sensors

MDPI

Article

An Implementation of Trust Chain Framework with Hierarchical Content Identifier Mechanism by Using Blockchain Technology

Hsing-Chung Chen [1,2,*], Bambang Irawan [1,3], Pei-Yu Hsu [1], Jhih-Sheng Su [1], Chun-Wei (Jerry) Lin [4], Prayitno [1,5], Karisma Trinanda Putra [6], Cahya Damarjati [7], Chien-Erh Weng [8,*], Yao-Hsien Liang [1] and Pi-Hsien Chang [9]

[1] Department of Computer Science and Information Engineering, Asia University, Taichung 413305, Taiwan; bambang.irawan@esaunggul.ac.id (B.I.); hcliebe2019@gmail.com (P.-Y.H.); 108121019@live.asia.edu.tw (J.-S.S.); prayitno@polines.ac.id (P.); 108021072@live.asia.edu.tw (Y.-H.L.)
[2] Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 404327, Taiwan
[3] Department of Computer Science, Esa Unggul University, West Jakarta 11510, Indonesia
[4] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied, 5063 Bergen, Norway; jerrylin@ieee.org
[5] Department of Electrical Engineering, Politeknik Negeri Semarang, Semarang 50194, Indonesia
[6] Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Bantul 55183, Indonesia; karisma@ft.umy.ac.id
[7] Department of Information Technology, Universitas Muhammadiyah Yogyakarta, Bantul 55183, Indonesia; cahya.damarjati@umy.ac.id
[8] Department of Telecommunication Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 81157, Taiwan
[9] Information Management Center, Taichung City Government, Taichung 407610, Taiwan; ps491@taichung.gov.tw
* Correspondence: shin8409@ms6.hinet.net or cdma2000@asia.edu.tw (H.-C.C.); ceweng@nkust.edu.tw (C.-E.W.)

Abstract: Advances in information technology (IT) and operation technology (OT) accelerate the development of manufacturing systems (MS) consisting of integrated circuits (ICs), modules, and systems, toward Industry 4.0. However, the existing MS does not support comprehensive identity forensics for the whole system, limiting its ability to adapt to equipment authentication difficulties. Furthermore, the development of trust imposed during their crosswise collaborations with suppliers and other manufacturers in the supply chain is poorly maintained. In this paper, a trust chain framework with a comprehensive identification mechanism is implemented for the designed MS system, which is based and created on the private blockchain in conjunction with decentralized database systems to boost the flexibility, traceability, and identification of the IC-module-system. Practical implementations are developed using a functional prototype. First, the decentralized application (DApp) and the smart contracts are proposed for constructing the new trust chain under the proposed comprehensive identification mechanism by using blockchain technology. In addition, the blockchain addresses of IC, module, and system are automatically registered to InterPlanetary File System (IPFS), individually. In addition, their corresponding hierarchical CID (content identifier) values are organized by using Merkle DAG (Directed Acyclic Graph), which is employed via the hierarchical content identifier mechanism (HCIDM) proposed in this paper. Based on insights obtained from this analysis, the trust chain based on HCIDM can be applied to any MS system, for example, this trust chain could be used to prevent the counterfeit modules and ICs employed in the monitoring system of a semiconductor factory environment. The evaluation results show that the proposed scheme could work in practice under the much lower costs, compared to the public blockchain, with a total cost of 0.002094 Ether. Finally, this research is developed an innovation trust chain mechanism that could be provided the system-level security for any MS toward Industrial 4.0 in order to meet the requirements of both manufacturing innovation and product innovation in Sustainable Development Goals (SDGs).

check for updates

Citation: Chen, H.-C.; Irawan, B.; Hsu, P.-Y.; Su, J.-S.; Lin, C.-W.; Prayitno; Putra, K.T.; Damarjati, C.; Weng, C.-E.; Liang, Y.-H.; et al. An Implementation of Trust Chain Framework with Hierarchical Content Identifier Mechanism by Using Blockchain Technology. *Sensors* **2022**, *22*, 4831. https://doi.org/10.3390/s22134831

Academic Editor: Nikos Fotiou

Received: 25 May 2022
Accepted: 22 June 2022
Published: 26 June 2022

Keywords: trust chain framework; blockchain; comprehensive identification mechanism; manufacturing innovation and product innovation

1. Introduction

In the last decade, the manufacturing system (MS) has been one of the major concerns in the development of Industry 4.0. Both manufacturing and assembly are competitive sectors, which is the target for parties who take illegal profits. Many interrelated parties take roles in this sector. Manufacturing industries include around 628,536 companies in the US and contribute to 24% of the US GDP according to statistics in 2020 [1]. Moreover, geolocation also has implications; for instance, although the designs of the components are still dominated by the US, Taiwan and South Korea dominate the chip-manufacturing industry, contributing to 83% of global production [2]. Then, in late 2020, a chip shortage emerged by the industry in the downstream sector, including the car and computer industries [3]. Manufacturing enterprises must shift towards MS to maintain advantages in a competitive global market [4,5].

Mass customized zero-defect production is a significant challenge in MS [6,7]. The more massive the demand for customized products, the higher the potential for counterfeiting. Furthermore, some counterfeiters have begun manufacturing their products in the same factory as the original product [8]. To solve the above challenges, manufacturing enterprises need to maintain their innovation. For instance, to increase production efficiency and save manufacturing costs, the company needs to increase production capacity using some of the latest technological developments, such as the Industrial Internet of Things (IIoT) [9]. Next, manufacturing companies need to collaborate with several suppliers and other manufacturing industries to produce final collaborative products that meet global market requirements. Finally, transparency during the production phases needs to be improved, providing consumers with product safety and well-tracked warranty claims. To adopt cutting-edge, manufacturers must implement advanced information technology. Unfortunately, the MS foundation is based on centralized computing with poor flexibility [10]. In addition, the development of trust imposed during crosswise collaborations with suppliers and other manufacturers is poorly maintained, thus creating complex, poorly managed, and error-prone collaborations [11]. Therefore, trustable methods and chains are the only way to track product originality and reduce counterfeits.

As one of the advances in new-generation information technologies, blockchain improves digital transactions in efficiency, transparency, and security. As a new paradigm, the comprehensive identification mechanism using blockchain technology states that data could be communicated, tracked, and stored securely [12]. The trust chain enables distributed trust to be scalable. An immutable record of the transaction is created by allowing two parties to agree on a trade or transaction and storing the proof of the transaction in a firm form. A blockchain consists of a series of blocks that record each transaction. Every peer maintains a copy of their blockchain, containing records of all transactions. Blockchain changes the paradigm of a conventional centralized system into a decentralized trusted system. The existing centralized architecture suffers from the weakness as follows. (1) IIoT is built because all end nodes, that is, connected to a centralized system, are initially verified by the system. After one of the end nodes is disconnected from the system, there is no guarantee that the system could trust the same node. A re-validation process needs to be performed by the system, and the end node is broken in the system. (2) The performance of the massive centralized systems is greatly affected by latencies and errors. Moreover, if new nodes repeatedly manage further sub-branches inside the old one, a longer network chain will be developed, thus increasing the latencies and errors. (3) The competitive manufacturing sector is forcing companies to switch to current MS because customers now want to be involved in the manufacturing process to make a more customized and personalized product. Meanwhile, current MS faces security problems because the system does not guarantee the validity of new users. Blockchain technology combined with a decentralized database offers the practical concept of realizing decentralized current MS. However, in the industrial sector, blockchain technology is only a concept and lacks implementation for the current MS [13–15]. It is still challenging to provide a consensus technique suitable for applications [16] in the industrial domain with reliable validation techniques.

The main contributions of this paper are stated below.

(1) Implement the consortium blockchain via applying the Quorum blockchain network (QBN). Combining it with a decentralized database, the InterPlanetary File System (IPFS) provides flexibility, traceability, and security mechanisms for current MS.
(2) Both nature Merkle trees in QBN plus Merkle DAG (Directed Acyclic Graph) are employed and implemented for storing the CIDs and the corresponding product version files of ICs, modules, and systems, individually.
(3) The web-based DApp is implemented by using the smart contracts designed in this study to interact with the QBN together with the IPFS collaboration framework, where the hierarchical content identifier mechanism (HCIDM) is first proposed and designed for each customized system product in order to create the system-level trust chain and easily validate or trace the trusted product versions of IC, module, and the customized system itself, comprehensively.
(4) The consensus-oriented transaction logic based on the voting-based consensus protocol [16] in the implemented QBN shows that it could improve latency and throughput compared with the previous consensus protocols [17].

In addition, the prototype concept's implementation is presented to meet the requirements of the MS plus consortium blockchain for the manufacturing industry. The evaluation will be demonstrated that the proposed idea is practical and efficient with a flexible, traceable, and secure decentralized working environment in this paper.

Finally, the remains of this paper are organized below. Section 2 discusses related works, while in Section 3, we present the system model used in this study. Section 4 evaluates experimental results. Section 5 presents discussions and analyses. Finally, Section 6 summarizes this study.

2. Related Works

In this section, the current MS and its relative technologies will be introduced. Next, some solutions for implementing different MSs using the blockchain technology approach will be discussed. Additionally, the smart contracts will be also explained. The distributed data storage system, for example, IPFS will be addressed. Finally, Dapp will be given short descriptions and informed on how to co-work with smart contracts.

Currently, MS is still improving industrial technology for satisfying the requirements of manufacturing efficiency, transparency, and security [12]. In the other words, MS still pays his full efforts to optimize production and product transactions as a broad manufacturing concept toward Industry 4.0 by integrating advanced manufacturing with information engineering [18]. With advancements in information engineering, these cutting-edge models might likely be employed to address the shortcomings of the present production model [19–21]. Furthermore, with the emergence of the IIoT concept based on small intelligent sensors, the future production line is supported by heterogeneous sensory modules that functionally collaborate to support the manufacturing process [22]. With more functional nodes connected in a centralized IIoT system, manufacturing enterprises must shift towards an MS with many promised benefits, for example, intelligent, and remote management. Each functional module is fitted with numerous sensory units that monitor the module's components to ensure a more precise perception and grasp of the manufacturing process. Thus, management parties could precisely monitor each component, improving their oversight of complex parts to ensure that production phases go smoothly. However, the MS concept needs to be revised with a new approach with increasingly varied product demands. The MS should bring customers closer to the manufacturing process, making it easier to get personalized custom products [12,23]. However, this centralized private system, for example, the current MS, cannot handle various problems, including counterfeiting [24], copyright [25], and consumer protection [26], due to the lack of united records from all parties who participate in production and distribution processes. Even though the manufacturer is an enterprise-level party, in the process, they cooperate with third parties in a multilevel subcontracting mechanism making each base component hard to be traced. Therefore, a decentralized, simple, traceable, and secure network is required in the development of current MS, and this is what blockchain technology offers.

Blockchain defines that although they support both private and public networks, the data can be confidently communicated, traced, and securely stored [27]. In a concept that combines MS and IIoT, several functional sensors measure real-time data from the production lines [18]. This intelligent concept paves the way for better monitoring and understanding of industrial phases to perform effective and efficient production. Moreover, the current MS concept that brings consumers closer to the manufacturing processes could expose the system to the outside world, thus increasing the chance of cyberattacks [28]. Traditional protection mechanisms fail to protect centralized systems due to the low computing resource at edge devices [29]. The deployment of centralized systems might be more expensive for companies as powerful central servers and regular maintenance are required. Table 1 presents various solutions for implementing different MSs using the blockchain technology approach.

Table 1. Comparison with various manufacturing systems.

Authors	Year	Objective	Technologies
Zhang, C. et al. [12]	2020	Their permissioned blockchain could achieve better throughput and lower latency than the permissionless blockchain, making it more resource-efficient and appropriate for MBCoT.	Integrates IIoT with the permissioned blockchain.
Zhong, R.Y et al. [18]	2017	This article presented a framework of Industry 4.0-based IMS, in which research topics are categorized into smart design, smart machines, smart monitoring, smart control, and smart scheduling.	The IoT, CPSs, cloud computing, big data analytics, and other information and communication technologies (ICTs).
Ma, J. et al. [24]	2020	Manufacturers could use the system to store relevant information on product sales in blockchain which is accessible to everyone.	Based on Ethereum blockchain.
Xiao, L et al. [25]	2020	The secure, intelligent contract protocol of Intellectual Property circuit protection under the blockchain environment further enhances security and reliability.	The blockchain environment.
Heo, G et al. [26]	2021	The proposed SBBC system includes off-chain and on-chain modules to establish secure and reliable digital content trading.	The secret block-based blockchain (SBBC).

In general, blockchain could be constructed with four parts consisting of the ledger, smart contracts, consensus, and cryptography [30]. In a blockchain network, the ledger archives all transactions by the parties, and each party keeps a copy of it. A smart contract authorizes access to the ledger. A consensus is used to synchronize the ledger across the network. Finally, the cryptography mechanism envelops the network transactions and the ledger data, making these binds hard to break and trace by eavesdroppers. In terms of blockchain platforms, there are two kinds of platforms, namely, without authorization (i.e., permissionless blockchain) and with authorization (i.e., permission blockchain) [31]. Any party could participate in the permissionless blockchain network without authorization, e.g., Bitcoin [32] and Ethereum [33].

On the contrary, the permission blockchain implies that members should have a specific identity, that is, Hyper Fabric [34] and Quorum [35]. Substantially, both higher throughput and lower latency are performed by permission blockchain due to their nature of consensus. One consensus algorithm in a permissionless blockchain, such as Proof of

Works (PoWs), is complex and costly to handle Sybil attacks [36]. On the other hand, PoA consensus in a permissioned blockchain ensures that all members are equally motivated to succeeding their network, although with a lower number of members but with a more trusted identity [37]. This concept is more implementable for enterprise-level applications such as manufacturing industry cases.

In addition, the ledger functions in a networked environment based on decentralization in the blockchain world. Each ledger block contains data regarding platform transactions. The block is produced using the data, hash function, and hash value from the preceding block. The process is called a chain [38,39]. According to most definitions, a blockchain is a decentralized database where cryptographic signatures verify transactions. Consensus algorithms like proof of stake (PoS) and PoW allow blockchain to confirm transactions based on the agreement of all peers [40–42]. Both PoS and PoW protocols ensure that transactions between peer nodes in a blockchain network are safe and dependable. Public, private, or consortium are the classifications in blockchain [4]. The public blockchain network is an architecture that could read given notes. It denotes being accessible to anyone on Earth, validating its status as a node, and participating in consensus. Crypto-economics is primarily concerned with the convergence of economic incentives with cryptographic verification via proof-of-work (Bitcoin) or proof-of-stake (Ethereum). Blockchain technology is generally regarded as fully decentralized. One downside is the substantial computational power (resources) necessary to maintain large-scale distributed ledgers, crucial in densely deployed IoT networks. By implementing rigorous constraints, the shared blockchain network protects end-users from developers. A fully private blockchain gathers written permissions into the hands of a single institution. The scope of reading permissions could be public or arbitrary. In many circumstances, internal applications such as database management and auditing should not require general readability, while public audibility may be desired in others. A consortium blockchain is a multi-center blockchain system built and maintained collaboratively by an agreed group. Typically, each node corresponds to an entity organization. After finishing both authentication and authorization processes, each node can join, access, and transmit transactions. In addition, each consortium member has specific data permissions. The blockchain consortium is the dominant approach as it enables voluminous and robust collaboration across companies and organizations, promoting the healthy and orderly development of the blockchain industry.

Next, smart contracts are computer programs that facilitate, verify, and enforce legal transactions. Implemented through blockchain transactions, linked to cryptocurrencies, and an input interface for contracting parties. When implemented on the blockchain, smart contracts become autonomous entities capable of performing specific tasks automatically when certain conditions are met. The execution of smart contracts on the blockchain is not subject to censorship, downtime, fraud, or third-party interference. Ethereum is the most widely used smart contract platform in the sector. Smart contracts provide distributed trusted computing on a blockchain platform. Embedded hardware and software translate the current contract clauses into smart contracts to verify that the contract is eligible. Code-based smart contracts can interact with other contracts, make decisions, store data, and send tokens/money to other contracts.

Moreover, IPFS is a distributed data storage system that addresses content and assigns a unique hash to each item stored [43]. IPFS CID version "0" and "1" enables a high-throughput, resource-efficient storage paradigm with concurrent access. IPFS generates a hash that is 46 bytes in length as a result, putting transaction data in IPFS, and the hash generated by IPFS in the blockchain block results in significant savings in storage space [43]. A larger file is fragmented and stored on multiple nodes. The term "node" refers to the various computing devices that comprise the IPFS network. When data are retrieved, it is associated with a hash key based on the content to which it refers. Using these hash keys, one could simultaneously access data from several nodes. The term "content-addressable storage" refers to this principle.

Additionally, a decentralized application (DApp) [44] is a decentralized network-based program that combines smart contracts and a user interface on the front end due to the open and transparent nature of Ethereum smart contracts, which are analogous to an open API [40]. Our DApp could integrate smart contracts established by others. The backend code of DApp is hosted on a decentralized peer-to-peer network. Compared to legacy systems, which rely on a centralized server to execute backend code. In general, a programmer could easily write a DApp frontend code and user interface in any language to make calls to its backend (such as an App). Additionally, it is possible to host its frontend on a distributed storage system such as IPFS [45].

The primary purpose of distributed data storages for blockchain is to provide storage for all transaction records. It does not provide a huge distribution data set for all data types. Blockchain is currently the most efficient and secure methods [46–48] of storing information as a distributed ledger, facilitating data sharing among multiple parties under their collaborations. Moreover, there are many blockchain applications [49–53] by using and deploying blockchain technologies to provide the security in distribution system. IPFS is a peer-to-peer decentralized file system [45]. Its protocol is used to create a network infrastructure similar to the Bitcoin blockchain protocol, with the advantages of storing immutable data and removing redundant files on the network. In other words, IPFS provides a high-throughput, content-addressed block storage model, and content-related hyperlinks in the created network infrastructure. It could form a generalized Merkle DAG [54]. IPFS does also combine decentralized hash tables, incentivized block exchanges, and a self-certifying namespace [55]. IPFS has no single point of failure and nodes do not need to trust each other [55]. Decentralized content delivery saves bandwidth and prevents DDoS attacks that HTTP schemes might encounter [56]. Therefore, blockchain and IPFS could collaborate to produce decentralized storage. It could store and distribute various data types without relying on third parties. The approach of combing both blockchain and IPFS is used to circumvent storage limits and allow secure storage of individual data. It could improve privacy, security, and distributed data management.

The related works, mentioned above, show that the convergences of blockchain and decentralized databases may potentially overcome the drawbacks of the current MS. However, blockchain applications, specifically in the industrial domain, are still in their infancy. Integrating blockchain into a current MS concept is still challenging. Thus, this study shortcoming the above challenges with a practical integration of blockchain technology in current MS scenarios.

In this paper, the consortium blockchain technology, namely the Quorum Blockchain Network (QBN) [45], is used to develop a trust chain system. The quorum blockchain network comprises four critical components:

1. The transaction manager provides access to encrypted transaction data for private transactions, manages locally stored data, and facilitates communication with other transaction managers.
2. A cryptographic enclave is responsible for maintaining private keys and the encryption and decryption of personal transaction data.
3. The Byzantine Fault Tolerance Consensus (BFT) [45] relies on voting participants joining the quorum chain to validate and send it on the network, based on Ethereum capabilities.
4. The network manager manages network access, enabling the creation of a permissioned network.

3. Preliminaries

This section discusses some main concepts underlying the traceability technology of blockchain-based integrated systems, modules, and ICs. To get started, explore the role of blockchain and the IPFS in maintaining immutable records of ownership systems, modules, and ICs and the role of smart contracts in automating the registration, authentication, traceability, and transfer of ownership of systems, modules, and ICs. In this study, the

DApp is designed by using a hierarchical blockchain smart contract and interacts with the IPFS collaboration platform.

In this work, the system manufacturers could use this framework to store necessary information about finished products on blockchain to protect their finished products from copyright infringement and counterfeiting. The Quorum blockchain technology allows encrypted data to be visible only to the parties participating in the transaction, and therefore the system is based on this principle. The production process of a new system starts from the system manufacturer's request to the module manufacturer. Figure 1 shows the distributed chain flow for system manufacturers, including System Manufacturer (SM), Module Manufacturer (MM), and IC Manufacturer (ICM), forming a chain of alliances. The proposed framework allows the verification and recording of completed data immediately.

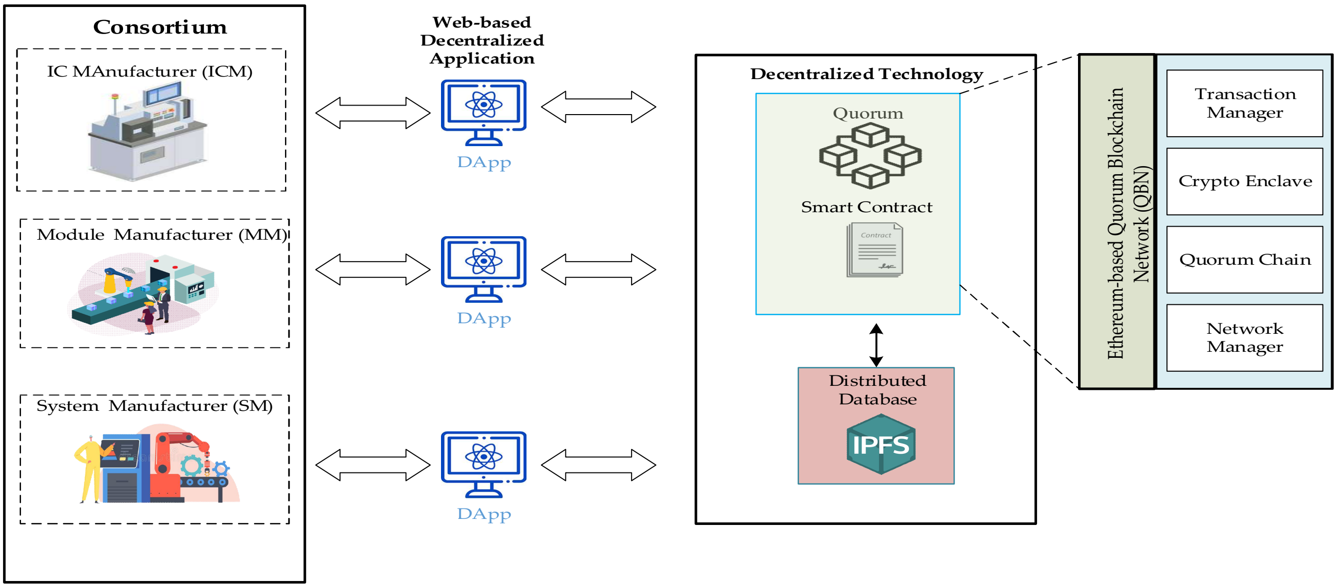


Figure 1. The proposed framework including Quorum blockchain, distributed database (IPFS), and DApp.

Each participant will also register in the proposed blockchain and then be assigned a private-public key pair to identify unique members on the quorum blockchain. The 256-bit private key generates the public key via Elliptical Curve Cryptography [38,46]. The private key must remain secret. Next, every transaction requires a digital signature calculated by Equation (1) below.

$$Kpri[0,1]^{256}. \tag{1}$$

Afterward, a 512-bit public key is produced using the ECDSA (Elliptical Curve Digital Signing Technique) algorithm from the private key. The primary advantage of ECDSA [38,46] over other Public Key Cryptography (PKC) algorithms, such as RSA, is that it needs significantly more minor keys and signatures to obtain a comparable level of security. Signatures are validated using the public key. According to the ECDSA protocol [38,46], Equation (2) is given below.

$$Kpub = ECDSA(Kpri). \tag{2}$$

A transaction's owner embeds their public key and then signs it with the private key. Due to the small size of 512-bit public keys, they could be shared and used in blockchain transactions. To protect against length extension attacks, use a double hash approach in which the 512-bit public key is hashed using SHA-256 and then hashed again with RIPEMD-160 to obtain the 160-bit address by Equation (3).

$$Kaddr = RIPEMD_{160}\left(SHA_{256}\left(K_{pub}\right)\right). \tag{3}$$

QBN is critical in authenticating participants and maintaining a directory that associates related party identifiers with transaction addresses. Any identity management system based on blockchain technology can ensure the veracity of user entity information. Users can manage private keys and addresses through digital wallets, conducting any transaction. Digital wallets are a combination of software and hardware or a custom-built hardware device that stores the private-public key and address.

4. Our Proposed Framework

In this paper, we propose a framework divided into three phases: Initialization Phase in Section 4.1, Registration Phase in Section 4.2, and Production Phase in Section 4.3.

The architecture of this framework could be illustrated in Figure 2.

4.1. Initialization Phase

Each participant with a different role forms a consortium during the initialization phase. Table 2 shows the initial code structure of the smart contracts used in the QBN to organize the trust chain, in which the left-side structure describes a system, modules, and ICs. In contrast, a right-side structure reveals the types of roles involved. In other words, each manufacturer plays a specific role in the consortium. The QBN architecture has a basic chain code structure shown in Table 2, with the symbol notation described in Table 3. When a new system product is created, the left side contains information about the system created by the system manufacturer. In contrast, the right side depicts the structure of the trust chain and the enumeration of the various roles. In addition, the notations are addressed in Table 3.

4.2. Registration Phase

SM, as the node manager, will create the first blockchain network. QBN is a blockchain network that we created. Both MM and ICM join the quorum blockchain network through the SM node manager and get the corresponding public and private keys in QBN. Each role refers to a role node registered in the blockchain system, where the registration procedure is described in Figure 3. In QBN, each participant has a node name, role, IP address, node latency, and an E-wallet application with the account address that is the blockchain address. Moreover, this account address is derived by a public key in the designated blockchain network, where the public key is uniquely associated with a single private key. This public key and private pair are stored in the E-wallet.

Table 2. The code structure of smart contract.

The System Code Structure	The Types of Roles
• Type system struct{ S_ID string system_inform string system_Crea_Datetime time.Time Module_manufac_ID string Module_manufac_Datetime DateTime IC_Manufac_ID string IC_ Manufac _Datetime DateTime MS_Sign string MM_Sign string ICM_Sign string } • Type Module struct{ M_ID string Module_manufac_Datetime DateTime IC_Manufac_ID string IC_ Manufac _Datetime DateTime MM_Sign string ICM_Sign string } • Type IC struct{ IC_ID string IC_inform string IC_Manufac_ID string IC_ Manufac _Datetime DateTime ICM_Sign string }	• Type Roles string const{ System Manufac Module Manufac IC Manufac } • Type Node_table struct{ Node_name string Role string public-key string IP address Enode_ID string }

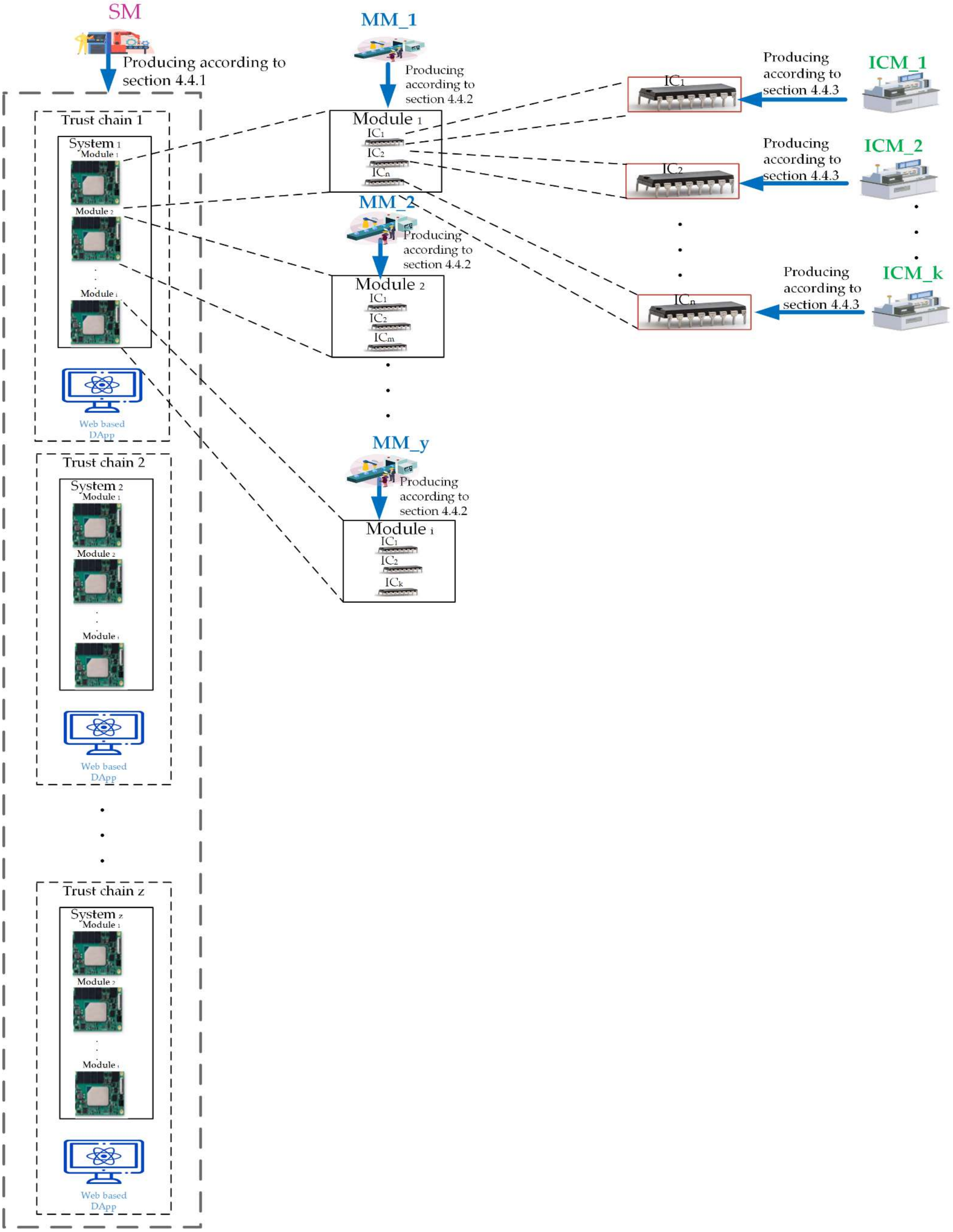


Figure 2. The architecture of this framework.

Table 3. Notations.

$\boldsymbol{ID_r}$	**The Identity of a Manufacturer. For example, System Manufacture, Module Manufacture, and IC Manufacture.**
SM	The System Manufacturer.
MM	The Module Manufacturer.
ICM	Ihe IC manufacturer.
$Kpri\ _r$	The private key for participant *r*.
$Kpub_r$	The public key for participant *r*.
$Kaddr_r$	The address key for participant *r*.
G	A generating point based on Elliptic Curve E.
S_IDi	S_ID is the system's *i*th identifier (blockchain address).
M_IDi	M_ID is the module's *i*th identifier (blockchain address).
IC_IDi	IC_ID is the system's *i*th identifier (blockchain address).
E	The Elliptic Curve is defined as a finite group.
Ti	It is the *i*th timestamped value.
ΔT	The validity threshold of the time stamp.
M_{SM}	The nessage sent by the system manufacturer.
M_{MM}	The message sent by the module manufacturer.
M_{ICM}	The message sent by the IC manufacturer.
(R_{ri},S_{ri})	The Elliptic Curve signature value for the role *r*.
(A_{ri},B_{ri})	The value of the ECDSA signature for role *r*.
E_{pukx} *(M)/*E_{prkx} *(M)*	Encrypt or Decrypt Algorithm the message *M* by using the public key or private key party of *role r*, separately
s	The indicator of design system.
m	The indicator of design module.
ic	The indicator of design IC.

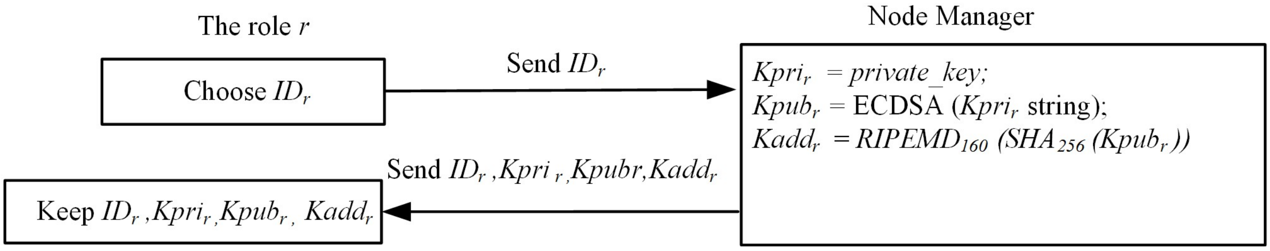

The role r
Node Manager
Choose ID_r
Send ID_r
$Kpri_r$ = $private\_key;$
$Kpub_r$ = ECDSA ($Kpri_r$ string);
$Kadd_r$ = $RIPEMD_{160}$ $(SHA_{256}$ $(Kpub_r))$
Send ID_r ,$Kpri$ $_r$,$Kpubr$,$Kadd_r$
Keep ID_r ,$Kpri_r$,$Kpub_r$, $Kadd_r$

Figure 3. The registration procedure for the participant, e.g., SM, MM, and ICM.

The detailed steps for the *Registration Procedure* are described below in this subsection.

Registration Procedure

Step 1. The notation ID_r refers to all roles (*r* roles) involved in QBN. The corresponding manufacturer then submits his product ID_r to the blockchain node, separately, where the blockchain node performs the verification of the legitimacy and then registers the identity to QBN.

Step 2. The node manager will then calculate the public key $Kpub_r$ and the private key $Kpri_r$ in the blockchain network.

$$Kpub_r = Kpri_r G$$

Step 3. After confirming that all participants have been already registered, the node manager will perform Algorithm 1. Each role *r* will then receive the generated $(ID_r, Kpri_r, Kpub_r)$ from the node manager.

Step 4. The role *r* in the system receives and saves the signature message parameters.

Next, the interaction relationship among DApp, IPFS, and QBN in the proposed framework is illustrated in Figure 4.

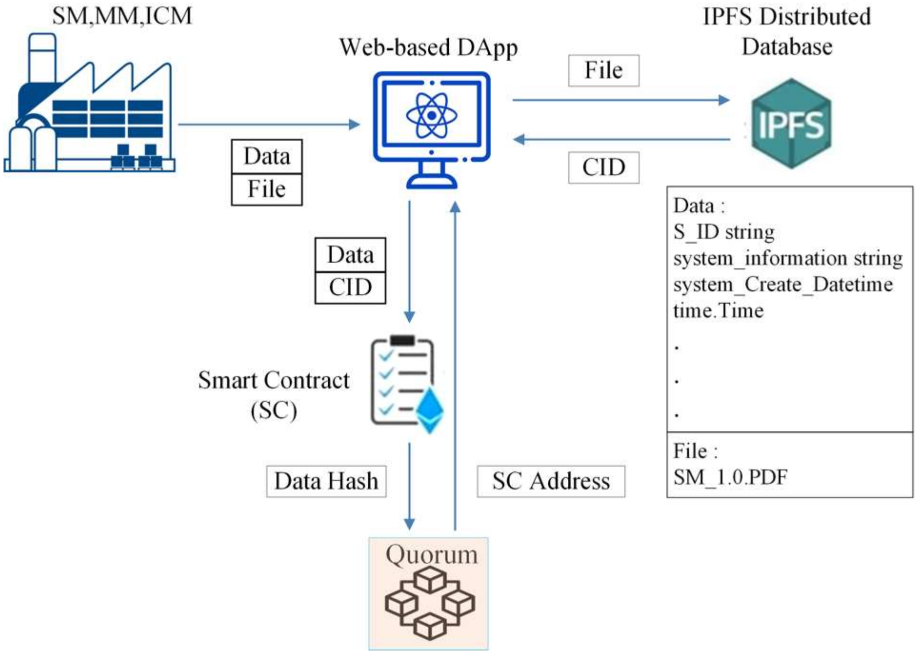


Figure 4. The interaction relationship among DApp, IPFS, and QBN in the proposed framework.

Algorithm 1: Registration to Node Manager.

```
Input: $ID_r$ together with its role r;
Output: $ID_r, Kpri_r, Kpub_r,\ Kaddr_r$;
$Kpri_r$ = private_key; //Node manager will assign a private key to identify
unique member by according to Equtaion (1);
$Kpub_r$ = ECDSA ($Kpri_r$ string); //According to Equtaion (2)
$Kaddr_r = RIPEMD_{160}(SHA_{256}\ (Kpub_r))$; //According to Equtaion (3)
Function Append ($ID_r$ string, r stiing, $Kpri_r$,            $Kpub_r$) {
return String "Successful Registration"; // It has been registerd to QBN
}
return $ID_r, Kpri_r, Kpub_r, Kaddr_r$
```

4.3. Production Phase

During the phase of making a new product, the participants involved are SM, MM, and ICM. Algorithms 2 and 3 show the details of these processes for the product design employed in the off-chain. Algorithm 4 is used to create a new system production. These algorithms consisting of Algorithms 2–4 are the main DApps proposed for this framework.

Algorithm 2: Ordering Function.

```
Input: s, m, ic, Order_Quantity, $kaddr_r$
Output: String notification
for (i: = 0; i < the number of product; i++) {
 struct details order {
 char s type [100];
 char m type [100];
  char ic type [100];
 Order_Quantity {s [int]; m [int]; ic [int];}
 char $kaddr_r$ [42];
  }
 }
 return String notification
```

Algorithm 3: Order Verification Function.

```
Input: s, m, ic, offering price, Order_Quantity, $kaddr_r$
Output: String notification, approved price
    varchar offering price;
    varchar approved price;
    int       Order_Quantity;
    if (offering price =="1") {
    approved price:= offering price × Order_Quantity;
    }
    else {
    result = "Fail", "Disagreement";
    }
    return String notification, approved price
```

Algorithm 4: New System.

```
Input: ID_SM, ID_MM, S_INF, Signature
Output: String notification
load current TP[]
for (i:=0; i<S_IDs.length; i++) {
 TP[S_ID [ i ]].system_Detail.system_manufacture=ID_SM
 TP[S_ID [ i ]].system_Detail.system_information=S_INF
 TP[S_ID [ i ]].system_Detail.create_datetime=Datetime.Now()
TP[S_ID [ i ]].system_Detail.module_manufacture=ID_MM
 TP[S_ID [ i ]].system_Detail.Signature = Signature
}
return String notification
```

4.4. The Main Processes for the Implement Platform

After completing the phases of a system design of SM, module design of MM, and IC design of ICM shown in Figure 5 from Step 1 to Step 4, the new IC production, new module production, and new system production are performed from Step 5 to 25, separately. Next, the three main processes for the implementation platform are proposed and presented in this section. The details of the all-party are necessary to realize the three proposed processes. The initial step will register participants in the consortium blockchain [45]. Next, the system, module, and IC will be registered separately in the private Quorum blockchain we created. Both Quorum blockchain and IPFS store every transaction as a transaction record.

IPFS generates the corresponding CID used for authentication throughout a device's lifetime. It allows traceability and proof of ownership without an explicit need for a trusted intermediary. Furthermore, the authorized parties could utilize the data indexed by CID on the blockchain to authenticate, track, and analyze the product. Figure 5 depicts our proposed approach using the created blockchain.

In the implemented web-based DApp, the CIDs with a hierarchical relationship will be organized by using Merkle DAG via the proposed hierarchical content identifier mechanism (HCIDM) for creating and maintaining the latest trust chain according to both Definition 1 and Definition 2 which are first proposed in this framework. Both Definition 1 and Definition 2 are shown below.

Definition 1. *The latest trust chain is designed via organizing the hierarchical CIDs below.* $\left\{CID_{IC_1}, CID_{IC_2}, \ldots, CID_{IC_\mu}\right\} \preceq \{CID_{module_1}, CID_{module_2}, \ldots, CID_{module_v}\} \preceq \{CID_{system_\chi}\}$ *with the latest version, where the notation "$\preceq$" is the hierarchical organization relationship indicated the trust chain used in this framework.*

Definition 2. *The latest trust chain consisting of the latest version plus the previous versions is also designed via organizing the hierarchical CIDs below.*

$$\left\{CID_{IC_1}, CID_{IC_2}, \ldots, CID_{IC_\mu}\right\}||\ldots||\left\{CID'_{IC_1}, CID'_{IC_2}, \ldots, CID'_{IC_\mu}\right\} \preceq \left\{CID_{module_1}, CID_{module_2}, \ldots, CID_{module_v}\right\}||\ldots||\left\{CID'_{module_1}, CID'_{module_2}, \ldots, CID'_{module_v}\right\} \preceq \left\{CID_{system_\chi}\right\}||\ldots||\left\{CID'_{system_\chi}\right\}$$

with the latest CIDs version together with the previous CIDs versions, if it is necessary, depending on the rules made by system manufacturer, where the notation "$\preceq$" is also the hierarchical organization relationship.

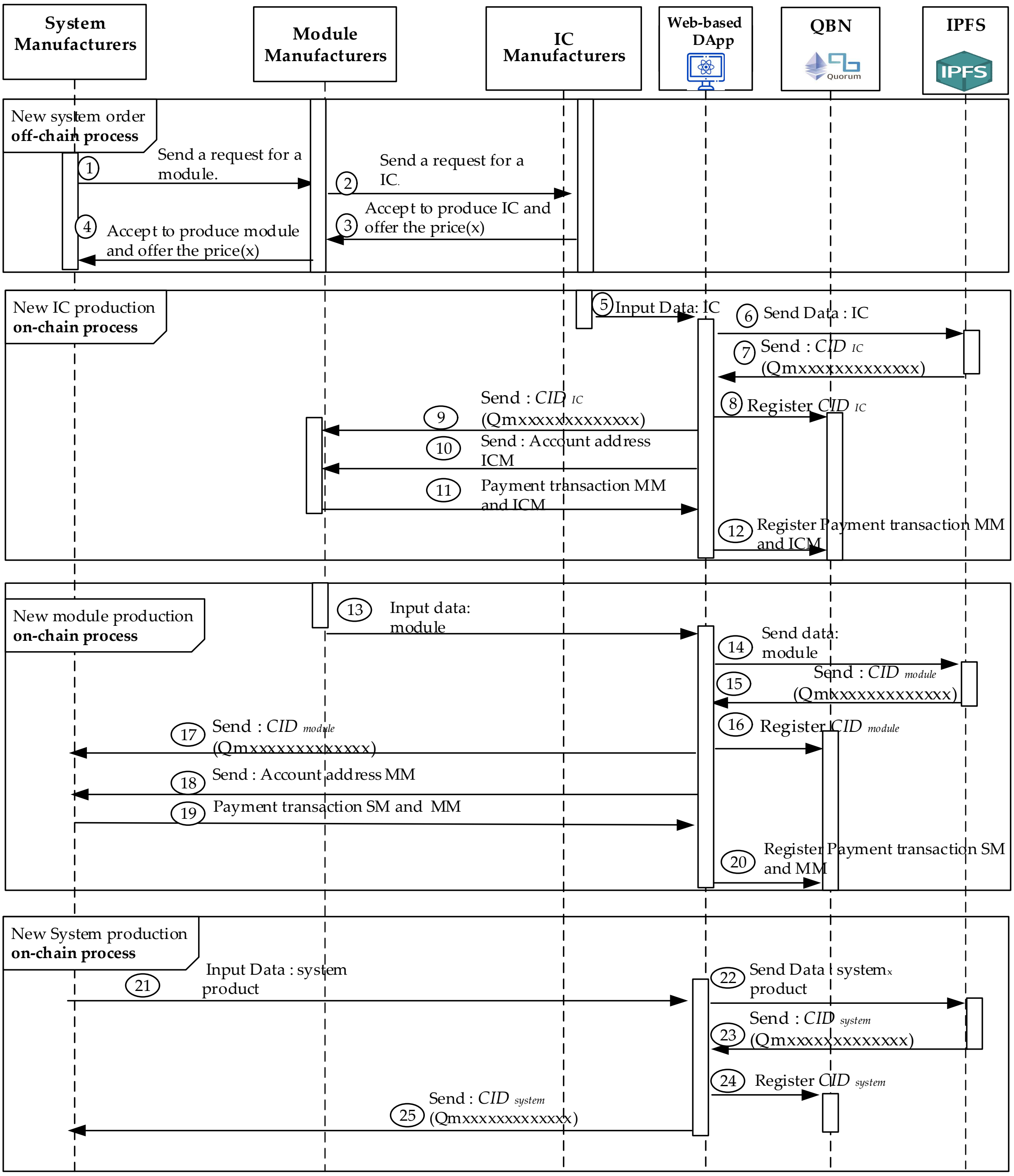

System Manufacturers
Module Manufacturers
IC Manufacturers
Web-based DApp
QBN
Quorum
IPFS
IPFS
New system order off-chain process
1 Send a request for a module.
2 Send a request for a IC.
3 Accept to produce IC and offer the price(x)
4 Accept to produce module and offer the price(x)
New IC production on-chain process
5 Input Data: IC
6 Send Data : IC
7 Send : CID IC (Qmxxxxxxxxxxxxxx)
8 Register CID IC
9 Send : CID IC (Qmxxxxxxxxxxxxxx)
10 Send : Account address ICM
11 Payment transaction MM and ICM
12 Register Payment transaction MM and ICM
New module production on-chain process
13 Input data: module
14 Send data: module
15 Send : CID module (Qmxxxxxxxxxxxxx)
16 Register CID module
17 Send : CID module (Qmxxxxxxxxxxxxx)
18 Send : Account address MM
19 Payment transaction SM and MM
20 Register Payment transaction SM and MM
New System production on-chain process
21 Input Data : system product
22 Send Data : systemχ product
23 Send : CID system (Qmxxxxxxxxxxxxxx)
24 Register CID system
25 Send : CID system (Qmxxxxxxxxxxxxx)

Figure 5. The sequence diagram of stakeholders in producing the system.

In addition, the web-based DApp is designed by using the smart contracts in this study. It could interact with the QBN together with the IPFS collaboration framework, where the HCIDM is first proposed and designed for each customized system product in order to create the system-level trust chain and easily validate or trace the trusted product versions of IC, module, and the customized system itself, comprehensively.

In the real application, each trust chain for each customized system product will be first created by the system manufacturer and stored in the database managed by the owner or buyer for this 'system product'. Finally. It could be accessed by using the web-based DApp issued by the system manufacturer and managed by the owner or buyer. For instance, the customized 'System 1' will be protected via the customized 'Trust Chain 1' shown in Figure 2. Next, the trust chain for each customized system product could be employed and co-work with the database managed by the owner's system administrator.

Therefore, this framework could be employed to trace and validate the current versions or previous versions of ICs, modules, or systems.

4.4.1. New System Producing Process

A blockchain consortium initiates the registration phase of each role in the scheme. QBN authenticates each participant, SM, MM, and ICM. After the authentication procedure is completed, each participant will get the public key, account addresses, and E-node ID sent by the node manager. The participants could communicate through a secured channel. Figure 5 illustrates its detailed procedure. The detailed steps of the production processes are shown in ***New System Producing Procedure*** for the new system.

New System Producing Procedure

Step 1. The chain will start when SM wants to create a new system, and SM first submits to MM the production certificate, module order, and production plan.

Step 2. MM will determine the manufacture of the module after receiving the order information from BC. The MM sends a command to the ICM about the required IC information.

Step 3. ICM replies to MM regarding producing the required IC at the agreed price.

Step 4. After receiving information from ICM, MM will provide the response requested by SM via sending the ability to produce the required module and the agreed price. From step 1 to step 4, transaction procedures are carried out off-chain in the order process by SM, MM, and ICM as consortium participants will perform Algorithms 2 and 3.

Step 5. ICM continues to decentralize data to IPFS network nodes after producing new ICs by inputting via DApp.

Step 6. DApp will forward the IC data that has been produced to IPFS to get the CID_{IC} hash value.

Step 7. Each piece of data is cryptographically hashed to be a secure unique CID_{IC} by IPFS. Next, it will be then passing it to DApp.

Step 8. The CID_{IC} obtained will be registered by ICM into QBN through a smart contract.

Step 9. The ICM sends the new CID_{IC} value for the completed IC to the MM via DApp.

Step 10. The ICM sends account addresses for payment transactions via DApp, where the account addresses are belonging to the account owner, ICM, himself.

Step 11. The MM makes payment transactions to an ICM via DApp.

Step 12. The smart contract we proposed does the transaction processes between ICM and MM regarding IC product, where the payments will be deployed and distributed in QBN.

Step 13. The DApp is used as an interface in order to input the module data. After all material requirements are complete, MM starts the module production. The completed module will be registered to IPFS in order to get the CID_{module} value for the new module.

Step 14. DApp automatically sends the data module to IPFS and generates the new module's CID_{module} value.

Step 15. IPFS sends CID_{module} data based on the last gathered data for this registered module.

Step 16. After the CID_{module} value is received, DApp will then automatically register and process it to the created QBN.

Step 17. The new CID_{module} value of the MM module will be sent to SM via DApp. The MM account address is sent to SM for payment transactions via DApp.

Step 18. SM sends payment transactions to ICM via DApp according to the received account address.

Step 19. The smart contract performs the transactions between MM and SM, which is regarding module payments. Additionally, it will be registered and distributed in QBN.

Step 20. The Smart contract transactions between MM and SM regarding module payments will be registered and distributed in QBN.

Step 21. SM starts to manufacture the new system when all the necessary modules are gathered completely. The new system will be registered its data into IPFS to get the corresponding CID_{System} value through the DApp.

Step 22. IPFS receives system-generated data sent by DApp, then generates the corresponding CID_{System} value for the new system and assigns it to SM.

Step 23. IPFS sends back CID_{System} data to SM.

Step 24. The DApp automatically registers the new system CID_{System} value consisting of the CID_{module} values of the assembled module together with the corresponding CID_{IC} values of IC into QBN. Finally, the CID values will be organized by Merkle DAG [55] and become the hierarchical CID values tree for the trust chain we proposed.

Step 25. Finally, the system data, only stored in IFPS, is registered successfully with the last version via recording its corresponding new CID_{System} value into the QBN performed by the DApp. Then, the node manager, SM, will perform *Algorithm 4*. The IPFS will deliver the corresponding CID_{System} value back to SM.

In addition, the DApp, together with the smart contracts, is designed and created by SM for constructing and managing the latest trust chain.

4.4.2. New Module Producing Process

MM could manufacture the modules with new versions without requiring an order from SM. Figure 6 shows the sequence in which MM generates a new module version and has the same functionality as the previous module. The new version of the module has additional features, which are the advantages of the original version. The detailed steps of the new module producing processes shown in ***New Module Producing Procedure*** for producing the new version module. The first two steps are used to process the order transactions, which involve the off-chain processed to obtain an MM agreement with ICM. Moreover, Step 3 and the remaining steps of this procedure are the on-chain processes for the QBN we created for these processes.

New Module Producing Procedure

Step 1. The first step starts when MM produces a new version of the module, MM will order the required number of ICs at ICM.

Step 2. After receiving the order of transaction, ICM will check its ability to produce the IC ordered by MM. If ICM can make the IC requested by MM, ICM will provide the price and the ability to produce the IC. Next, both the consortium participants, MM and ICM will perform Algorithms 2 and 3.

Step 3. ICM manufactures ICs according to MM orders of transactions. After production is complete, ICM will input IC data via DApp.

Step 4. DApp will send IC data automatically to IPFS in order to get the CID_{IC} value.

Step 5. IPFS performs the IC data calculation process in order to generate the CID_{IC} value and sends it back to DApp.

Step 6. Smart contracts we designed and deployed in QBN will register the CID_{IC} values via DApp, designed in this project, as intermediaries.

Step 7. DApp sends CID_{IC} Value back to MM generated by IPFS.
Step 8. DApp sends ICM account addresses to MM for receiving the cryptocurrency during the payment transactions while providing some ICs.
Step 9. MM makes payment transactions to ICM based on account addresses received via DApp.
Step 10. This DApp will register MM payment transactions to ICM by using the smart contract deployed in QBN.
Step 11. After producing the new version of the module, MM will input the new version data module via the DApp.
Step 12. The DApp sends the data entered by the MM to IPFS in order to get the CID_{module} value for the new version of the module.
Step 13. IPFS generates a new version of the data module and sends its corresponding CID_{module} value to the DApp.
Step 14. Finally, the module data, only stored in IFPS, is registered with the last version via recording its corresponding new CID_{module} value into the QBN performed by the DApp.
Step 15. The IPFS will deliver the corresponding CID_{module} value back to MM.

Moreover, the DApp together with the smart contracts, which are deployed and managed by SM, are performed in order to involve the information of the new module in the new trust chain.

4.4.3. New IC Chip Producing Process

ICM could also produce new versions of ICs without receiving some orders from MM or SM. Figure 7 illustrates the ICM participant in developing new versions of IC chips. The detailed steps of ***New IC Chip Producing Procedure*** are shown below.

New IC Chip Producing Procedure

Step 1. ICM registers the data of the produced IC chip with the last version to IPFS through the DApp.
Step 2. After receiving the registering IC data sent from ICM via DApp, IPFS will store them and generate a new and corresponding CID_{IC} hash value.
Step 3. The DApp will automatically register this corresponding CID_{IC} value using the deployed DApp smart with the specific contracts we designed into QBN.
Step 4. Finally, DApp will send back the generated CID_{IC} to ICM. Additionally, the IC product data, stored in IFPS, are also registered with the last version via recording its corresponding new CID_{IC} value into the QBN performed by the DApp and the related smart contracts.
Step 5. The IPFS will deliver the corresponding CID_{IC} value back to ICM.

Process transactions in a blockchain-enabled framework based on QBN show how each stakeholder involved in the blockchain quorum network interacts. ICM as an IC manufacturer, MM as a module manufacturer, and SM as a system interact to get the required information. The transactions indicate the blockchain node network authority, SM as a node manager, which will return the public and private keys and corresponding digital identity credentials.

Furthermore, the DApp together with the smart contracts, which are deployed and managed by SM, are performed in order to involve the new information of the IC chip into the new trust chain.

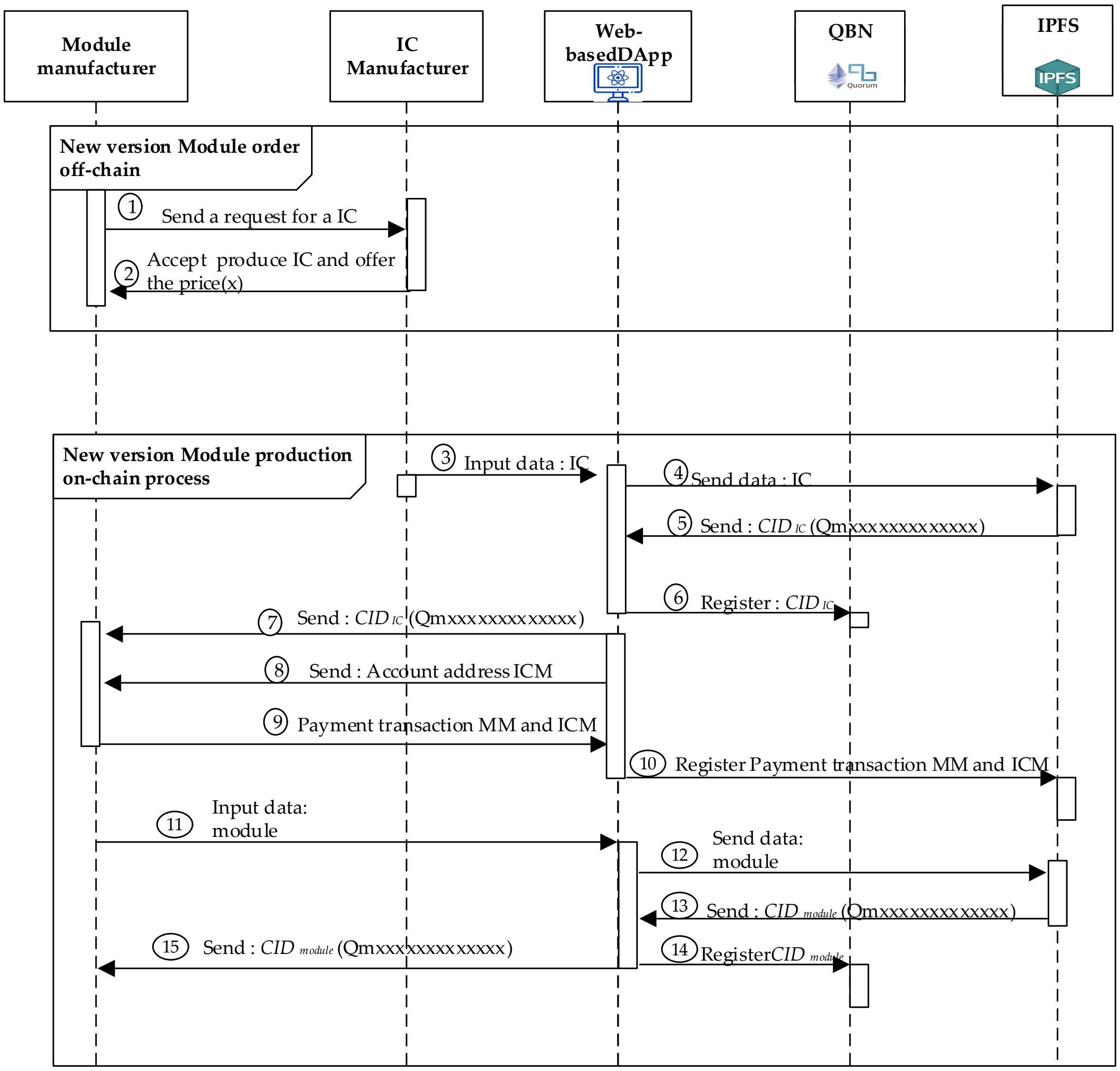

Module manufacturer
IC Manufacturer
Web-basedDApp
QBN
Quorum
IPFS
IPFS
New version Module order off-chain
1 Send a request for a IC
2 Accept produce IC and offer the price(x)
New version Module production on-chain process
3 Input data : IC
4 Send data : IC
5 Send : CID IC (Qmxxxxxxxxxxxxx)
6 Register : CID IC
7 Send : CID IC (Qmxxxxxxxxxxxxx)
8 Send : Account address ICM
9 Payment transaction MM and ICM
10 Register Payment transaction MM and ICM
11 Input data: module
12 Send data: module
13 Send : CID module (Qmxxxxxxxxxxxxx)
14 RegisterCID module
15 Send : CID module (Qmxxxxxxxxxxxxx)

Figure 6. The sequence diagram in producing the new version of module.

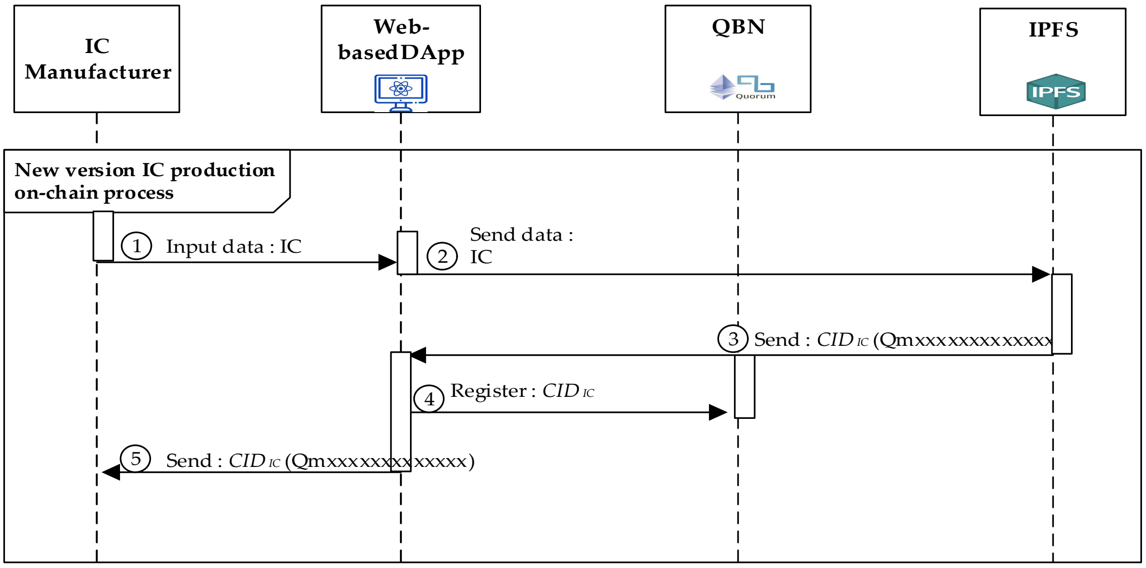

IC Manufacturer
Web-basedDApp
QBN
Quorum
IPFS
IPFS
New version IC production on-chain process
1 Input data : IC
2 Send data : IC
3 Send : CID IC (Qmxxxxxxxxxxxxx
4 Register : CID IC
5 Send : CID IC (Qmxxxxxxxxxxxxx)

Figure 7. The sequence diagram of producing the new version of IC.

5. Experimental Results

The proposed system is implemented using blockchain and IPFS technologies to be an in-depth framework for this work. The QBN holds a unique hash CID generated by stakeholders for each specific product and issued by IPFS. Solidity is a high-level programming language we use to write all smart contracts in this work. The Solidity language supports inheritance, and libraries can be imported and designed in an Ethereum Virtual Machines (EVM) environment. The resources of experiments for this study are adapted PowerEdge R710-Dell server, shown in Figure 8, to construct a consortium blockchain. Its specifications consist of 2.26 GHz 6 GB Intel® Xeon® 5000 Sequence 570 W 2.26 GHz, E5520, 6 GB, DDR3-SDRAM, 160 GB, and Ubuntu 20.04 x64. Furthermore, one node contains an Intel(R) Core (TM) i7-9700F processor running at 3.00 GHz dual OS Linux Ubuntu 20.04.3 and Windows 10. Remix, Truffle, and Visual Studio as support tools.

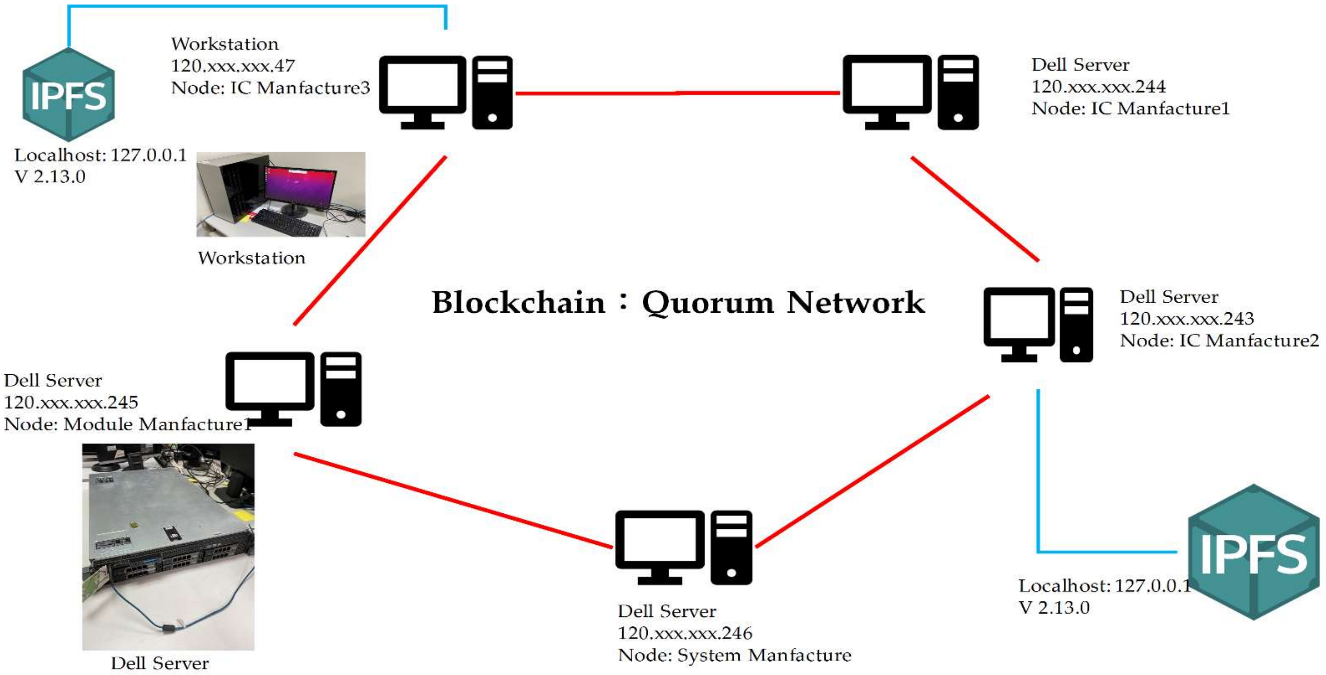


Figure 8. The PowerEdge R710-Dell server is applied to construct the consortium blockchain.

Additionally, the six actors are involved in the framework with the secure decentralized trust chain for the developed system. The roles and duties are described below.

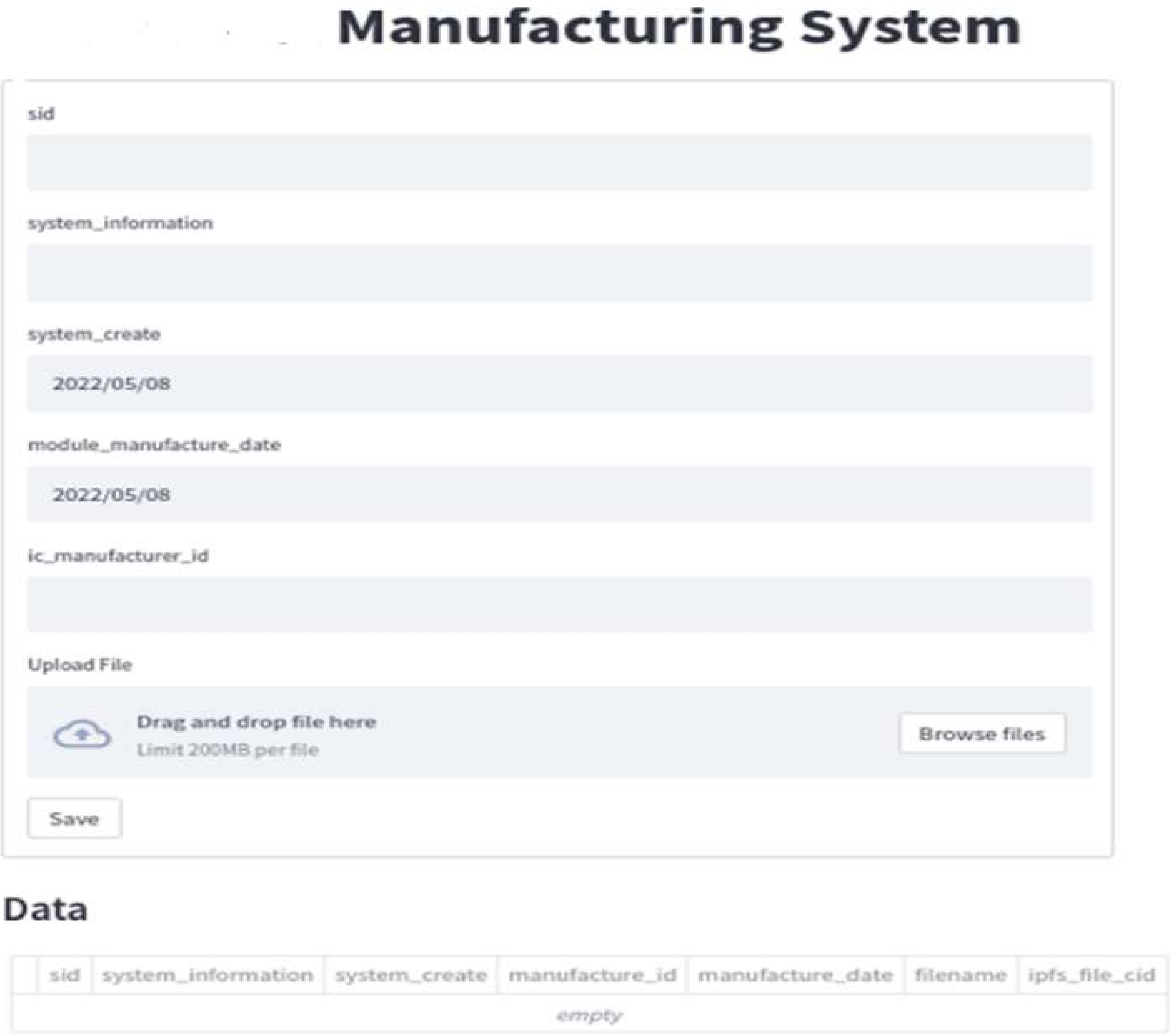
SM: This actor manages the design to create a new system. Production of the system requires a license from the system manufacturer. The chain only works when the system manufacturer decides to build a new system. Each part of the new system is assigned a unique identification code.

1. MM: This actor supplies the module material to manufacture the new system. SM will initiate communication with MM by sending the required manufacturing license and ordering information.
2. ICM: It will produce information regarding IC ordered by MM. ICM is responsible for converting raw materials into IC chips. The ICM will provide a unique identification code for each completed IC chip.
3. QBN: Determines the legitimacy of participants/nodes joining the blockchain architecture. Manage constantly increasing groups of records. Collections of transactions are grouped and allocated ledger chain blocks to blocks.
4. IPFS: This decentralized protocol provides secure data storage and a way to generate addressable hashes of uploaded files. The IPFS protocol allows synchronized data distribution. Implementing a distributed hash table (DHT) and a CID-based storage system.

5. DApp: It is a software program that uses smart contracts. Smart contracts could be accessed through DApp, which offers a convenient interface. The DApp that runs on a blockchain network is an example of a cryptocurrency application.

First, each participant, for example, SM, MM, and ICM, will register on QBN, individually. After registration processing, each participant will be assigned a private-public key pair according to Equations (1)–(3) in order to identify his or her unique member on the QBN.

After ICM completes IC production, ICM will create an IC chip, either based on the design order agreed with MM or a new version. ICM will also upload the complete IC data and the corresponding files via DApp, as shown in Figure 9. From IPFS, the CID with 46 characters beginning with the character "Qm" is automatically generated. Then, it will be registered on the blockchain by using the deployed smart contract.



Manufacturing System

sid

system_information

system_create

2022/05/08

module_manufacture_date

2022/05/08

ic_manufacturer_id

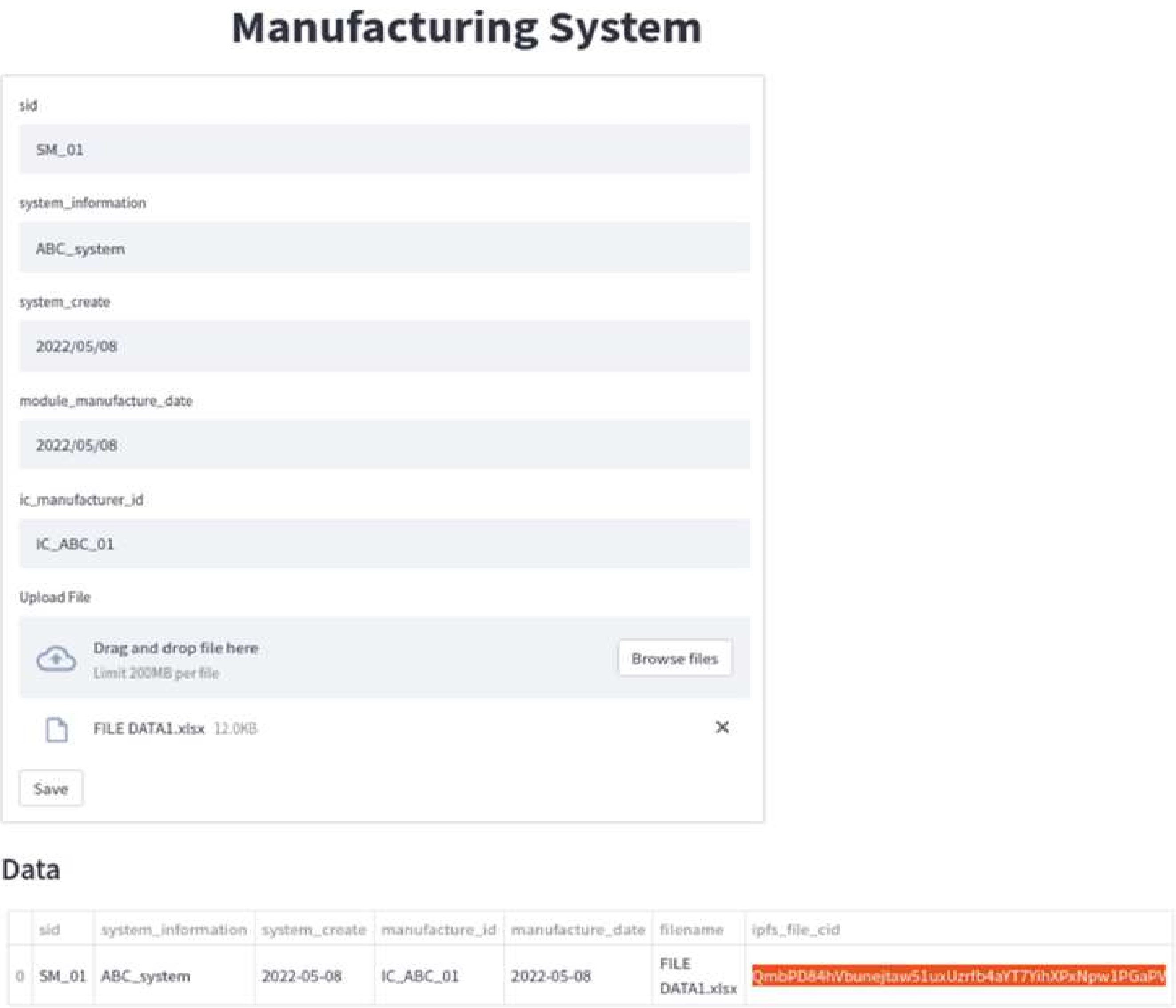
Upload File

Data

	sid	system_information	system_create	manufacture_id	manufacture_date	filename	ipfs_file_cid
				empty			

(**a**)

Figure 9. *Cont.*



Manufacturing System

sid

SM_01

system_information

ABC_system

system_create

2022/05/08

module_manufacture_date

2022/05/08

ic_manufacturer_id

IC_ABC_01

Upload File

FILE DATA1.xlsx 12.0KB

Data

	sid	system_information	system_create	manufacture_id	manufacture_date	filename	ipfs_file_cid
0	SM_01	ABC_system	2022-05-08	IC_ABC_01	2022-05-08	FILE DATA1.xlsx	QmbPD84hVbunejtaw51uxUzrfb4aYT7YihXPxNpw1PGaPV

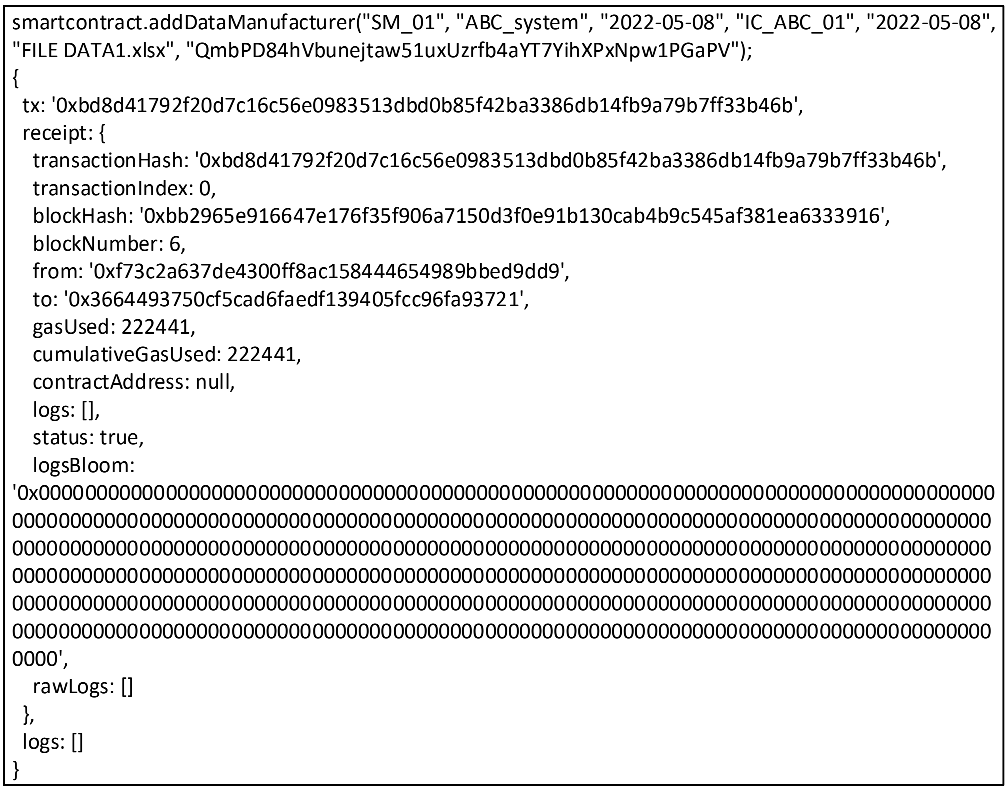
(**b**)

Figure 9. (**a**) DApp of SM (this SM could be called a new MS production); (**b**) DApp of SM with the input data.

After receiving the orders from SM, MM will start to produce the new module which should be fulfilling all the requirements including the requested ICs. MM will then register each generated data and the corresponding CID of this new module into QBN via the implemented smart contract. IPFS will also provide the CID value after the MM uploads the completed module file. MM uses DApp to interact with IPFS and QBN, as shown in Figure 9. Once all the required modules have been completed and accepted by SM with the specified specifications, SM will start producing a new system (called an "MS" system in this paper) or a new version of the existing system.

SM will immediately start recording data and uploading files via DApp after creating the product, as shown in Figure 9. IPFS will generate CID for files containing new system information or upgraded version.

QBN executes smart contracts, propagated via Web3 Application Programming Interface (API) to enter data and CID into blockchain with transaction hash result "0xbd8d41792-f20d7c16c56e0983513dbd0b85f42ba3386db14fb9a79b7ff33b46b" and with CID number "QmbPD84hVbunejtaw51uxUzrfb4aYT7YihXPxNpw1PgaPV," as seen in Figure 10. The files would be uploaded to IPFS, which could be in various forms, such as text, images, videos, etc. ICM sends data to MM and uploads to IPFS about built ICs, and MM does the same to support the creation of new systems. SM's final phase is to generate a new system and upload it to IPFS. The result of the SM file uploading is in the form of master data consisting of several modules incorporated into the new system. The module also consists of several integrated ICs, so that the data stored in IPFS by SM will be in the form of a multilevel hierarchy. All stakeholders could ensure that the new dataset system follows a pre-agreed design.



```
smartcontract.addDataManufacturer("SM_01", "ABC_system", "2022-05-08", "IC_ABC_01", "2022-05-08",
"FILE DATA1.xlsx", "QmbPD84hVbunejtaw51uxUzrfb4aYT7YihXPxNpw1PGaPV");
{
  tx: '0xbd8d41792f20d7c16c56e0983513dbd0b85f42ba3386db14fb9a79b7ff33b46b',
  receipt: {
    transactionHash: '0xbd8d41792f20d7c16c56e0983513dbd0b85f42ba3386db14fb9a79b7ff33b46b',
    transactionIndex: 0,
    blockHash: '0xbb2965e916647e176f35f906a7150d3f0e91b130cab4b9c545af381ea6333916',
    blockNumber: 6,
    from: '0xf73c2a637de4300ff8ac158444654989bbed9dd9',
    to: '0x3664493750cf5cad6faedf139405fcc96fa93721',
    gasUsed: 222441,
    cumulativeGasUsed: 222441,
    contractAddress: null,
    logs: [],
    status: true,
    logsBloom:
'0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000',
    rawLogs: []
  },
  logs: []
}
```

Figure 10. Example of transaction Hash.

Public, private, or consortium blockchains could be been categorized according to their real applications and requirements. In general, both public blockchain and consortium blockchain are usually used for public networks. A public blockchain is a permissionless, non-restrictive, distributed ledger system, meaning that anyone with internet access right could join and participate in a blockchain network. The public blockchain maintains the confidence of the entire users' communities because everyone in the network feels motivated to contribute to the improvement of the public network. However, it suffers from both scalability and time-consuming issues. On the other hand, a consortium blockchain is best suited for organizations that need to implement both private and public systems. In this type, there are multiple central authorities or multiple organizations that provide access to pre-selected nodes for reading, writing, and auditing the blockchain. It maintains its decentralized nature because no single authority governs the control. The cost of production is also relatively lower than the previous public systems. Thus, this section compares the public blockchain represented by the Rinkeby test network [41] with the consortium blockchain [45] represented by Quorum.

Furthermore, the cost of operating our system is computed with minor state changes, execution moves, or storage space utilization, and we decide to pay miners in Ether. In Ethereum, gas is the measurement for the cost of the execution task. When the user chooses to change the status of such a smart contract, the user must pay for the corresponding state modification gas. Different program operations require different amounts of gas to run. Gas has a fixed operating cost. For example, each contracting process costs 164,175 Wei gas on block 1, and the cost of a contract call on block 2 is 42,431 Wei gas. The charge for each block can be seen in Table 4. On the public blockchain, users can set gas prices for any Wei. However, the higher the gas price, the faster transactions are completed within a block because miners prefer higher-priced transactions. The relationship between both acceptance speed and gas price among the transactions, which is shown in Table 5. The gas price on the consortium blockchain is "0". EthGasStation is an open-source initiative designed to increase gas price transparency [45]. To pay the smart contract's procedure cost using Ether, first convert it to US dollars. Navigate the Foreign Exchange (FX) rates website (see https://fx-rate.net/ETH/USD/) to obtain the current exchange.

Table 4. The comparison of transaction costs between Truffle and QBN operation using 12 blocks.

Transaction	Gas Used	Gas Price		Transaction Cost		Block No.
		Public Blockchain [41]	Consortium Blockchain [45]	Public Blockchain [41]	Consortium Blockchain [45]	
No. transaction	0	0.000042 Eth	0 Eth	0	0	0
Contract Creation	164,175	0.000042 Eth	0 Eth	6.89535	0	1
Contract Call	42,431	0.000042 Eth	0 Eth	1.782102	0	2
Contract Creation	96,189	0.000042 Eth	0 Eth	4.039938	0	3
Contract Call	27,341	0.000042 Eth	0 Eth	1.148322	0	4
Contract Creation	222,254	0.000042 Eth	0 Eth	9.334668	0	5
Contract Call	27,341	0.000042 Eth	0 Eth	1.148322	0	6
Contract Call	84,015	0.000042 Eth	0 Eth	3.52863	0	7
Contract Call	26,415	0.000042 Eth	0 Eth	1.10943	0	8
Contract Call	34,815	0.000042 Eth	0 Eth	1.46223	0	9
Contract Call	38,415	0.000042 Eth	0 Eth	1.61343	0	10
Contract Call	34,815	0.000042 Eth	0 Eth	1.46223	0	11

Table 5. The relationship of both acceptance speed and gas price among the transactions.

Transaction Speed	Gas Price
Very likely in <15 s, high	Max fee:(0.000042 ETH)
Likely in <30 s, medium	Max fee:(0.0000315 ETH)
Maybe in 30 s, low	Max fee:(0.00002961 ETH)

When experimenting with the scheme, the ether exchange rate was referenced by the Foreign Exchange (FX) website (also referred to on https://fx-rate.net/ETH/USD/ accessed on 14 June 2022); the ether exchange rate reached a high of USD 3810.525 per ether and a low of USD 1,124.493. Furthermore, the median of the highest and lowest, USD 2,726.836, will be the benchmark in calculating costs in our system. Multiplying the gas required to perform the appropriate function by the current value of the ether, we get the price. Utilize Remix to determine the amount of gas required to run our system's functionalities. The Remix is a web-based integrated development environment (IDE) for developers creating Solidity DApps. The output of the system function execution is shown in Table 6. The evaluation results show that the proposed scheme could work in practice under the much lower costs, compared to the public blockchain, with a total cost o "0.002094" in Ether. According to the highest, the lowest, and the average dollar price as the reference (https://fx-rate.net/ETH/USD/ accessed on 14 June 2022) the most increased initial production cost for the recorded contract is USD 7.97922888. The lowest initial cost is USD 2.354688342, and the average price for each production is USD 5.709994584.

Table 6. The gas cost in executing functions in the proposed framework.

Costs The (Description of Functions)	Gas Used (Units)	Gas Price: (Gwei)	Gas Fee (Gwei)	Gas Fee (Ether)	Total Cost (US Dollar)
Deploy_the_contract	608,245	2.500000866	1,520,613.027	0.001521	1.85
Send_account	43,836	2.500000866	109,590.038	0.00011	0.13
Send_IPFS_cid	90,028	2.500000866	225,070.078	0.000225	0.27
Received_address	68,365	2.500000866	170,912.5592	0.000171	0.21
Send_address	26,760	2.500000866	66,900.02317	0.000067	0.08
	Total: 837,234		Total: 2,093,085.725	Total: 0.002094	Total: 2.54

6. Discussions and Analyses

The MS with the trust chain of IC-module-system enjoys valuable benefits with the advances in blockchain technology and distributed data processing in terms of flexibility, traceability, and security compared with centralized MS. The proposed framework could manage ubiquitous transactions between manufacturers and communicate with manufacturers to form a trusted chain network that supports more personalized products following ever-changing trends. From a technology point of view, the permissioned blockchain could improve network protection against cyber-attacks with an additional control layer given the drawback of current MS, inviting many parties to join the network, and making direct contact with enterprises. This type of secured blockchain technology also could achieve competitive throughput with low latency. As evident in [38,40,48], the effectiveness of the permissioned blockchain outperforms permissionless blockchain, that is, public Ethereum as a comparison. This study proposes verifiable practices for integrating the permissioned blockchain technology to support tangible insights in applying MS using cases in manufacturing industries. Several instances in manufacturing industries involving ICM, MM, and SM are described by integrating QBN solutions. Additionally, the smart contract and consensus mechanism used in the QBN system significantly impacts the solution's efficiency. The performance of the proposed method, how costly the implementation, and the energy consumption would be specific considerations for the enterprises to switch to using these technologies.

Due to the complexity of transactions, product traceability, and product counterfeiting, the manufacturing industry faces many challenges. This problem can be solved by using blockchain technology to manage transaction security, and IPFS stores large amounts of distributed data. Transactions in the manufacturing industry are very complicated because of the information, such as product data, payment transactions, and product ownership. A secure chain of trust is needed in the production process to ensure product authenticity and traceability, QBN development with IPFS can be realized efficiently to handle the complexities of MS in the real world. Thus, the application in designing the proposed system framework could answer the above challenges. A secure trust chain built hierarchically from the initial phase of production to the final phase could be a solution to the problems faced in the manufacturing industry.

These three cases, Case 1, 2 and 3, are discussed and analyzed below.

Case 1: IC-module-system data in the trust chain.

Merkle DAG [55] is utilized as a data object model for HCIDM, which is analogous to a Merkle tree proposed by Ralph C. Merkle [54]. A structure comprising two properties, Data, and Links, constitutes an IPFS object. Each Link structure includes the Name, Hash, and Size characteristics. Thus, HCIDM could assemble things and construct a directed acyclic graph by utilizing this object structure. Figure 11 depicts how HCIDM employed Merkle DAG to arrange the construction of a file or file directory in a web-based DApp. Figure 11 describes the HCIDM used in the proposed scheme, where the structure of HCIDs consists of three layers, the system manufacturer layer, the module manufacturer layer, and the IC manufacturer layer. In the IC manufacturer layer, it consists of leaf nodes where each leaf node contains a file name and IC data. Each leaf node has a CID hash value, for example, $CID_{address_{IC_x}} \in \left\{ CID_{address_{IC_1}}, CID_{address_{IC_2}}, \ldots, CID_{address_{IC_n}} \right\}$, $CID'_{address_{IC_x}} \in \left\{ CID'_{address_{IC_1}}, CID'_{address_{IC_2}}, \ldots, CID'_{address_{IC_n}} \right\}$ or $CID''_{address_{IC_x}} \in \left\{ CID''_{address_{IC_1}}, CID''_{address_{IC_2}}, \ldots, CID''_{address_{IC_n}} \right\}$. Each leaf node has a different path to its parent node, the node module.

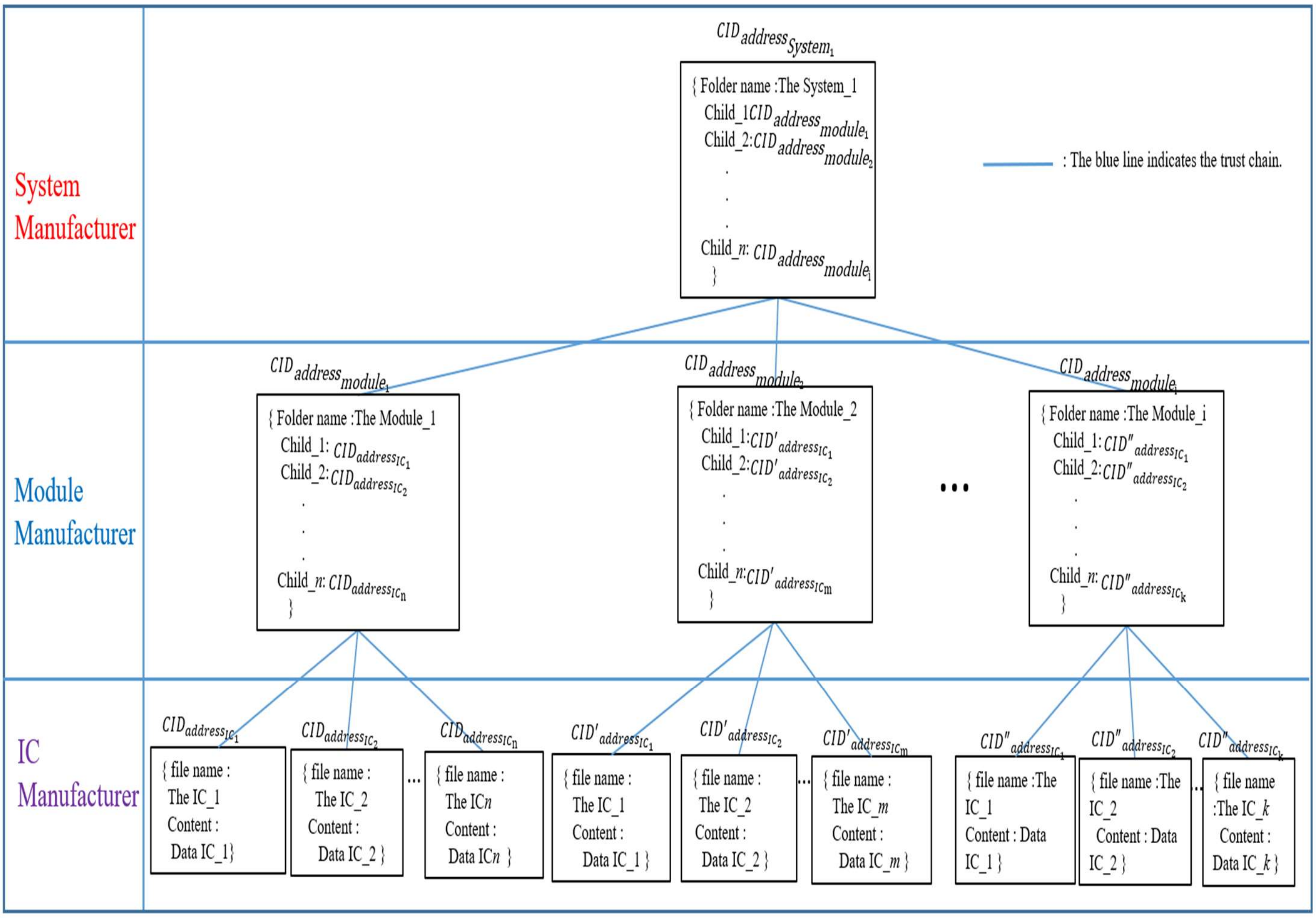

System Manufacturer
Module Manufacturer
IC Manufacturer
$CID_{address_{System_1}}$
{ Folder name :The System_1
Child_1$CID_{address_{module_1}}$
Child_2:$CID_{address_{module_2}}$
Child_n: $CID_{address_{module_i}}$ }
: The blue line indicates the trust chain.
$CID_{address_{module_1}}$
{ Folder name :The Module_1
Child_1: $CID_{address_{IC_1}}$
Child_2:$CID_{address_{IC_2}}$
Child_n: $CID_{address_{IC_n}}$ }
$CID_{address_{module_2}}$
{ Folder name :The Module_2
Child_1:$CID'_{address_{IC_1}}$
Child_2:$CID'_{address_{IC_2}}$
Child_n:$CID'_{address_{IC_m}}$ }
$CID_{address_{module_i}}$
{ Folder name :The Module_i
Child_1: $CID''_{address_{IC_1}}$
Child_2:$CID''_{address_{IC_2}}$
Child_n: $CID''_{address_{IC_k}}$ }
$CID_{address_{IC_1}}$
{ file name : The IC_1 Content : Data IC_1}
$CID_{address_{IC_2}}$
{ file name : The IC_2 Content : Data IC_2 }
$CID_{address_{IC_n}}$
{ file name : The IC*n* Content : Data IC*n* }
$CID'_{address_{IC_1}}$
{ file name : The IC_1 Content : Data IC_1 }
$CID'_{address_{IC_2}}$
{ file name : The IC_2 Content : Data IC_2 }
$CID'_{address_{IC_m}}$
{ file name : The IC_*m* Content : Data IC_*m* }
$CID''_{address_{IC_1}}$
{ file name :The IC_1 Content : Data IC_1 }
$CID''_{address_{IC_2}}$
{ file name :The IC_2 Content : Data IC_2 }
$CID''_{address_{IC_k}}$
{ file name :The IC_*k* Content : Data IC_*k* }

Figure 11. Merkle DAG in HCIDM performed in web-based DApp is activated and organized the trust chain in our framework on when any version data of IC, module or system is recorded.

The module manufacturer layer consists of several module nodes, where each module node contains the module folder file name and the child's CID hash value. Each module node has a CID hash value at the same layer, e.g., $CID_{address_{module_x}} \in \left\{ CID_{address_{module_1}}, CID_{address_{module_2}}, \ldots, CID_{address_{module_n}} \right\}$.

At the system manufacturer layer, the customized product, system_1, is the top parent node in the Merkle DAG tree. The top node system_1 contains the folder name and all CID hash values of the modules and ICs via his children links to each module node. The result forms HCIDs of interrelated directories and subdirectories, which are used to create a system-level trust chain.

Case 2: Update the IC-module-system data in the trust chain.

How to update any version data of IC, module, or system is an additional essential issue that HCIDM has to address. Each IPFS file has an associated hash address. When the IPFS stored file is modified, the hash address will be changed, too. Whenever any new version data of IC, module, or system is added to an IPFS network, it will be assigned a new CID which is a unique hash value of the new version data of IC, module, or system.

In traditional IPFS networks, it is identified and referenced in this way. Recalculating the hash when retrieving the file will verify its integrity. Modified files will fail the verification. When a file is legally modified, IPFS takes care of versioning the file. It indicates that the updated version of the file is stored with the new CID, and the previous version of the old file could be kept and retrieved via the previous CID. Figure 12 describes how to update the information for the updated IC version by using Merkle DAG employed by HCIDM. The first step is to create a Mutable File System (MFS) directory [43]. In Case 2, if anyone wants to add or update any version of IV, module, or system in this HCID tree, the hash values of the leaf nodes and subdirectory nodes regarding the related path will be changed, too. In this Case 2, the ICM makes IC_1 with a new version. Then the IC_1 with the new version will get a new CID ($CID_{address_{IC_{1\_NV}}}$). The module node in Figure 12, which is the parent of the IC node with the new IC_1 version, will update the link directory by updating a new CID value retrieved by IPFS implemented in this framework. After updating all the related files whose links are belonging to updating the Merkle DAG path, for example, $\left\langle CID_{address_{IC_{1\_NV}}} \leftrightarrow CID_{address_{module_{1\_NV}}} \leftrightarrow CID_{address_{system_{1\_NV}}} \right\rangle$, the parent module will get a new CID hash value $CID_{address_{module_{1\_NV}}}$. The system_1 node in the system build layer will also update the directory by updating the CID for the updated module $CID_{address_{module_{1\_NV}}}$, then the system_1 node will get the new hash value $CID_{address_{system_{1\_NV}}}$. With this procedure, the trust chain could be updated, via Merkle DAG employed by HCIDM, to be the new latest trust chain according to Definition 1 or Definition 2.

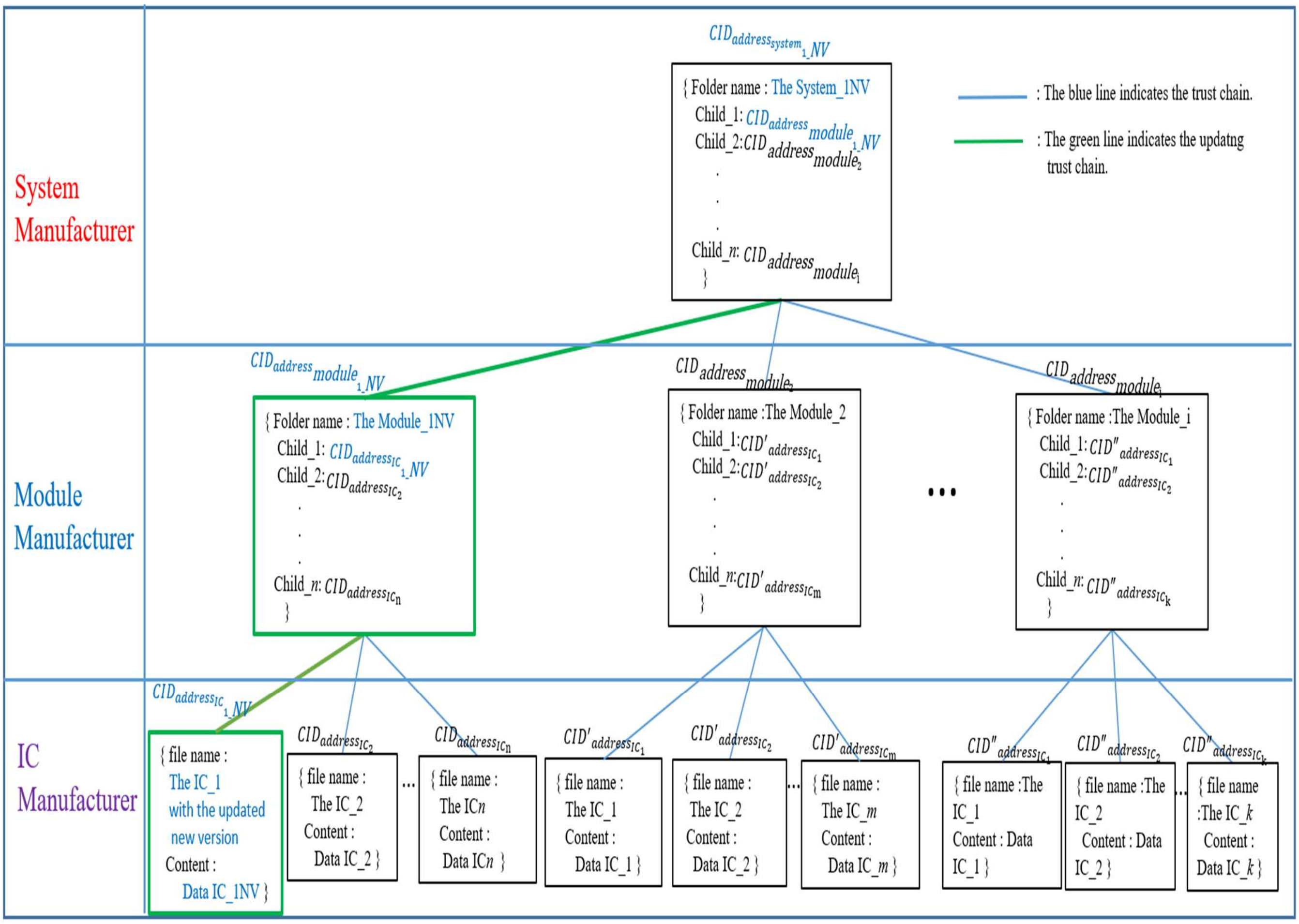

System Manufacturer
Module Manufacturer
IC Manufacturer
: The blue line indicates the trust chain.
: The green line indicates the updatng trust chain.
CID address system 1_NV
{ Folder name : The System_1NV
Child_1: CID address module 1_NV
Child_2: CID address module 2
Child_n: CID address module i
}
CID address module 1_NV
{ Folder name : The Module_1NV
Child_1: CID address IC 1_NV
Child_2: CID address IC 2
Child_n: CID address IC n
}
CID address module 2
{ Folder name :The Module_2
Child_1: CID' address IC 1
Child_2: CID' address IC 2
Child_n: CID' address IC m
}
CID address module i
{ Folder name :The Module_i
Child_1: CID" address IC 1
Child_2: CID" address IC 2
Child_n: CID" address IC k
}
CID address IC 1_NV
{ file name : The IC_1 with the updated new version Content : Data IC_1NV }
CID address IC 2
{ file name : The IC_2 Content : Data IC_2 }
CID address IC n
{ file name : The ICn Content : Data ICn }
CID' address IC 1
{ file name : The IC_1 Content : Data IC_1 }
CID' address IC 2
{ file name : The IC_2 Content : Data IC_2 }
CID' address IC m
{ file name : The IC_m Content : Data IC_m }
CID" address IC 1
{ file name :The IC_1 Content : Data IC_1 }
CID" address IC 2
{ file name :The IC_2 Content : Data IC_2 }
CID" address IC k
{ file name :The IC_k Content : Data IC_k }

Figure 12. Merkle DAG in HCIDM performed in web-based DApp is activated and re-organized the trust chain in our framework when any version of IC, module, or system is updated.

Case 3: Delete the IC-module-system data on the chain of trust.

The HCIDM preserves the history of computing devices by keeping each stored file version and building a solid network for data mirroring. Each participant, ICM, MM, or SM uses the web-based DApp, based on HCIDM, to automatically store, update and manage the version data of their product. Figure 13 describes how a version data file is deleted in the ICM leaf node. ICM will carry out the garbage collection process both manually and automatically. In this case, the file to be deleted has a CID hash value ($CID_{address_{IC_1}}{}^{*}$). ICM will delete the version data file regarding an old IC chip in order to revoke it. ICM will then publish the revoked information to the MM who is his parent node. Next, MM will update the file in this IPFS in order to get a new CID hash value $CID_{address_{module_1}}{}^{*}$. After getting updated information from the associated child module(s), SM will also update the contents in his folder shown in Figure 13 and get the new CID hash value $CID_{address_{system_1}}{}^{*}$. With this procedure, the trust chain could be updated to be the new latest trust chain according to Definition 1 or Definition 2. By carrying out this procedure, the security for the trust chain could also remain secure and solid.

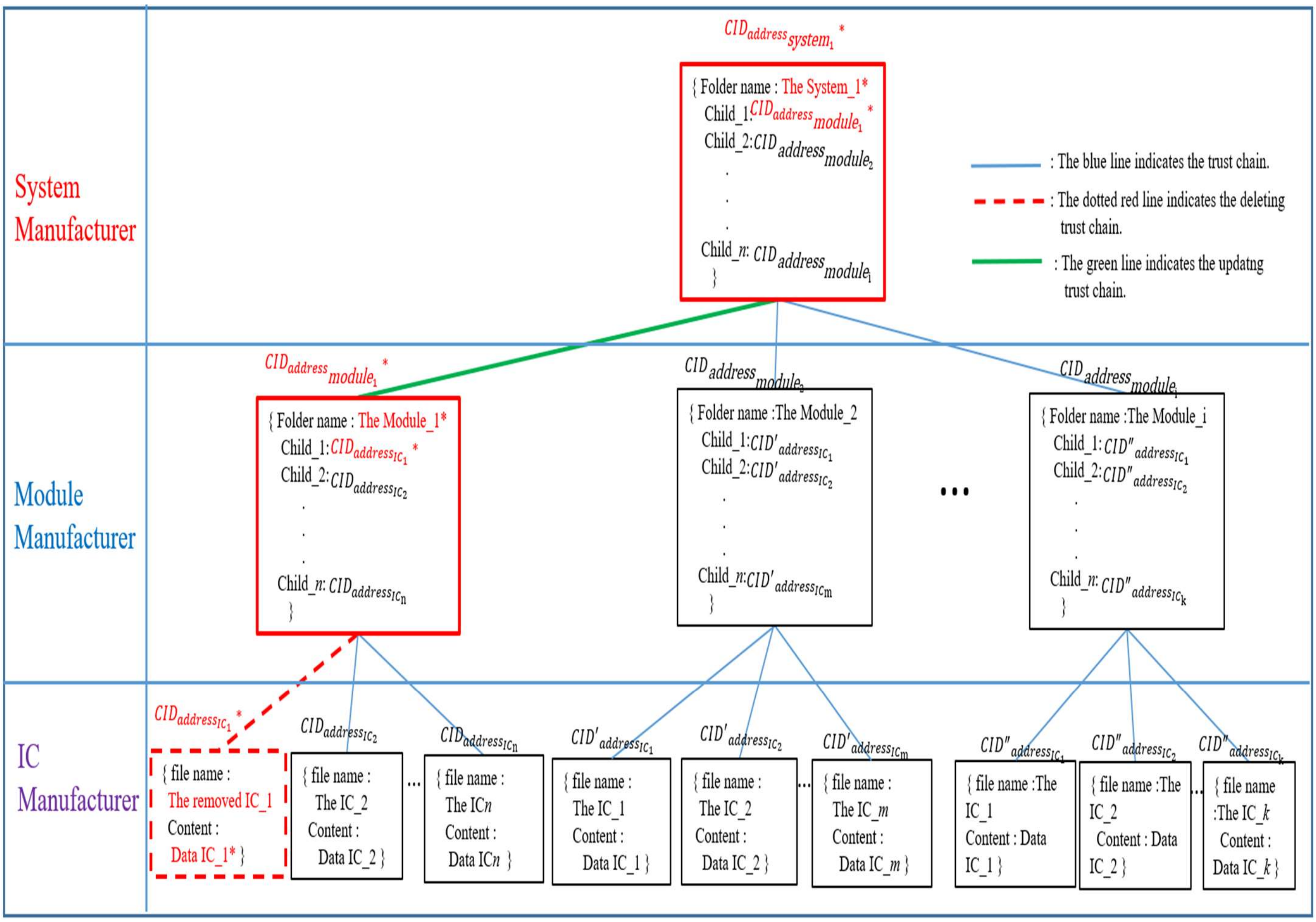

System Manufacturer
Module Manufacturer
IC Manufacturer
$CID_{address_{system_1}}{}^{*}$
{ Folder name : The System_1*
Child_1: $CID_{address_{module_1}}{}^{*}$
Child_2: $CID_{address_{module_2}}$
Child_n: $CID_{address_{module_i}}$ }
: The blue line indicates the trust chain.
: The dotted red line indicates the deleting trust chain.
: The green line indicates the updatng trust chain.
$CID_{address_{module_1}}{}^{*}$
{ Folder name : The Module_1*
Child_1: $CID_{address_{IC_1}}{}^{*}$
Child_2: $CID_{address_{IC_2}}$
Child_n: $CID_{address_{IC_n}}$ }
$CID_{address_{module_2}}$
{ Folder name :The Module_2
Child_1: $CID'_{address_{IC_1}}$
Child_2: $CID'_{address_{IC_2}}$
Child_n: $CID'_{address_{IC_m}}$ }
$CID_{address_{module_i}}$
{ Folder name :The Module_i
Child_1: $CID''_{address_{IC_1}}$
Child_2: $CID''_{address_{IC_2}}$
Child_n: $CID''_{address_{IC_k}}$ }
$CID_{address_{IC_1}}{}^{*}$
{ file name : The removed IC_1 Content : Data IC_1* }
$CID_{address_{IC_2}}$
{ file name : The IC_2 Content : Data IC_2 }
$CID_{address_{IC_n}}$
{ file name : The IC*n* Content : Data IC*n* }
$CID'_{address_{IC_1}}$
{ file name : The IC_1 Content : Data IC_1 }
$CID'_{address_{IC_2}}$
{ file name : The IC_2 Content : Data IC_2 }
$CID'_{address_{IC_m}}$
{ file name : The IC_*m* Content : Data IC_*m* }
$CID''_{address_{IC_1}}$
{ file name :The IC_1 Content : Data IC_1 }
$CID''_{address_{IC_2}}$
{ file name :The IC_2 Content : Data IC_2 }
$CID''_{address_{IC_k}}$
{ file name :The IC_*k* Content : Data IC_*k* }

Figure 13. Merkle DAG in HCIDM performed in web-based DApp is activated and re-organized the trust chain in our framework on when IC_1 data is deleted, where the symbol "*" is used to indicate the version of CID address (or said hash value) has been undated.

7. Conclusions

The proposed framework with a comprehensive identification mechanism is implemented in this paper. It integrates the three latest advances in computer engineering, including the permissioned blockchain, DApp, and distributed database mechanism, in order to implement the comprehensive identification mechanism with the trust chain of IC-module-system for MS toward secure Industry 4.0. The results show that it is flexible and traceable, enabling current MS with a comprehensive identification mechanism for achieving a more secure and reliable MS. This study emphasized using system architecture and proof-of-concept algorithms by integrating several components, including the permissioned blockchain network, smart contracts, and a consensus mechanism. In addition, to validate the proposed architecture, simulations were carried out using the QBN, DApps, and IPFS database. The results also revealed significant contributions. Firstly, each object in the current MS connected to a centralized system makes the system hard to scale, trace, and monitor. A decentralized system provides a more scalable network achieving a more flexible implementation while maintaining security issues. Secondly, the proposed method works at a much lower price compared to the public blockchain. Moreover, the highest initial production cost for the recorded contract is USD 7.97922888, the lowest initial price is USD 2.354688342, and the average cost for each production is USD 5.709994584, where they were calculated on 14 June 2022, via accessing the referred website on https://fx-rate.net/ETH/USD/. Thus, its implementation to face the challenges of current MS is feasible. Finally, blockchain technology, decentralized database, and DApp offer the practical concept to improve the current MS concept. However, there are several limitations in this work such as market transparency and verifiability. Based on insights obtained from this analysis, the trust chain based on HCIDM can be applied to any MS system, for example, this trust chain could be used to prevent the counterfeit modules and ICs employed in the monitoring system of semiconductor factory environment. Finally, this research is developed an innovation trust chain mechanism which could be provided the system-level security for any MS toward Industrial 4.0 in order to meet the requirements of both manufacturing innovation and product innovation in Sustainable Development Goals (SDGs). In future work, this study will enhance current efforts on extending the proposed system to achieve a secure end-to-end transparent and verifiable supply chain.

Author Contributions: Conceptualization, H.-C.C. and B.I.; Methodology, H.-C.C., B.I. and C.-W.L.; Software, B.I., Y.-H.L., J.-S.S. and P.; Validation, H.-C.C. and B.I.; Formal analysis, B.I. and K.T.P.; Investigation, H.-C.C. and B.I.; Resources, B.I., P.-Y.H., C.-W.L. and J.-S.S.; Data curation, B.I. and J.-S.S.; Writing—original draft preparation, H.-C.C. and B.I.; Writing—review and Editing, H.-C.C., C.-E.W. and C.D.; Supervision, H.-C.C.; Project administration, H.-C.C., C.-E.W. and P.-Y.H.; Funding acquisition, H.-C.C., P.-Y.H. and C.-E.W. Server resources supports, P.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Ministry of Science and Technology (MOST), Taiwan, under MOST grant numbers: 111-2218-E-468-001-MBK, 110-2218-E-468-001-MBK, 110-2221-E-468-007, and 110-2218-E-002-044.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Ministry of Science and Technology (MOST), Taiwan, under MOST grant numbers: 111-2218-E-468-001-MBK, 110-2218-E-468-001-MBK, 110-2221-E-468-007, and 110-2218-E-002-044. This work was also supported in part by the Ministry of Education under grant number. I109MD040. This work was supported in part by Asia University, Taiwan, and China Medical University Hospital, China Medical University, Taiwan, under grant numbers: ASIA-110-CMUH-22, ASIA108-CMUH-05, ASIA-106-CMUH-04, and ASIA-105-CMUH-04. This work was also supported in part by Asia University, Taiwan, UMY, Indonesian, under grant no. 107-ASIA-UMY-02. In addition, the server, PowerEdge R710-Dell, is supported by Pi-Hsien Chang who works for Information Management Center, Taichung City Government, Taiwan. Database design approach and its implementation are supported by Po-Jui Chung.

Conflicts of Interest: This article can be counted as the KPI record of Ph.D. degree for only the first student author Bambang Irawan; the KPI research result of the four projects, 111-2218-E-468-001-MBK, 110-2218-E-468-001-MBK, 110-2221-E-468-007, and 110-2218-E-002-044, supported by the Ministry of

Science and Technology (MOST), Taiwan, for the first and corresponding author Prof. Hsing-Chung Chen. Moreover, the authors declare no other conflict of interest.

References

1. BISWorld. Industry Market Research, Reports, and Statistics. Available online: https://www.ibisworld.com/industry-statistics/market-size/manufacturing-united-states (accessed on 1 November 2021).
2. BBC. Why Is There a Chip Shortage for Computers and Cars? Available online: https://www.bbc.com/news/technology-55936011 (accessed on 2 November 2021).
3. Reuters. South Korea Factory Output Falls at Fastest Pace since May 2020 on Auto Chip Shortage. Available online: https://www.reuters.com/markets/asia/skorea-factory-output-falls-fastest-pace-since-may-2020-auto-chip-shortage-2021-11-29/ (accessed on 2 November 2021).
4. Liu, C.; Tang, D.; Zhu, H.; Nie, Q. A Novel Predictive Maintenance Method Based on Deep Adversarial Learning in the Intelligent Manufacturing System. *IEEE Access* **2021**, *9*, 49557–49575. [CrossRef]
5. Fu, Y.; Hou, Y.; Wang, Z.; Wu, X.; Gao, K.; Wang, L. Distributed Scheduling Problems in Intelligent Manufacturing Systems. *Tsinghua Sci. Technol.* **2021**, *26*, 625–645. [CrossRef]
6. Caiazzo, B.; Di Nardo, M.; Murino, T.; Petrillo, A.; Piccirillo, G.; Santini, S. Towards Zero Defect Manufacturing Paradigm: A Review of the State-of-the-Art Methods and Open Challenges. *Comput. Ind.* **2022**, *134*, 103548. [CrossRef]
7. Espinoza Pérez, A.T.; Rossit, D.A.; Tohmé, F.; Vásquez, Ó.C. Mass Customized/Personalized Manufacturing in Industry 4.0 and Blockchain: Research Challenges, Main Problems, and the Design of an Information Architecture. *Inf. Fusion* **2022**, *79*, 44–57. [CrossRef]
8. Jiang, L.; Shan, J. Genuine Brands or High Quality Counterfeits: An Investigation of Luxury Consumption in China: Luxury Consumption in China. *Can. J. Adm. Sci. Rev. Can. Sci. Adm.* **2018**, *35*, 183–197. [CrossRef]
9. Kiangala, K.S.; Wang, Z. An Effective Communication Prototype for Time-Critical IIoT Manufacturing Factories Using Zero-Loss Redundancy Protocols, Time-Sensitive Networking, and Edge-Computing in an Industry 4.0 Environment. *Processes* **2021**, *9*, 2084. [CrossRef]
10. Leng, J.; Ye, S.; Zhou, M.; Zhao, J.L.; Liu, Q.; Guo, W.; Cao, W.; Fu, L. Blockchain-Secured Smart Manufacturing in Industry 4.0: A Survey. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 237–252. [CrossRef]
11. Zhang, Y.; Xu, X.; Liu, A.; Lu, Q.; Xu, L.; Tao, F. Blockchain-Based Trust Mechanism for IoT-Based Smart Manufacturing System. IEEE Trans. *Comput. Soc. Syst.* **2019**, *6*, 1386–1394. [CrossRef]
12. Zhang, C.; Zhou, G.; Li, H.; Cao, Y. Manufacturing Blockchain of Things for the Configuration of a Data- and Knowledge-Driven Digital Twin Manufacturing Cell. *IEEE Internet Things J.* **2020**, *7*, 11884–11894. [CrossRef]
13. Rana, N.P.; Dwivedi, Y.K.; Hughes, D.L. Analysis of Challenges for Blockchain Adoption within the Indian Public Sector: An Interpretive Structural Modelling Approach. *Inf. Technol. People* **2022**, *35*, 548–576. [CrossRef]
14. Javaid, M.; Haleem, A.; Pratap Singh, R.; Khan, S.; Suman, R. Blockchain Technology Applications for Industry 4.0: A Literature-Based Review. *Blockchain Res. Appl.* **2021**, *2*, 100027. [CrossRef]
15. Hassen, O.A.; Abdulhussein, A.A.; Darwish, S.M.; Othman, Z.A.; Tiun, S.; Lotfy, Y.A. Towards a Secure Signature Scheme Based on Multimodal Biometric Technology: Application for IOT Blockchain Network. *Symmetry* **2020**, *12*, 1699. [CrossRef]
16. Li, K.; Li, H.; Wang, H.; An, H.; Lu, P.; Yi, P.; Zhu, F. PoV: An Efficient Voting-Based Consensus Algorithm for Consortium Blockchains. *Front. Blockchain* **2020**, *3*, 11. [CrossRef]
17. Oyinloye, D.P.; Teh, J.S.; Jamil, N.; Alawida, M. Blockchain Consensus An Overview of Alternative Protocols. *Symmetry* **2021**, *13*, 1363. [CrossRef]
18. Chen, Q.; Xie, Q.; Yuan, Q.; Huang, H.; Li, Y. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [CrossRef]
19. Deng, H.; Cheng, Y.; Feng, Y.; Xiang, J. Industrial Laser Welding Defect Detection and Image Defect Recognition Based on Deep Learning Model Developed. *Symmetry* **2021**, *13*, 1731. [CrossRef]
20. Putra, K.T.; Chen, H.-C.; Prayitno; Ogiela, M.R.; Chou, C.-L.; Weng, C.-E.; Shae, Z.-Y. Federated Compressed Learning Edge Computing Framework with Ensuring Data Privacy for PM2.5 Prediction in Smart City Sensing Applications. *Sensors* **2021**, *21*, 4586. [CrossRef]
21. Chen, Q.; Xie, Q.; Yuan, Q.; Huang, H.; Li, Y. Research on a Real-Time Monitoring Method for the Wear State of a Tool Based on a Convolutional Bidirectional LSTM Model. *Symmetry* **2019**, *11*, 1233. [CrossRef]
22. Latif, S.; Driss, M.; Boulila, W.; Huma, Z.E.; Jamal, S.S.; Idrees, Z.; Ahmad, J. Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions. *Sensors* **2021**, *21*, 7518. [CrossRef]
23. Baboli, A.; Okamoto, J.; Tsuzuki, M.S.G.; Martins, T.C.; Miyagi, P.E.; Junqueira, F. Intelligent Manufacturing System Configuration and Optimization Considering Mobile Robots, Multi-Functional Machines and Human Operators: New Facilities and Challenge for Industrial Engineering. *IFAC-PapersOnLine* **2015**, *48*, 1912–1917. [CrossRef]
24. Ma, J.; Lin, S.-Y.; Chen, X.; Sun, H.-M.; Chen, Y.-C.; Wang, H. A Blockchain-Based Application System for Product Anti-Counterfeiting. *IEEE Access* **2020**, *8*, 77642–77652. [CrossRef]

25. Xiao, L.; Huang, W.; Xie, Y.; Xiao, W.; Li, K.-C. A Blockchain-Based Traceable IP Copyright Protection Algorithm. *IEEE Access* **2020**, *8*, 49532–49542. [CrossRef]
26. Heo, G.; Yang, D.; Doh, I.; Chae, K. Efficient and Secure Blockchain System for Digital Content Trading. *IEEE Access* **2021**, *9*, 77438–77450. [CrossRef]
27. Fernandez-Carames, T.M.; Fraga-Lamas, P. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* **2018**, *6*, 32979–33001. [CrossRef]
28. Iarovyi, S.; Mohammed, W.M.; Lobov, A.; Ferrer, B.R.; Lastra, J.L.M. Cyber–Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems. *Proc. IEEE* **2016**, *104*, 1142–1154. [CrossRef]
29. Qiu, C.; Yu, F.R.; Yao, H.; Jiang, C.; Xu, F.; Zhao, C. Blockchain-Based Software-Defined Industrial Internet of Things: A Dueling Deep Learning Approach. *IEEE Internet Things J.* **2019**, *6*, 4627–4639. [CrossRef]
30. Geissler, S.; Prantl, T.; Lange, S.; Wamser, F.; Hossfeld, T. Discrete-Time Analysis of the Blockchain Distributed Ledger Technology. In Proceedings of the 2019 31st International Teletraffic Congress (ITC 31), Budapest, Hungary, 27–29 August 2019; IEEE: Manhattan, NY, USA, 2019; pp. 130–137.
31. Belotti, M.; Bozic, N.; Pujolle, G.; Secci, S. A Vademecum on Blockchain Technologies: When, Which, and How. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3796–3838. [CrossRef]
32. Nakamoto, S.N. Bitcoin: A Peer-to-Peer Electronic Cash System. 9. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 19 June 2021).
33. Buterin, V. A next generation smart contract & decentralized application platform. *White Pap.* **2014**, 3.
34. Cachin, C. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*; IBM Research: Ruschlikon, Switzerland, 2016; Volume 310, pp. 1–4.
35. Alpos, O.; Cachin, C.; Zanolini, L. How to Trust Strangers: Composition of Byzantine Quorum Systems. In Proceedings of the 2021 40th International Symposium on Reliable Distributed Systems (SRDS), Chicago, IL, USA, 20–23 September 2021; pp. 120–131.
36. Li, Q.; Li, H.; Wen, Z.; Yuan, P. Research on the P2P Sybil Attack and the Detection Mechanism. In Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 24–26 November 2017; IEEE: Manhattan, NY, USA, 2017; pp. 668–671.
37. Kaur, M.; Khan, M.Z.; Gupta, S.; Noorwali, A.; Chakraborty, C.; Pani, S.K. MBCP: Performance Analysis of Large Scale Mainstream Blockchain Consensus Protocols. *IEEE Access* **2021**, *9*, 80931–80944. [CrossRef]
38. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
39. König, L.; Unger, S.; Kieseberg, P.; Tjoa, S.; Blockchains, J.R.C. The Risks of the Blockchain A Review on Current Vulnerabilities and Attacks. *J. Internet Serv. Inf. Secur.* **2020**, *10*, 110–127.
40. Asif, R.; Ghanem, K.; Irvine, J. Proof-of-PUF Enabled Blockchain: Concurrent Data and Device Security for Internet-of-Energy. *Sensors* **2020**, *21*, 28. [CrossRef] [PubMed]
41. Hui, H.; An, X.; Wang, H.; Ju, W.; Yang, H.; Gao, H.; Lin, F. Survey on Blockchain for Internet of Things. *J. Internet Serv. Inf. Secur.* **2019**, *9*, 1–30.
42. Alizadeh, M.; Andersson, K.; Schelen, O. A survey of secure internet of things in relation to blockchain. *J. Internet Serv. Inf. Secur. (JISIS)* **2020**, *10*, 47–75.
43. Content Addressing and CIDs. Available online: https://docs.ipfs.io/concepts/content-addressing/#identifier-formats (accessed on 12 May 2022).
44. Bahga, A.; Madisetti, V. Blockchain Platform for Industrial Internet of Things. *J. Softw. Eng. Appl.* **2016**, *9*, 533–546. [CrossRef]
45. Balakumar, S.; Kavitha, A.R. Quorum-Based Blockchain Network with IPFS to Improve Data Security in IoT Network. *Stud. Inform. Control.* **2021**, *30*, 85–98. [CrossRef]
46. Shih, C.S.; Hsieh, W.Y.; Kao, C.L. Traceability for Vehicular Network Real-Time Messaging Based on Blockchain Technology. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* **2019**, *10*, 1–21. [CrossRef]
47. Chen, C.-L.; Shang, X.; Tsaur, W.-J.; Weng, W.; Deng, Y.-Y.; Wu, C.-M.; Cui, J. An Anti-Counterfeit and Traceable Management System for Brand Clothing with Hyperledger Fabric Framework. *Symmetry* **2021**, *13*, 2048. [CrossRef]
48. Liu, N.; Yu, M.; Zang, W.; Sandhu, R.S. Cost and Effectiveness of TrustZone Defense and Side-Channel Attack on ARM Platform. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* **2020**, *11*, 1–15.
49. Chen, H.-C.; Damarjati, C.; Prasetyo, E.; Arofiati, F.; Sugiyo, D. Blockchain Technology Benefit in Tackling Online Shopping Transaction Revocation Issue. In Proceedings of the 6th International Conference on Frontiers of Educational Technologies (ICFET 2020), Tokyo, Japan, 5–8 June 2020; pp. 191–195. [CrossRef]
50. Chen, H.-C.; Liang, Y.-H.; Hsu, P.-Y. Reconfigurable PM2.5 Sensor Green Deployment Mechanism Based on Blockchain Technology. In Proceedings of the 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications, National Chung Hsing University, Taichung, Taiwan, 18–20 November 2021.
51. Chen, H.-C. A Trust Evaluation Gateway for Distributed Blockchain IoT Network. In *Wireless Internet, WICON 2018, Proceedings of the WiCON 2018—11th EAI International Wireless Internet Conference, Taipei, Taiwan, 15–16 October 2018*; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Chen, J.L., Pang, A.C., Deng, D.J., Lin, C.C., Eds.; Springer: Cham, Switzerland, 2018; Volume 264, pp. 156–162. [CrossRef]

52. Chen, H.-C.; Irawan, B.; Shae, Z.-Y. A Cooperative Evaluation Approach Based on Blockchain Technology for IoT Application. In *Wireless Internet, WICON 2018, Proceedings of the 12th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2018), Kunibiki Messe, Matsue, Japan, 4–6 July 2018*; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Chen, J.L., Pang, A.C., Deng, D.J., Lin, C.C., Eds.; Springer: Cham, Switzerland, 2019; Volume 264, pp. 913–921. [CrossRef]
53. Chen, H.-C.; Kuo, S.-S.; Chen, H.-M. Secure OTT Service Scheme Based on Blockchain Technology. In Proceedings of the 32nd IEEE International Conference on Advanced Information Networking and Applications (IEEE AINA-2018), Pedagogical University of Cracow, Cracow, Poland, 16–18 May 2018; pp. 645–650.
54. RMerkle, R.C. Protocols for Public Key Cryptosystems. In Proceedings of the 1980 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 14–16 April 1980; p. 122. [CrossRef]
55. Juan, B. IPFS-Content Addressed, Versioned, P2P File System. *arXiv* **2014**, arXiv:14073561.
56. Chaganti, R.; Bhushan, B.; Ravi, V. The role of Blockchain in DDoS attacks mitigation: Techniques, open challenges and future directions. *Comput. Secur.* **2022**, *120*. [CrossRef]