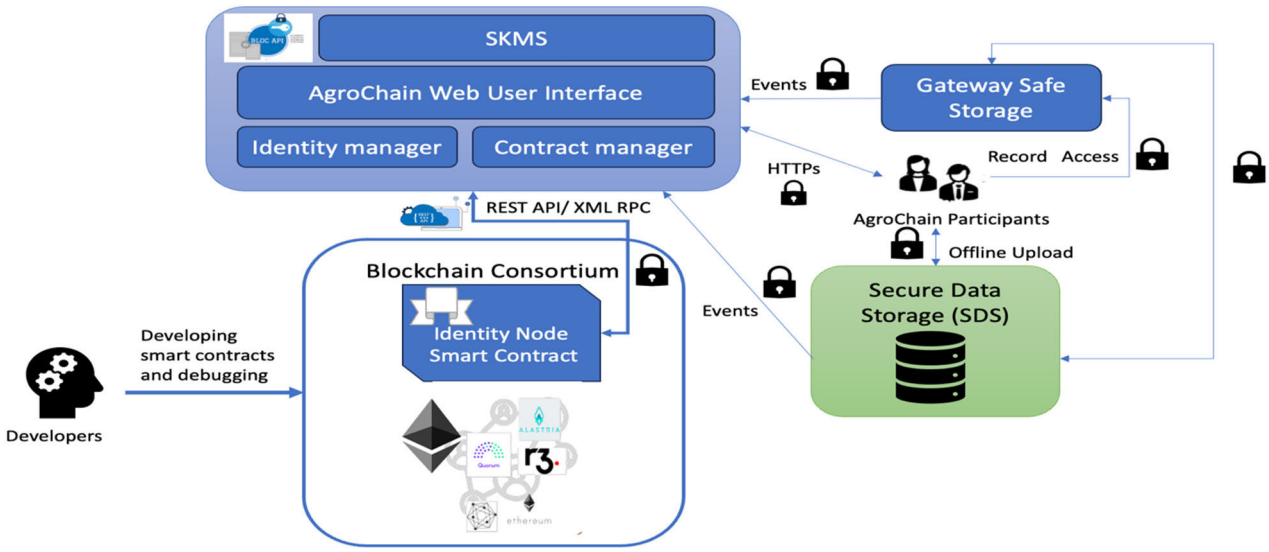

FIGURE 2. AgroChain centralized architecture.

- **AgroChain User Interface:** A web/mobile distributed or centralized application facilitating participant interaction with ASC processes.
- **Secure Storage Gateway:** An interface facilitating implementing data control as well as the downloading, uploading of files.
- **Smart Contracts Development Environment:** Enabling compiling, debugging, and deploying within the chosen Blockchain.
- **Secure Key Management System (SKMS):** Facilitates the establishment of secure communication channels among ASC participants and Identity Nodes. Leveraging certified hardware, this component guarantees secure key custody, encryption, and signature, effectively mitigating security risks linked to storing keys in configuration files or system memory.

Centralized, distributed or multi-blockchain centralized architecture could be selected by the service provider depending on his/her specific requirements (i.e., costs, performance, etc.).

As shown in figure 2, the centralized architecture is service-oriented indicating that the architecture is designed to prioritize the delivery of services. This typically involves organizing components and resources in such a way that they efficiently provide services to users or clients. In other words, the main focus of the architecture is to ensure that services are readily available and accessible to those who need them. The Blockchain access is facilitated via micro-services or a layer of services that interact with the Blockchain through delegation. Ensuring security mandates, the implementation of a robust authenticated identity management system for the Blockchain. Upon authentication, retrieval of the private key from a SKMS on the server side facilitates the signing of transactions. Furthermore, any authorized third-party account retains the capability to independently verify the ASC process.

Figure 3 shows the distributed architecture which is a Blockchain-based architecture that inherits from public networks. A wallet app stores the public-key pair in order to allow accessing the network. The key pair of a participant's account is stored in the network in order to verify the signed transactions from that wallet. Connection to the network is protocol-dependent, with XML-RPC being used in Ethereum. To integrate the network data with the distributed application, an interaction must be performed by the browser's client. For instance, tools such as the WEB3J library [34] and the Metamask extension [34] for Ethereum facilitate this process.

The Multi-blockchain centralized architecture abstracts the network technology used. This architecture is considered as a variant of the centralized architecture (see figure 4). The benefit lies in the redundancy of the custody chain across multiple networks. For instance, it serves as a solution to offer ASC services to third parties. This redundancy across various networks enhances reliability and validation to a greater extent.

As shown in the previous figures, the distributed architecture has additional components that are not presented in the centralized or multi-Blockchain architecture. Participants access the AgroChain application through a specific browser, connecting to their respective Identity Node via HTTPs. The application provides the necessary JavaScript to the browser for interaction with the Blockchain network client. Additionally, to facilitate the communication among participants and Identity Nodes, a wallet is needed. Accessing the wallet requires a JavaScript library and a browser extension. A combination of Metamask and WEB3J is common for the Ethereum network. The entity node provides an interface

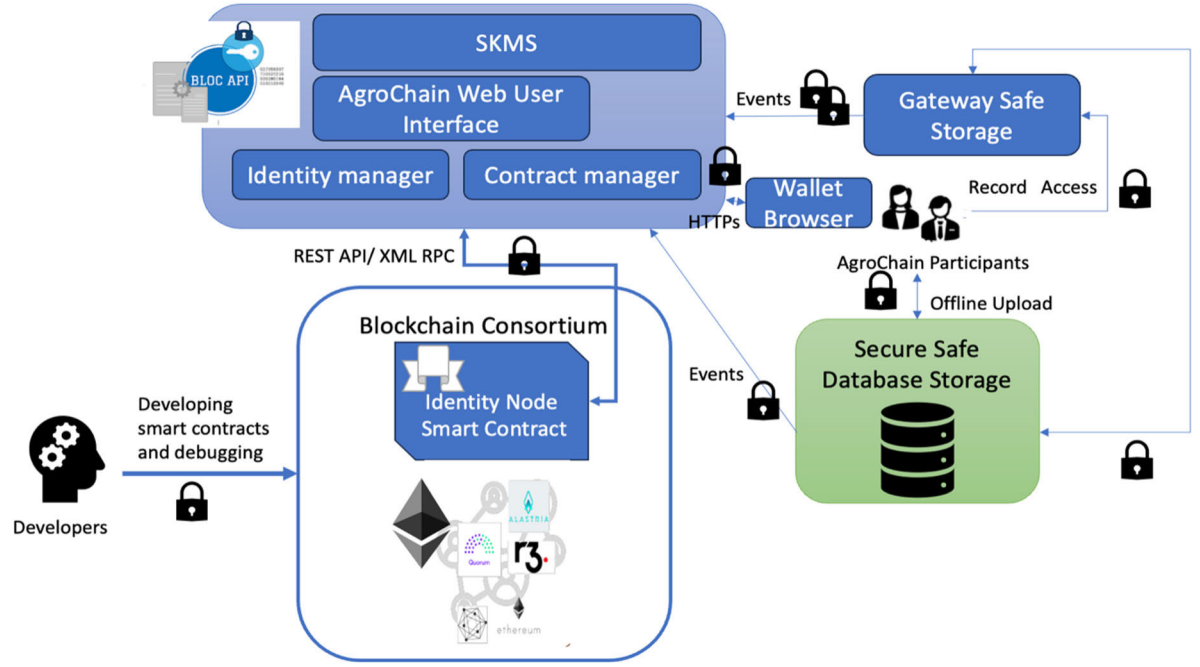

FIGURE 3. AgroChain distributed architecture.

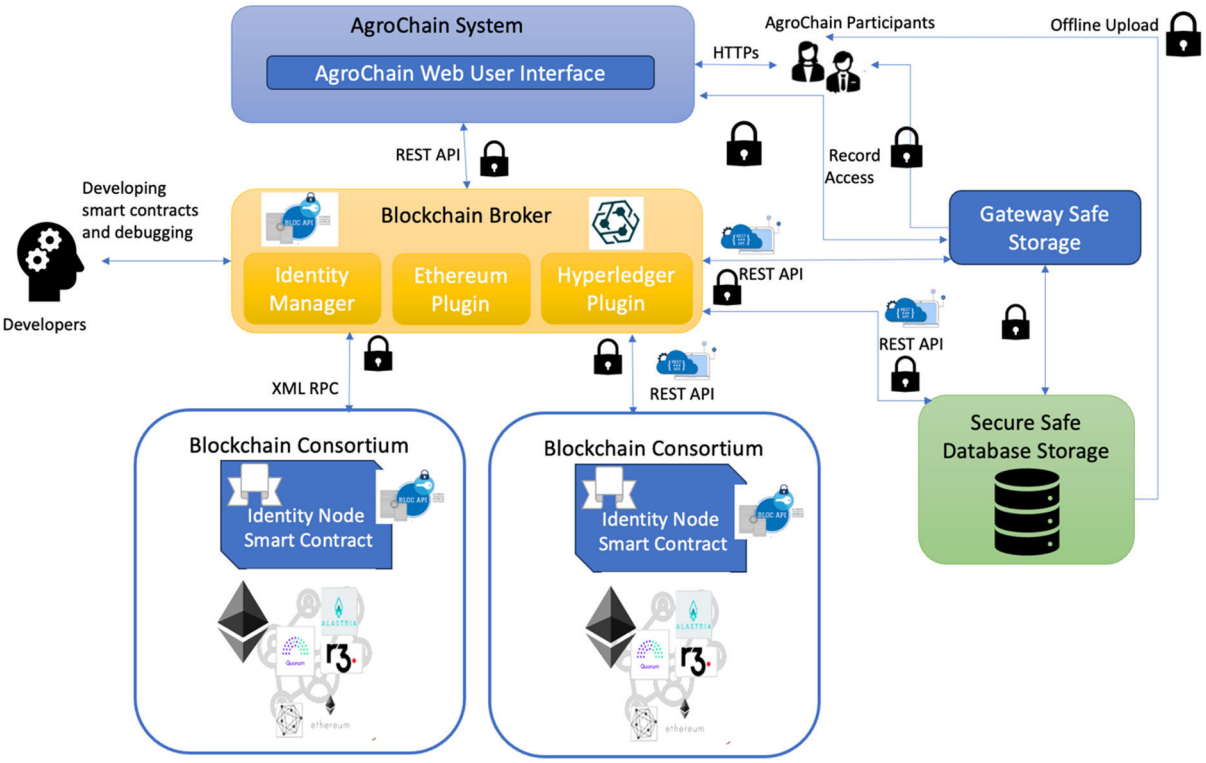

FIGURE 4. AgroChain multi-blockchain centralized architecture.

for external interaction. For Ethereum networks and Quorum specifically, communication is done via XML-RPC, while Hyperledger uses a REST-based API.

Furthermore, the secure database should be able to trigger events against the node for reporting data access. This can be achieved by implementing a Function as a Service (FaaS)

TABLE 1. Comparing the centralized and distributed AgroChain architecture.

Feature	Centralized AgroChain Architecture	Distributed AgroChain Architecture
Access Requirements	Application access does not require extra extensions.	Application access may require browser extensions and additional authentication mechanisms.
Identity Management	Identification, authentication, and authorization managed centrally.	Identity management may be decentralized, with each participant having their own identity node.
Interaction with Network	Through the user's network account, the application interacts with the network.	Via a wallet App storing public-key pair, the application may interact with the network
Cryptography Server	Requires a cryptography server for securely storing keys.	May not require a separate cryptography server if keys are managed locally on the user's device or wallet.
Secure safe storage Event Generation	Secure safe storage generates events directly against the application.	Secure safe storage may need to generate events against the smart contract, which in turn records the audit trail.
Multi-Blockchain Architecture	Applies to institutions desiring redundancy in blockchain and smart contracts.	Multitenancy may be necessary to serve multiple entities while maintaining isolation of information and access.
Connection Services Layer Separation	Connection services layer is integrated with the application.	Connection services layer is distinct from the application, along with smart contracts and identity management.
Multi-Tenant Infrastructure	Not necessary to support multiple entities on the same infrastructure.	Infrastructure must be multitenant to support multiple entities with isolated information and access.

using platforms like AWS Lambda. The FaaS reads event logs from the storage and then creates directly an audit log within the smart contract.

AgroChain is responsible for managing the uploading and downloading of data between the client's device and secure stores. It facilitates communication with various secure stores, eliminating the need for credentials within the secure store. Instead, access is granted through service accounts with precise permissions managed by the identity management module. When downloading, a secure Uniform Resource Identifier (URI) is generated to enable client-side access to the data. Moreover, a protocol is established for sending data to offline providers enabling them to perform data uploading or migration, preventing issues and cost overruns. Tools and environments are necessary for building and debugging smart contracts. For Ethereum networks, suitable options include the Remix and Truffle suite. Hyperledger provides projects and SDKs specifically tailored for this purpose. Table 1 summarizes the differences between the proposed centralized and distributed AgroChain architecture.

V. STUDY CASE: AGROCHAIN SMART CONTRACTS AND ETHEREUM FOR THE ASC PROCESS

A. AGROCHAIN DEVELOPED PROTOTYPE

The Proof of Concept (PoC) that is developed to explore the feasibility of completely implementing the proposed ASC process using smart contracts is illustrated in this section. The features of the prototype include: openness, security, and robustness. Three technological architectures, centralized, distributed, and multi-blockchain were presented. The distributed architecture that satisfies the proposed AgroChain design principles and the proposed ASC process model has been selected for the purpose of this PoC. The architecture of the proposed AgroChain that has been built under Quorum is presented in figure 3.

The proposed solution encompasses various functionalities including the creation of agricultural records, participant management, audit processes, and record invalidation, among others.

For secure storage of records, we use Google's Safe Storage service, known as Google Storage Coldline. The safe gateway component implements a REST interface using Google Storage API SDK and Node Express to facilitate the interaction with the safe storage (see Figure 5). For testing, a the official 7-node Quorum distribution test network and the Ganache implementation are utilized. For ensuring privacy and transaction authorization purposes, Quorum incorporates the ZKP protocol. A web application using the Web3j library and Angular is developed as a user interface. Connection to the Blockchain is performed via the Ethereum client to guarantee it will entirely be distributed. In our case, we have opted for Metamask with an Ethereum-compatible wallet. Solidity is adopted to implement the smart contracts using Remix as a development and debugging environment, which generates compiled bytecode for execution on EVMs. TruffleSuite [35] has been utilized for deploying contracts within the Blockchain.

B. AGROCHAIN COMPONENT INTERACTIONS

The interaction among the AgroChain components is detailed in Figure 5. The AgroChain client app is accessed via a

Metamask extension using any compatible browser, such as a Google Chrome and communicates with the AgroChain distributed application through HTTPs. AgroChain enables farmers, distributors, and other stakeholders to register within the system and trace the status of their products. Users can interact with AgroChain through a full node, which is a computer connected to the main chain of the Ethereum Blockchain, or through a service node that provides services to other nodes, such as a cloud service or a wallet. Both full nodes and service nodes are connected to Ethereum, allowing users to access the associated Decentralized Application (dApp). The JavaScript code, including the interface and style sheets, is downloaded from the Node server by the browser. However, direct interaction between the web server and the smart contract or the Blockchain network does not occur.

The client application directly interacts with the smart contract deployed on the network node. This interaction is facilitated through the Metamask extension and the Web3j JavaScript library in the client's browser.

Various REST services, including information uploading and downloading are provided by the Safe Storage. Interactions between the gateway and the application entails adjusting the uploading/downloading parameters, enabling the browser handling the information directly from the safe gateway. The safe gateway handles the collection and transmission of information to and from the safe storage. With the appropriate credentials, the gateway remotely stores files, providing an abstraction layer for the safe storage control without exposing the clients' credentials. Further enhancements on the gateway could provide more controls on the managed information, such as encrypting data or recalculating fingerprints.

In cases where the volume of electronic records is substantial (e.g., gigabytes, terabytes, or petabytes), the client may directly interact with the store via the use of native credentials and tools. Examples include Amazon AWS offline data migration options or Google's Gsutil tool.

Remix is utilized for developing and debugging smart contracts' tasks. It offers options for developing, compiling, debugging, and testing the smart contract, either by accessing an Ethereum virtual machine or through Web3j Metamask for interconnection with Ethereum. Additionally, the Truffle suite is employed for compiling and deploying the contracts, facilitating deployment on Ethereum-based Blockchain networks. This process also handles compiling and migrating the contract to the corresponding network.

As shown in the figure, following points summarize the AgroChain system dependencies.

- **Node Package Manager (NPM):** It is a command-line utility that serves as an online repository for the release of open-source Node.js projects. It also helps users install and manage their packages. The repository contains thousands of applications and libraries.

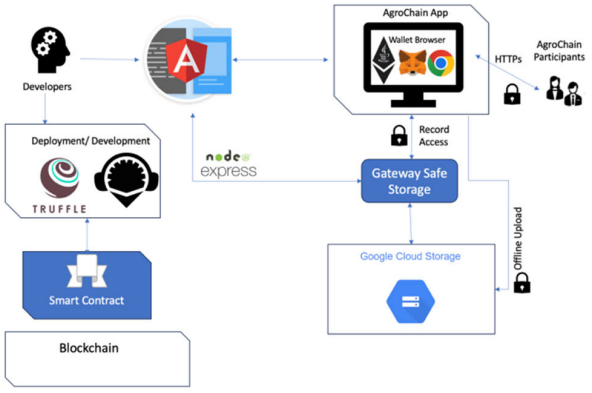

FIGURE 5. Architecture of the developed AgroChain prototype.

- **Truffle Framework:** The Truffle Framework allows the construction of decentralized apps on the Ethereum network. It includes a suite of tools that facilitate the utilization of the Solidity programming language for smart contract creation. It also allows testing and deploying smart contracts on the blockchain. It allows developing client-side applications.
- **Ganache:** Ganache functions as a local, in-memory blockchain. It's available for download from the Truffle Framework website and can be easily installed. Ganache furnishes users with 10 external accounts, each possessing Ethereum addresses on the local blockchain. Furthermore, every account is endowed with 100 dummy ether.
- **Metamask:** It is a browser extension that acts as an Ethereum wallet. After installing it, it allows users to store and perform transactions on any Ethereum address.
- **Node and Express** - Node and Express are used to setup the app. Node (or rather Node.js) is a cross-platform, open-source runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript. Express is the most popular Node web framework, and it serves as the foundation for several other prominent Node web frameworks.
- React - Client application is written with React, it is an open-source JavaScript framework used for quickly and efficiently constructing interactive user interfaces and web applications with substantially less code
- **Solidity**- Smart Contracts are written in Solidity which is a statically typed curly braces programming language suited for constructing Ethereum smart contracts.
- **Ethereum** - Dapp is developed using the Ethereum platform. Ethereum constitutes a worldwide network of computers governed by a predefined set of rules referred to as the Ethereum protocol. This network provides a platform for individuals to create and utilize communities, applications, organizations, and digital assets.

- **Truffle** - Truffle serves a testing framework and a development environment specifically designed for Blockchain utilization using the Ethereum Virtual Machine (EVM).

C. THE ASC SMART CONTRACTS

The business logic of the ASC process in the AgroChain is implemented by the developed smart contracts (refer to Appendix Figures 10, 11, 12, 13 and 14), managing digital records. The implemented functions of the proposed smart contracts can be categorized into four main categories as follows:

1) SMART CONTRACTS FOR ROLE MANAGEMENT

''Roles'' and ''Ownable'' contracts are implemented to manage different roles within the system, including farmers, distributors, retailers, and consumers. These contracts define roles, add or remove addresses from roles, and verify if an address has a specific role. The ''Ownable'' contract provides ownership control and allows for the setting of distributor and retailer margins. The sequence of interactions in the Ownable contract begins with the contract deployment by the deployer. The deployer becomes the initial contract owner (ContractOwner). Upon deployment, the constructor is executed, and the origOwner variable is set to the deployer's address. An event TransferOwnership is emitted, indicating the ownership transfer from address 0 to the deployer's address. The owner() function is used to fetch the current contract owner (ContractOwner), and the onlyOwner modifier ensures that only the contract owner can call certain functions. When the contract owner wants to transfer ownership to a new address (NewOwner), they call the transferOwnership function, and the onlyOwner modifier checks if the caller is the contract owner. If true, the _transferOwnership function is called, which again checks for ownership and then updates the origOwner variable to the new address. An event TransferOwnership is emitted to signify the change in ownership. The contract owner also has the option to renounce ownership by calling the renounceOwnership function. The onlyOwner modifier verifies that the caller is the contract owner before setting the origOwner variable to address 0 (zero address). An event TransferOwnership is emitted with the original owner's address and address 0 to indicate the ownership renouncement.

Additionally, the contract owner can set the distributor and retailer margins using the setDistributorMargin and setRetailerMargin functions, respectively. Again, the onlyOwner modifier ensures that only the contract owner can set these values. The margins are updated accordingly, allowing the contract owner to define the margins for distributors and retailers. Throughout the sequence, the onlyOwner modifier guarantees that only the contract owner can execute certain critical functions, providing basic authorization control to maintain ownership and control over the contract's settings.

2) ROLE-SPECIFIC CONTRACTS

''FarmerRole,'' ''DistributorRole,'' ''RetailerRole,'' and ''ConsumerRole'' contracts manage lists of participants in the respective roles and restrict access to certain functions based on role assignments. These contracts utilize the ''Roles'' library for role management and emit events when roles are added or removed. Role specific contracts include the following functions:

- Adding a farmer/ distributer/ retailer/ consumer

To add a new farmer/ distributer/ retailer or consumer, the ''addFarmer'', ''addDistributor'', ''addRetailer'' or addConsumer functions are called within the ''FarmerRole'', the ''DistributerRole'', the ''RetailerRole'' or the ''ConsumerRole'' contracts respectively. These contracts first verify whether the caller is already registered; if not, there addresses will be added.

- Removing a farmer/ distributer/ retailer/ consumer

To remove a farmer/ distributer/ retailer or consumer, the ''renounceFarmer'', ''renounceDistributor'', ''renounceRetailer'' or renounceConsumer functions are called within the ''FarmerRole'', the ''DistributerRole'', the ''RetailerRole'' or the ''ConsumerRole'' contracts respectively. The account caller is removed and an event is emitted to the blockchain.

- Verifying a farmer/ distributer/ retailer/ consumer

To check the caller accounts, the ''isFarmer'', ''isDistributer'', ''isRetailer'', ''isConsumer'' functions are called are called within the ''FarmerRole'', the ''DistributerRole'', the ''RetailerRole'' or the ''ConsumerRole'' contracts respectively.

3) SUPPLY CHAIN CONTRACT

The ''SupplyChain'' contract tracks the movement of agricultural products through the supply chain and manages product states such as Harvested, Processed, Packed, ForSale, Sold, Shipped, Received, and Purchased. It implements mechanisms to prevent price manipulation, including setting prices by farmers and calculating final prices with margins for distributors and retailers. The contract emits events at different stages of the supply chain process for transparency and auditability. Functions within the contract facilitate actions such as harvesting, processing, packing, selling, buying, shipping, receiving, and purchasing products.

The process starts by declaring and assigning a variable to the compiled smart contract artifact. The various constants and variables, such as the product ID, notes, and price, as well as the different account IDs and roles have to be declared and initialized.

The process then proceeds to the ''harvestItem()'' function of the smart contract by giving the farmer role to the originFarmerID account, emitting a ''Harvested()'' event. A Farmer can harvest an item by providing relevant details such as farm name, location, and product notes. This creates a new item with a unique Universal Product Code (UPC) and puts it in the Harvested state.

After that, the ''processItem()'' function is performed by emitting a ''Processed()'' event, marking an item as processed

using the "processItem()" function, and verifying the result set by fetching the item from the blockchain and comparing the returned values. It also checks if the emitted event was valid. The process then call the "packItem()" function by emitting a "Packed()" event, marking an item as packed using the "packItem()" function, and verifying the result set by fetching the item from the blockchain and comparing the returned values. It also checks if the emitted event was valid.

The next function to be performed is the "sellItem()" function, which emits a "ForSale()" event, marks an item as for sale using the "sellItem()" function, and verifies the result set by fetching the item from the blockchain and comparing the returned values. It also checks if the emitted event was valid. The Farmer sets the product's price, and the contract calculates and stores the middle price (with distributor margin) and final price (with retailer margin). The item is then marked as ForSale, and an event is emitted to inform external applications about the final price.

Following a distributor can buy the item, marking it as Sold and transferring the product's price to the Farmer. Then, the distributor can mark the item as Shipped. After that, a Retailer can receive the item, marking it as Received. Finally, a Consumer can purchase the item, marking it as Purchased.

4) SMART CONTRACT EXECUTION AND VERIFICATION

The process involves adding and verifying participants in different roles, such as farmers, distributors, retailers, and consumers. Functions within role-specific contracts handle the addition and removal of participants, along with verification of their roles. Ownership management is ensured through the "Ownable" contract, allowing only the owner to perform certain critical functions. The sequence of interactions includes adding/removing farmers, distributors, retailers, and consumers, along with ownership transfers and margin settings.

Figures 6, 7,8, and 9 present the main screens of the web interface of the AgroChain system.

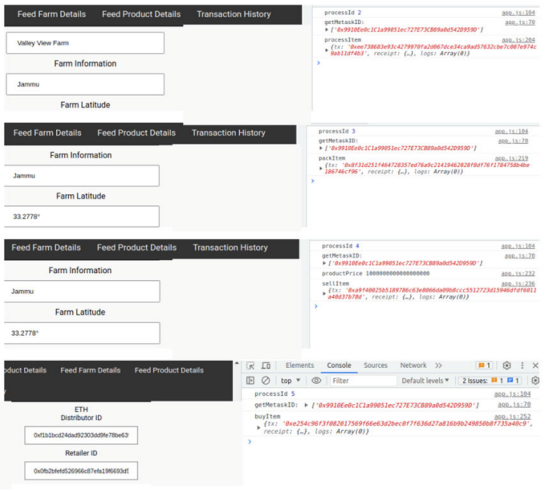

FIGURE 7. Screens of: Process an item, Pack an item, Sell an item and Buy an item.

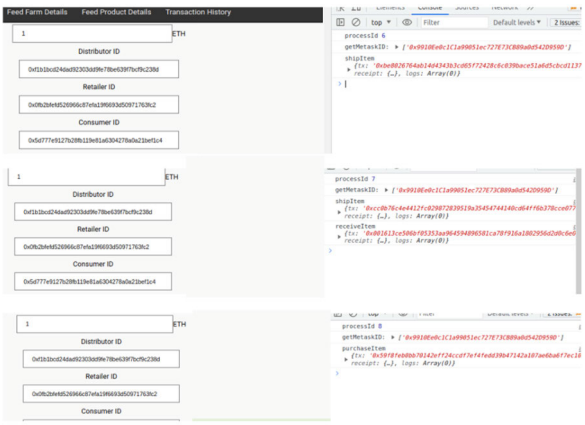

FIGURE 8. Screens of: Ship an item, Receive an item and Purchase an item.

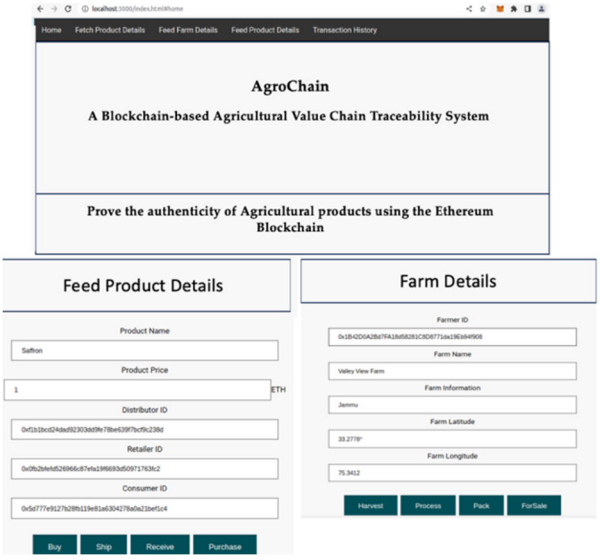

FIGURE 6. Home page, farm details and products details screens.

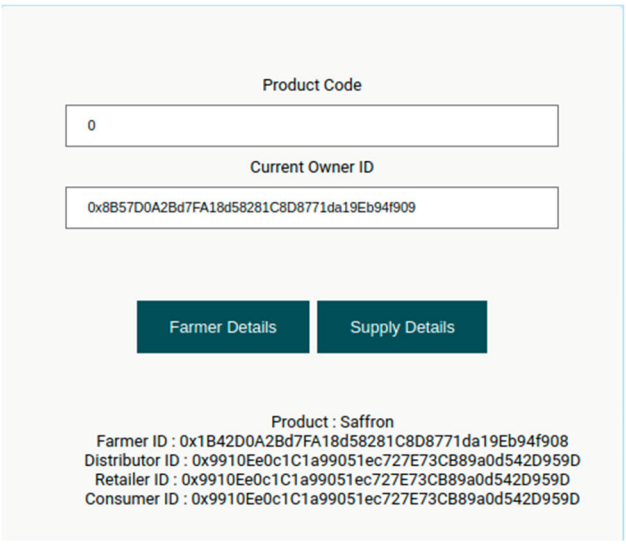

FIGURE 9. Supply details.

VI. DISCUSSION AND LIMITATION

Blockchain presents a viable solution to enhance the Agricultural Supply Chain (ASC) process, ensuring integrity and privacy among untrustworthy parties. The benefits of independent verification, decentralization, fault tolerance, and transparency are evident as demonstrated.

From a business standpoint, the costs associated with the ASC process are significant. However, our technological solution leverages smart contracts to streamline administrative procedures and reduce costs. Notably, costs are not incurred by access functions as the record registry and agricultural records themselves are stored in separate secure storage systems, as detailed previously. While GAS costs may be generated by audit and event generation in the implemented smart contract, the proposed AgroChain prototype has been developed for feasibility purposes demonstrating end-to-end implementation of the agricultural supply chain process with smart contracts in a robust, secure and an open manner.

For feasibility purposes, our prototype has been developed to validate the end-to-end implementation of the Agricultural Supply Chain (ASC) process using smart contracts, ensuring openness, security, and robustness. The principles and features of the proposed distributed architecture design outlined in Section IV are adhered to, encompassing record creation, access, and destruction, participant management, audit procedures, and more. Compatibility with mobile devices and native applications is ensured by the distributed application. The proposed smart contracts in the AgroChain prototype is designed to accommodate single and multiple entities, allowing for shared or dedicated usage. Optimization of size is aimed for in our solution while record integrity and privacy are maintained throughout the ASC process. Performance time is not considered a critical indicator, as real-time access or storage of record is not required by the process.

All events are managed by the proposed smart contracts, incorporating authentication/authorization permissions, associated data of the record, record access and registry. Furthermore, audit and tracking procedures are included in our prototype. An opportunity exists to standardize ASC smart contracts with support from authorities to enhance reliability and security, focusing on widely used programming languages as Solidity and Chaincode. Considering a multi-blockchain approach at the architectural level may prove advantageous, particularly since Blockchain technology is still evolving at the business level. Furthermore, deploying contracts within a consortium is imperative, given that the ASC process necessitates a comprehensive approach at both the users and process levels. The Blockchain technology is effective for technical purposes when coupled with robust record management and participant oversight. Our prototype reflects these aspects as a proof of concept.

Technologically, Alastria has incorporated Hyperledger into its roadmap, including Hyperledger Besu as a possible alternative Ethereum client project. Our modular and flexible technological architecture allows for the seamless integration of additional advancements to meet evolving requirements.

VII. CONCLUSION

This work presents a fully-functional Blockchain-based prototype, namely AgroChain that is designed to manage the Agricultural Supply Chain (ASC) process for any digital agricultural record. The prototype undergoes validation through the specification of three architectural designs (multi-blockchain, centralized, and distributed) and the delineation of several principles integrated into the prototype.

The proposed smart contracts are developed under Quorum, an Ethereum variant tailored for providing private business environments. As an open-source solution compatible with leading programming languages, it lays the groundwork for standardizing smart contracts in the ASC process. Promoting standardized smart contracts in a secure and straightforward manner is paramount. Leveraging the advantages of Zero-Knowledge Proof (ZKP) protocols and the principles embedded in our prototype, independent third parties can validate Blockchain transactions without accessing their content. Future enhancements to our prototype could include the incorporation of a multi-blockchain approach and support for Hyperledger Besu. Additionally, improvements to the user interface and interoperability of the smart contract with third parties, facilitated by ''Oracles'' gateways, present further avenues for refinement.

APPENDIX

The developed smart contracts are shown in the following figures.

```
pragma solidity >=0.4.23;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  constructor() public {
    owner = msg.sender;
  }

  modifier restricted() {
    if (msg.sender == owner) _;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }

  function upgrade(address new_address) public restricted {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}
```

FIGURE 10. Migration contract.

```
pragma solidity >=0.4.24;

import "./Roles.sol";

contract DistributorRole {
  using Roles for Roles.Role;

  event DistributorAdded(address indexed account);
  event DistributorRemoved(address indexed account);

  Roles.Role private distributors;

  constructor() public {
    _addDistributor(msg.sender);
  }

  modifier onlyDistributor() {
   require(isDistributor(msg.sender), "Caller is not a distributor.");
    _;
  }

  function isDistributor(address account) public view returns (bool) {
    return distributors.has(account);
  }

  function addDistributor(address account) public onlyDistributor {
    _addDistributor(account);
  }

  function renounceDistributor() public {
    _removeDistributor(msg.sender);
  }

  function _addDistributor(address account) internal {
    distributors.add(account);
    emit DistributorAdded(account);
  }

  function _removeDistributor(address account) internal {
    distributors.remove(account);
    emit DistributorRemoved(account);
  }
}
```

FIGURE 11. DistributerRole contract.

```
pragma solidity >=0.4.24;

import "./Roles.sol";

contract FarmerRole {
  using Roles for Roles.Role;

  event FarmerAdded(address indexed account);
  event FarmerRemoved(address indexed account);

  Roles.Role private farmers;

  constructor() public {
    _addFarmer(msg.sender);
  }

  modifier onlyFarmer() {
    require(isFarmer(msg.sender), "Caller is not a farmer");
    _;
  }

  function isFarmer(address account) public view returns (bool) {
    return farmers.has(account);
  }

  function addFarmer(address account) public onlyFarmer {
    _addFarmer(account);
  }

  function renounceFarmer() public {
    _removeFarmer(msg.sender);
  }

  function _addFarmer(address account) internal {
    farmers.add(account);
    emit FarmerAdded(account);
  }

  function _removeFarmer(address account) internal {
    farmers.remove(account);
    emit FarmerRemoved(account);
  }
}
```

FIGURE 12. FarmerRole contract.

```
pragma solidity >=0.4.24;

import "./Roles.sol";

contract RetailerRole {
  using Roles for Roles.Role;

  event RetailerAdded(address indexed account);
  event RetailerRemoved(address indexed account);

  Roles.Role private retailers;


  constructor() public {
    _addRetailer(msg.sender);
  }

  modifier onlyRetailer() {
    require(isRetailer(msg.sender), "Caller is not a retailer.");
    _;
  }

  function isRetailer(address account) public view returns (bool) {
    return retailers.has(account);
  }

  function addRetailer(address account) public onlyRetailer {
    _addRetailer(account);
  }

  function renounceRetailer() public {
    _removeRetailer(msg.sender);
  }

  function _addRetailer(address account) internal {
    retailers.add(account);
    emit RetailerAdded(account);
  }

  function _removeRetailer(address account) internal {
    retailers.remove(account);
    emit RetailerRemoved(account);
  }
}
```

FIGURE 13. RetailerRole contract.

```
pragma solidity >=0.4.24;

import "./Roles.sol";

contract ConsumerRole {
  using Roles for Roles.Role;

  event ConsumerAdded(address indexed account);
  event ConsumerRemoved(address indexed account);


  Roles.Role private consumers;


  constructor() public {
        _addConsumer(msg.sender);}

  modifier onlyConsumer() {
    require(isConsumer(msg.sender), "Caller is not a consumer.");
    _;
  }

  function isConsumer(address account) public view returns (bool) {
    return consumers.has(account);
  }

  function addConsumer(address account) public onlyConsumer {
    _addConsumer(account);
  }

  function renounceConsumer() public {
    _removeConsumer(msg.sender);
  }

  function _addConsumer(address account) internal {
    consumers.add(account);
    emit ConsumerAdded(account);
  }

  function _removeConsumer(address account) internal {
    consumers.remove(account);
    emit ConsumerRemoved(account);
  }
}
```

FIGURE 14. ConsumerRole contract.

REFERENCES

[1] C. Bai, M. Quayson, and J. Sarkis, "Analysis of blockchain's enablers for improving sustainable supply chain transparency in Africa cocoa industry," *J. Cleaner Prod.*, vol. 358, Jul. 2022, Art. no. 131896, doi: 10.1016/j.jclepro.2022.131896.

[2] S. Tyagi and M. Kathuria, "Role of zero-knowledge proof in blockchain security," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COM-IT-CON)*, vol. 1, Faridabad, India, May 2022, pp. 738–743, doi: 10.1109/COM-IT-CON54601.2022.9850714.

[3] E.-Y. Daraghmi, Y. A. Daraghmi, and S. M. Yuan, "MedChain: A design of blockchain-based system for medical records access and permissions management," *IEEE Access*, vol. 7, pp. 164595–164613, 2019, doi: 10.1109/ACCESS.2019.2952942.

[4] E.-Y. Daraghmi, Y.-A. Daraghmi, and S.-M. Yuan, "UniChain: A design of blockchain-based system for electronic academic records access and permissions management," *Appl. Sci.*, vol. 9, no. 22, p. 4966, Nov. 2019, doi: 10.3390/app9224966.

[5] Z. Alsaed, R. Khweiled, M. Hamad, E. Daraghmi, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "Role of blockchain technology in combating COVID-19 crisis," *Appl. Sci.*, vol. 11, no. 24, p. 12063, Dec. 2021, doi: 10.3390/app112412063.

[6] E.-Y. Daraghmi, M. Abu Helou, and Y.-A. Daraghmi, "A blockchain-based editorial management system," *Secur. Commun. Netw.*, vol. 2021, pp. 1–17, May 2021, doi: 10.1155/2021/9927640.

[7] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[8] R. Modi, *Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain*. Birmingham, U.K.: Packt, 2018.

[9] K. Baird, S. Jeong, Y. Kim, B. Burgstaller, and B. Scholz, "The economics of smart contracts," 2019, *arXiv:1910.11143*.

[10] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:209444308

[11] S. Shahab and Z. Allam, "Reducing transaction costs of tradable permit schemes using blockchain smart contracts," *Growth Change*, vol. 51, no. 1, pp. 302–308, Mar. 2020, doi: 10.1111/grow.12342.

[12] W. Zou, D. Lo, P. S. Kochhar, X. D. Le, X. Xia, Y. Feng, Z. Chen, and B. Xu, "Smart contract development: Challenges and opportunities," *IEEE Trans. Softw. Eng.*, vol. 47, no. 10, pp. 2084–2106, Oct. 2021, doi: 10.1109/TSE.2019.2942301.

[13] X. Liu, K. Muhammad, J. Lloret, Y.-W. Chen, and S.-M. Yuan, "Elastic and cost-effective data carrier architecture for smart contract in blockchain," *Future Gener. Comput. Syst.*, vol. 100, pp. 590–599, Nov. 2019, doi: 10.1016/j.future.2019.05.042.

[14] *Alastria*. Accessed: Apr. 20, 2024. [Online]. Available: https://alastria.io/

[15] Y. Liu, X. Ma, L. Shu, G. P. Hancke, and A. M. Abu-Mahfouz, "From Industry 4.0 to agriculture 4.0: Current status, enabling technologies, and research challenges," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4322–4334, Jun. 2021, doi: 10.1109/TII.2020.3003910.

[16] D.-H. Shih, K.-C. Lu, Y.-T. Shih, and P.-Y. Shih, "A simulated organic vegetable production and marketing environment by using Ethereum," *Electronics*, vol. 8, no. 11, p. 1341, Nov. 2019, doi: 10.3390/electronics8111341.

[17] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & Internet of Things," in *Proc. Int. Conf. Service Syst. Service Manage.*, Jun. 2017, pp. 1–6, doi: 10.1109/ICSSSM.2017.7996119.

[18] M. H. Ali, L. Chung, A. Kumar, S. Zailani, and K. H. Tan, "A sustainable blockchain framework for the halal food supply chain: Lessons from Malaysia," *Technological Forecasting Social Change*, vol. 170, Sep. 2021, Art. no. 120870, doi: 10.1016/j.techfore.2021.120870.

[19] M. P. Caro, M. S. Ali, M. Vecchio, and R. Giaffreda, "Blockchain-based traceability in agri-food supply chain management: A practical implementation," in *Proc. IoT Vertical Topical Summit Agricult.-Tuscany (IoT Tuscany)*, May 2018, pp. 1–4, doi: 10.1109/IoT-TUSCANY.2018.8373021.

[20] S. Malik, S. S. Kanhere, and R. Jurdak, "ProductChain: Scalable blockchain framework to support provenance in supply chains," in *Proc. IEEE 17th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2018, pp. 1–10, doi: 10.1109/NCA.2018.8548322.

[21] F. Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," in *Proc. 13th Int. Conf. Service Syst. Service Manage. (ICSSSM)*, Jun. 2016, pp. 1–6, doi: 10.1109/ICSSSM.2016.7538424.

[22] V. Sudha, R. Kalaiselvi, and P. Shanmughasundaram, "Blockchain based solution to improve the supply chain management in Indian agriculture," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Mar. 2021, pp. 1289–1292, doi: 10.1109/ICAIS50930.2021.9395867.

[23] X. Li, D. Wang, and M. Li, "Convenience analysis of sustainable e-agriculture based on blockchain technology," *J. Cleaner Prod.*, vol. 271, Oct. 2020, Art. no. 122503, doi: 10.1016/j.jclepro.2020.122503.

[24] L. Ge, C. Brewster, J. Spek, A. Smeenk, J. Top, F. V. Diepen, B. Klaase, C. Graumans, and M. D. R. de Wildt, "Blockchain for agriculture and food: findings from the pilot study," Wageningen Econ. Res., Wageningen, The Netherlands, Tech. Rep. 2282300245, 2017, doi: 10.18174/426747.

[25] S. V. Yadav, "A systematic literature review of blockchain technology in agriculture," in *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, Jun. 2019, pp. 973–981.

[26] J. Lin, Z. Shen, A. Zhang, and Y. Chai, "Blockchain and IoT based food traceability for smart agriculture," in *Proc. 3rd Int. Conf. Crowd Sci. Eng.*, Jul. 2018, pp. 1–6, doi: 10.1145/3265689.3265692.

[27] W. Hong, Y. Cai, Z. Yu, and X. Yu, "An agri-product traceability system based on IoT and blockchain technology," in *Proc. 1st IEEE Int. Conf. Hot Information-Centric Netw. (HotICN)*, Aug. 2018, pp. 254–255, doi: 10.1109/HOTICN.2018.8605963.

[28] *UNDP Global Centre for Technology, Innovation and Sustainable Development, Blockchain for Agri-Food Traceability*, UNDP, Singapore, 2021.

[29] D.-H. Nguyen, N. H. Tuong, and H.-A. Pham, "Blockchain-based farming activities tracker for enhancing trust in the community supported agriculture model," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 737–740, doi: 10.1109/ICTC49870.2020.9289297.

[30] S. Revathy and S. S. Priya, "Blockchain based producer-consumer model for farmers," in *Proc. 4th Int. Conf. Comput., Commun. Signal Process. (ICCCSP)*, Sep. 2020, pp. 1–5, doi: 10.1109/ICCCSP49186.2020.9315214.

[31] S. Umamaheswari, S. Sreeram, N. Kritika, and D. R. Jyothi Prasanth, "BIoT: Blockchain based IoT for agriculture," in *Proc. 11th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2019, pp. 324–327, doi: 10.1109/ICoAC48765.2019.246860.

[32] A. Vangala, A. K. Das, N. Kumar, and M. Alazab, "Smart secure sensing for IoT-based agriculture: Blockchain perspective," *IEEE Sensors J.*, vol. 21, no. 16, pp. 17591–17607, Aug. 2021, doi: 10.1109/JSEN.2020.3012294.

[33] *OpenZepellin Contracts*. Accessed: Apr. 20, 2024. [Online]. Available: https://github.com/OpenZeppelin/openzeppelin-contracts

[34] MetaMask. *MetaMask: The Crypto Wallet for Defi, Web3 Dapps and NFTs*. Accessed: Apr. 20, 2024. [Online]. Available: https://metamask.io/

[35] Trufflesuite. *Truffle Suite—Truffle Suite*. Accessed: Apr. 20, 2024. [Online]. Available: https://trufflesuite.com/

EMAN-YASER DARAGHMI received the B.S. degree in communication and information technology from Al-Quds Open University, in 2008, and the M.S. and Ph.D. degrees in computer science from National Chiao Tung University, Taiwan, in 2011 and 2015, respectively. She is currently the Dean of the Integrated Education College, Palestine Technical University—Kadoorie (PTUK). She is an Associate Professor with the Department of Computer Science, PTUK. She has extensive experience in developing decentralized applications and smart contracts. Her research interests include significant contributions to the fields of blockchain and distributed ledger technologies.

SHADIA JAYOUSI is currently pursuing the master's degree with the Department of Cybercrimes and Digital Evidence Analysis, PTUK. Her research interests include deep learning, machine learning, and cybercrimes.

YOUSEF-AWWAD DARAGHMI received the bachelor's degree in electrical engineering from An-Najah National University, Palestine, in 2002, and the master's degree in computer science and information engineering and the Ph.D. degree in computer science and engineering from National Chiao Tung University, Taiwan, in 2007 and January 2014, respectively. He is currently an Associate Professor with the Computer Systems Engineering Department, Palestine Technical University—Kadoorie. His research interests include blockchain, intelligent transportation systems, and vehicular ad-hoc networks. He serves as a TPC member for several conferences and a reviewer for highly distinguished journals. He received the Best Paper Award from ITST, in 2012.

RAED S. M. DARAGHMA was born in Palestine, in 1977. He received the master's degree in electrical and communication engineering from Jordan University of Science and Technology, Jordan, in 2010, and the Ph.D. degree from Anadolu University, Türkiye, in 2016. He has published a number of papers and journals. He has engaged in educational work for many years mainly teaching digital communication, mobile, and digital communication networks with Palestine Technical University—Kadoorie (PTUK). His main research interests include wireless sensor networks, signal processing, MIMO radar, and electromagnetic waves.

HACÈNE FOUCHAL received the M.S. degree in computer science from Université de Paris Sud, Orsay, in 1989, the Ph.D. degree from Université de Paris 7, in 1995, and the Habilitation degree in computer science from Université de Reims Champagne-Ardenne, in 2001. Currently, he is a Full Professor with Université de Reims Champagne-Ardenne and the Vice-Head of the Mathematics, Mechanics and Computer Science Department. He has multiple publications in international sources. He has supervised more than 15 Ph.D. students. His research interests include sensor networks, distributed systems, eHealth systems, networking, testing, verification, and intelligent transport systems. He was the Steering Committee Chair of the International Conference of Principles of Distributed Systems (Springer, 2006–2012). He has been the General Chair and the Technical Program Chair of several conferences (GLOBECOM, ICC, OPODIS, and I4CS). He was the Chair of the Technical Committee Communication Software of the IEEE Communication Society, from 2017 to 2019. He has been a Guest Editor of many publications, including *Concurrency and Computation: Practice and Experience* (Wiley) and *Journal of Computational Science* (Elsevier). He has been involved in several European projects on intelligent transport systems (C2A, SCOOP@F, InterCor, C-Roads, and Indid), since 2014.