Journal of Intelligent Manufacturing (2022) 33:1451–1465
https://doi.org/10.1007/s10845-020-01735-2

A blockchain technology based trust system for cloud manufacturing

Reza Vatankhah Barenji[1]

Received: 22 April 2020 / Accepted: 30 December 2020 / Published online: 22 January 2021

Abstract
Cloud manufacturing (CM) is a new networked manufacturing model that delivers various on-demand manufacturing capabilities to the consumers from the providers. In this model, the provider and consumer never meet each other, thus "trust" is the major enabler for starting a collaboration. In another word, a user must be sure that the requested capability will not be provided with malicious results, and the provider should ensure that the payment will be made on time. In this paper, a novel Blockchain Technology (BCT)-based trust system called "Blocktrust" is proposed to address the trust problem of the CM. First, the CM framework that contains the digital firm, capability pool, and digital certificate issuing units is developed, and then, the private blocktrust peer-to-peer network is proposed and implemented based on Hyperledger fabric. Finlay, the feasibility of the blocktrust is examined under different testing scenarios. The reason for using a private network instead of the public is placing restrictions on who is allowed to participate in the network and also enjoying a network with fast transaction speed. Experiments show that the proposed blocktrust embedded CM is credible and practical.

Keywords Blockchain technology · Cloud manufacturing · Smart manufacturing · Industry 4.0 · Trust system

Introduction

In recent years, the endless competition in the global market drives manufacturing firms towards "digitalization" (Balta et al. 2018). Collaboration is a strategic and practical way to gain a competitive advantage for the digital firms where the core enabler of the collaboration is the "trust" (Helo et al. 2019). Trust needs to be evident in the relationships—how the products should produce, how the connections are vocalized, and how the funds flow. Without it, collaboration falls apart quickly and, sometimes, irreparably. It is a crucial pillar for sustainability, and vital protecting interests (Helo et al. 2019; Li et al. 2020a). Digital firms require computerized and internet-based tools for launching effective communication, and interoperation to perform a reliable capability sharing (Li et al. 2020a) Indeed, much of the newly published papers on this context is concerned with Cloud Manufacturing (CM) as a terminology, inspired by cloud computing and adapted to manufacturing enterprises (Li et al. 2020a; Yuan et al. 2019; Lu et al. 2019).

CM is a service-oriented cyber-physical system for inter-organizational collaboration which integrates various recognized engineering paradigms (i.e. IoT, digital twins, capability) with the power of cloud computing (Yuan et al. 2019). It is a centralized system for sharing different kinds of capabilities, where the system holds a pool containing soft capabilities (e.g. such as optimization, consultancy services, and cyber-based facility control), and hard capabilities (e.g. products and services) (Lu et al. 2019). It works under "Pay as You Go" terminology and every enrolled firm can share or get few capabilities from the pool (Yuan et al. 2019). Recently, the investigation has been made by some scholars on CM (D'Aniello et al. 2020; Talhi et al. 2019; Simeone et al. 2020). However, most of these researches focus mainly on the architecture of the system as well as the application of distributed features of CM for capability allocation (D'Aniello et al. 2020).

The common ways for creating trust among the partners is not applicable in CM model since; the involved firms are micro, small or medium size and are not well-known for their products or services; the CM platform does not stand behind the interactions; the capability selections accomplish without any meeting; and the parties know little or nothing about each other. To establish trust in CM, a trust system is needed to provide reliable trust-related information and must satisfy three

✉ Reza Vatankhah Barenji
Reza.vatankhah@hacettepe.edu.tr

[1] Department of Industrial Engineering, Hacettepe University, Cankaya, Ankara, Turkey

challenges: (1) provide information that allows service requesters (or service providers) to distinguish between trustworthy and non-trustworthy service providers (or service requesters), (2) encourage service providers as well as service requesters to be trustworthy, and (3) discourage participation for those who aren't (as a requester, provider or both). The trust system in CM should be beneficial to requesters and providers since both sides are responsible to provide certified responses. As an example, a provider might respond to be on time on providing products while the requester is responsible to get paid on a date. Both parties should believe that the provided data by the trust system is effective, accurate, and not manipulated. What matters is not only how the system works, but also the technology for keeping this information is vital for the proper working of the system.

Although significant advancements have been achieved in using real-time data for performance improvement in CM (Talhi et al. (2019); Simeone et al. 2020), there are unsolved issues for how to apply the real-time data-driven decision to trust problems in CM due to an increase in process complexity and unpredictable exceptions. This issue is summarized as follows. How to design a new and effective CM framework to offer real-time trust score for an IoT-enabled flexible job shop as a provider or a consumer of a service. Blockchain Technology (BCT) is an emerging decentralized and transactional data sharing technology across a network of untrusted participants (Nakamoto 2019). It is developed to reach agreement among collaborating parties without a central authority assistant (Nakamoto 2019). BCT peer to peer networks may provide a fully trusted system by running autonomous smart contracts. An effective trust system in CM affects many aspects of the system operations including reductions on search to partners, more effective search for a capability, improvements in cloud computing utilization, and faster and easier disaster recovery. With all these benefits, CM needs a universal, feasible, and empirical BCT-based trust system.

To address the above-mentioned challenge, a new trust system is presented in this study to provide trust to the collaborators by extending the BCT to the CM field. Under this system, capabilities can be embedded with trust scores. Then the nodes of the network can interact with each other in the selection, during, and after collaboration stages. The proposed peer to peer trust system is called "*Blocktrust*", working in such a way that no central authority contributed to the trust, but it is maintained.

State of the art

Cloud manufacturing

The increasing competition in the market (especially in countries offering low wages) motivate manufacturing enterprises to investigate and adopt new models of manufacturing and business and shift their long term supply chains to possibly short term broader manufacturing networks (Ghasempouri and Ladani 2019). The trend is towards industries in which the capabilities are accessible and employable from anywhere in the world at any time (Barenji 2013). CM has been introduced in the past few years (Li et al. 2020a; Yuan et al. 2019; Lu et al. 2019). It has been designed to accumulate and integrates huge, valuable capabilities (Barenji et al. 2015) from various disciplines and different manufacturing firms for consumers. In this model, capabilities are largely autonomous, geographically distributed, and heterogeneous in terms of their: operating environment, culture, social capital, and goals (D'Aniello et al. 2020). It is a model that allows an omnipresent, pervasive, useful, and on-demand network connection of the manufacturing enterprises to a common basin of configurable manufacturing capabilities (Lu et al. 2019). In the model many firms wish to effectively and efficiently use their sources, share some capabilities with others and some consumers use these capabilities with no investment in facilities and technology. A schematic configuration for CM is shown in Fig. 1.

In the CM model, ICT infrastructure plays a major enabler role by information exchange/sharing, safe communication, coordination, and collaboration services. It plays the hero of an "operating system" or executor, hiding the details of the involved companies (Wang et al. 2020). It manages distributed services belonging to varied owners and works under Service Oriented Architecture (SOA) using three main entities as, "Service provider", "Service Manager", and "Service Requester" (Yi 2020; Zhang et al. 2019). It causes a smoother information flow, and increases the flexibility of the business processes, and helps different partners to collaborate. "Service Manager" has a Capability Pool (CP) that holds virtualized capabilities from different stakeholders independent of their Enterprise Resource Planning (ERP) and Manufacturing Execution System (MES) type and brand (Viriyasitavat et al. 2018). CP is an environment host virtualized capabilities provided by different stakeholders and offer these capabilities to the other firms (i.e. customers). It interconnects requesters (e.g. individuals, companies,) and suppliers (e.g. manufacturing enterprises, supplies).

Blockchain technology

BCT is a distributed open ledger that provides a way for information to be recorded and shared by a community (Nakamoto 2019; Valdeolmillos et al. 2019). Each of the participated members maintains his or her copy of the information and all members must validate any updates collectively. The information might be a transaction, contracts, assets, identities, or practically anything else that can be described in digital form. BCT entries are

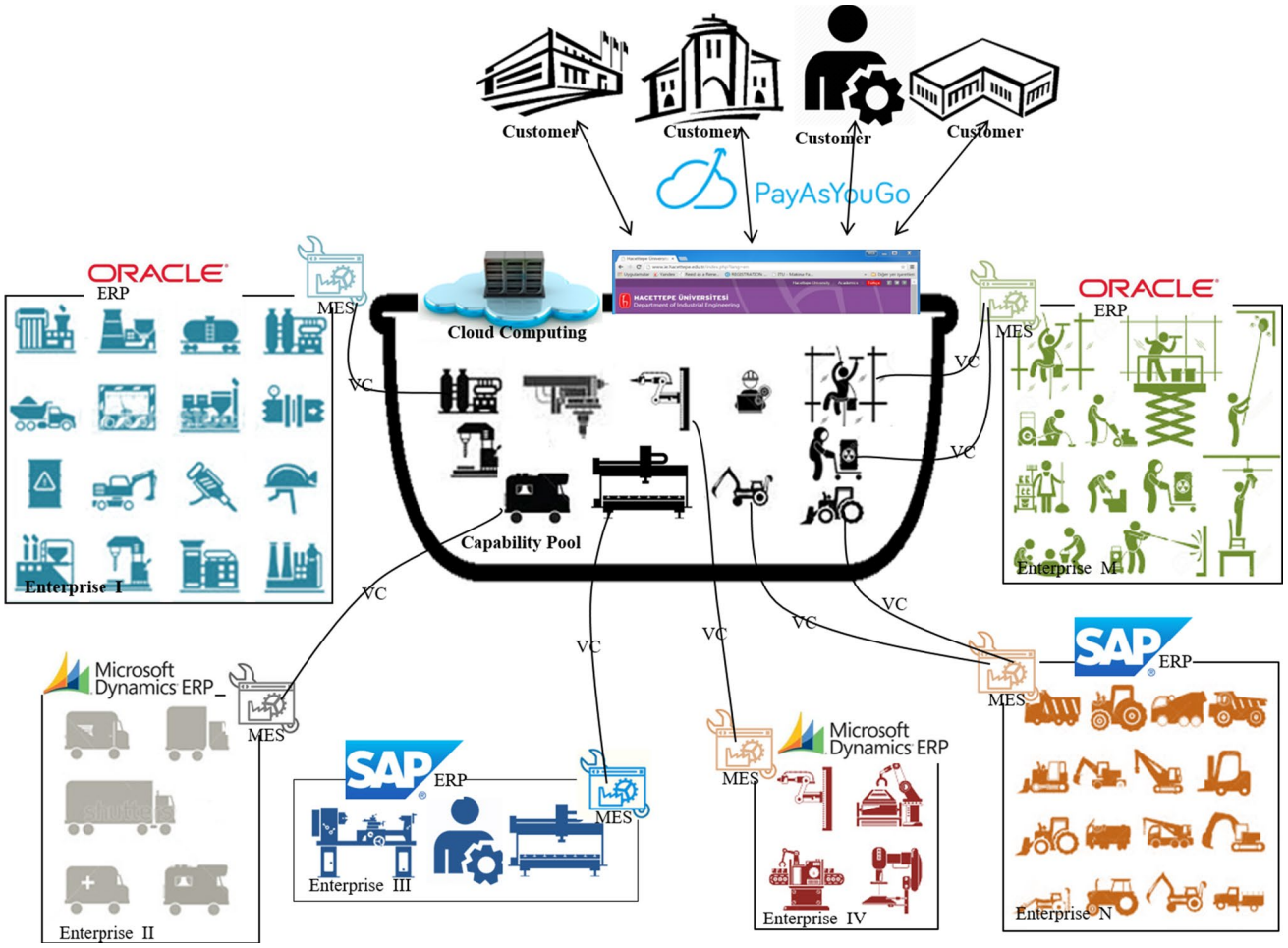

Fig. 1 Schematic configuration for Cloud manufacturing

permanent, transparent, and searchable, which makes it possible for community members to view transaction histories in their entirety. Each group of transactions is a new "block" added to the end of a "chain." A protocol manages how new edits or entries are initiated, validated, recorded, and distributed. Cryptographic validation is used to link the records chained in the blocks.

A chain contains many blocks, and the first block of the chain is called a genesis block. Each block references and identifies the previous block to form an unbroken chain. Each block in a typical ledger holds a block number (i.e. index) for showing the place of the block in the chain (e.g. Genesis block has index 0), the hash of the previous block for validating the previous block, Timestamp for demonstrating when the block is added to the chain, the stored data in the block, a random unique number to prevent a reply attack on the chain (i.e. Nonce;), and hash that is generated for this block. Modifying/ changing the stored data of a block will change its hash. This will invalidate all the subsequent blocks in the chain (since their hashes are based on the hash of the previous block).

To create a super useful form of digital record and sharing, three principal technologies are combined in blockhead including; a) private and public-key cryptography, b) a distributed network with a shared ledger, and c) an incentive to service the network's transactions, record-keeping and security (Khaqqi et al. 2018). Two peers wish to transact over the internet, each of them holds a private key and a public key. The main purpose of the first technology in BCT is to create a secure digital identity reference. Identity is based on possession of a combination of private and public cryptographic keys which is named as a digital certificate. Strong control of ownership is not enough to secure digital relationships; it just solves the authentication problem of the network. It must be combined with a means of approving transactions and permissions (authorization). For BCT, this begins with a distributed network. A distributed network is a large network with validators, the validators use mathematical verification

to reach a consensus that witnessed the same thing at the same time. When cryptographic keys are combined with this network, a super useful form of digital interaction emerges. A distributed open ledger is a database that is shared and synchronized across the peers of the network, spread across multiple sites, institutions, or geographies. In a distributed open ledger, the entire ledger is replicated by each peer, and as such, the chain is highly tampered proof. A public ledger is open to anyone who wants to participate in using open-source software. All participants can add to the chain (create new blocks), but cannot modify existing blocks. Enterprises can set up private blockchains to protect the privacy and security of their data. Participation in a private blockchain requires an invitation, which itself is also validated by the network starter or a set of rules that can put into place. Such a network is known as a permissioned network and puts a restriction on who is allowed to join. Private blockchains can also restrict participant activity such that certain transactions can only be carried out by certain participants and not others, even though they're on the network. This creates an added layer of privacy. An example of a private blockchain is Linux Foundation's Hyperledger Fabric, designed to cater to enterprise requirements. BCT applications for manufacturing are still in their infancy and few applications use BCT in this context (Li et al. 2018; Liu et al. 2020; Barenji et al. 2019).

Trust in collaboration

Confidence in the third party in collaboration increases efficiency that is a phenomenon in economics called external effects (Li et al. 2019). Lack of confidence is a huge hamper on developing collaboration and it affects its performance. Scoring the trust rate for companies in a standard way might bring confidence and better operational and business performance on inter-organizational collaboration (Zhu et al. 2020a). Therefore, the semantic delamination of "trust" under collaboration is necessary (Yang et al. 2019). It is expected that in a community all the firms be embedded with trust in a commonly recognized norm and all they relied on trust commitments systematically and all act honestly when face with various possibilities. It imposes an obligation to keep the promise thus convincing the others that will not act against their interests, and accepting without doubts and suspicions even without detailed information about the other party's actions. Trust is an organizational asset, which is the result of an assessment of the organization's competency and credibility to relying on the other party, and it is the result of historical ties in which the organization acquired from past collaborations, which is commonly called reputation. It is a key party for the Collaborative Network Organizations (CNOs) on creating interaction and relationships in platforms like virtual organization and cloud manufacturing (Zhu et al. 2020b).

Trust in inter-organizational relations traditionally might create in one/some of the following ways (reader would add more): (A) It might create among the local organizations by scrutinizing each other before, during, and after the interaction. (B) Global organizations by frequently interactions and relationship might create trust; for example, an organization in A location (e.g. AA) do a partnership with another in B (e.g. BB), AA and BB frequently interact with each other and learn to trust. (C) Trust created between two global organizations using local peers. For example; AA and BB trust each other, further organizations in A and B locations (e.g. XA and XB) learn about the AA and BB by their interactions (D) Trust might be borrowed from other organizations. For example, organizations are likely to be a joint venture of another well-known organization (E) trust might create over many years by establishing brand names.

The basic assertion of this paper is that trust can emerge due to the feedback value that can be distributed, stored, and assign to every provider and requester in the network using -BCT. An important change to meet the standard of Industry4.0 is how to digitalize services under printed by technological innovations. In this paper, we tackle the trustiness in CM by exploring the possibility of using BCT. CM provides many services with similar functionalities but differs in Quality of Service (QoS) values (e.g. availability, response time) and preferences (e.g. price, trust score). The service providers and consumers are looking for an effective methodology to select the partner (s) in this platform. Several techniques are available to select preferable and best-fit services in different contexts (Li et al. 2020b; Laili et al. 2020). However, these techniques discovered some unsolved issues for the CM platform:

- It is not a good proposition for a fully computerized system like CM to assume that reputation are inherently trusted.
- The use of the central authority of the CM brings some challenges including but not limited to scalability, maintenance overhead, managing the denial of service, and fraud.
- Managing attribute-based digital certificates in CM dynamic platform for geographically distributed service over the internet is empirically impractical since the trust values are dynamic and need real-time information.

To overcome the above complications, this paper introduces a trust-based system for CM buttress by BCT.

Proposed Cloud manufacturing framework embedded with blockchain technology

The trust system developed in this paper is embedded by BCT called "Blocktrust". Figure 2 demonstrates the CM framework embedded with blocktrust. The framework has three main units: Digital Firm Unit (DFU), Digital Certificate Issue Unit (DCIU), and Capability Pool Unit (CaPU) and works upon Service Oriented Architecture (SOA). CaPU is a service manager, DFU is service requester (consumer) and/or a service provider, and DCIU is a service provider. A new user (i.e. firm) should first enter and register on the system. At this stage, all the general information, technical capabilities, and the past interactions of the firm should provide to the CaPU. This unit requests a Digital Certificate (DC) to the firm from the DCIU. Any firm that holds a DC is a DFU that can requests/provide service(s) from/to the CaPU. In the framework, every unit has its operating system (i.e. ERP, CPOS, DCIOS) these operating systems manage the internal interaction of the unit. In a DFU, the internal interaction of the firms manage by Enterprise Resource Planning (ERP) and the unit interacts with others using Digital Firm Application (DFA). To real-time collecting data from the shop floor, the ERP system is in connection with Manufacturing Execution Systems (MESs). MES acquire the data of facilities and preserve this data on the MES data server. Using DFA, the units based on a request can share internal information with other units in a B2B manner.

Similarly, the operating system of the DCIU (i.e. Digital Certificate Issue Operating System (DCIOS) is responsible for managing internal and external interactions of the unit. This system is in connection with three main sub-systems including; registration, trust as a provider, trust as a consumer. The operating system of a CaPU is named as capability pool operating system. This system is in connection with three main sub-systems including login/logout, planning/scheduling, and pricing/ alliance/ payment. This unit is the core unit in the network and is responsible to collect the exceed capabilities from the DFAs and offer these capabilities to the requester DFAs. The technical, general, and awareness information of exceeding capabilities is provided to the CPOS. According to the requested capability features, CPOS provides the matched capabilities to the customer. In parallel pricing/alliance/payment sub-system supports contract related operation in the system.

All of the units hold a peer of "Blocktrust" network. Blocktrust contains many peers on a distributed peer to peer network. All the trust information of the units (i.e. ledger) is recorded in this network. CaPU or any other peer may hold a mining peer in case it is responsible for the consensus (i.e.

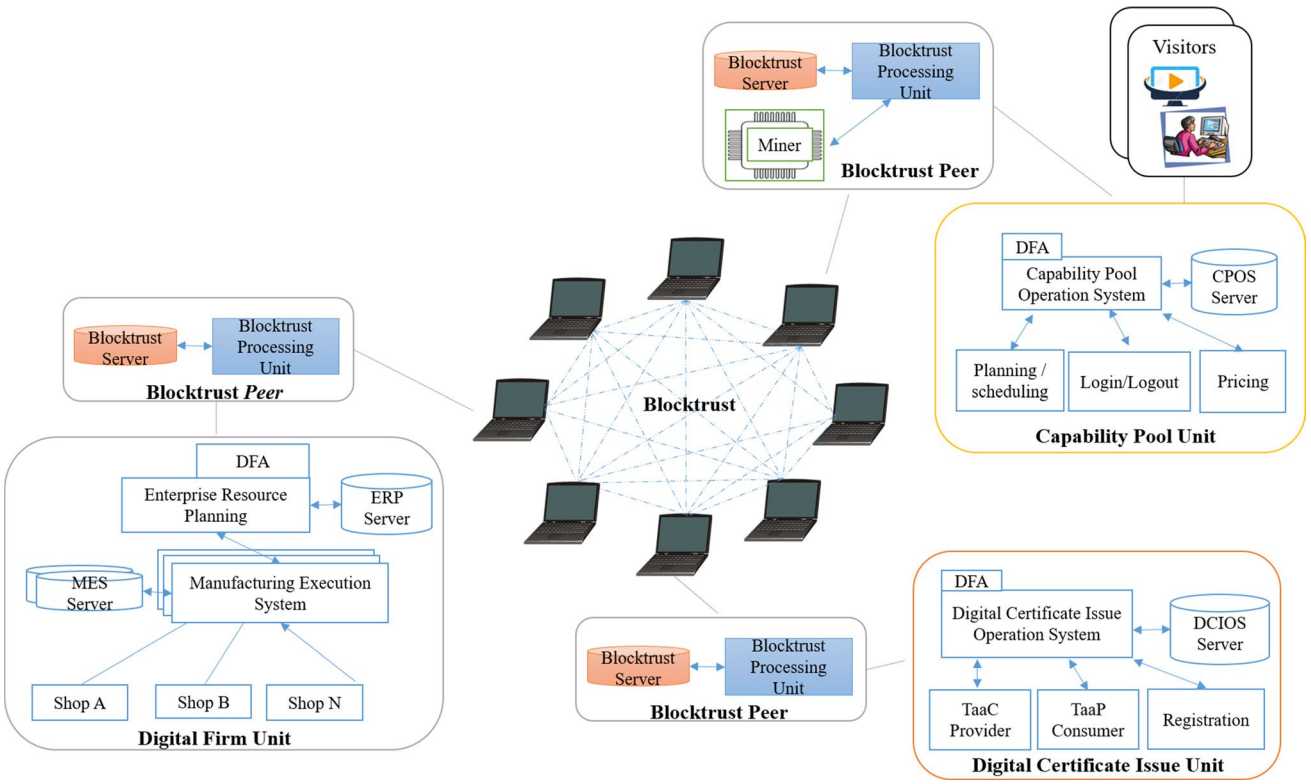

Fig. 2 CM framework embedded with blocktrust

ordering) process. Blocktrust provides an open, unrestricted trust record and rating of the firms. In this trust system, a firm utilizes and provides the foundation of the network at the same time. A peer of blocktrust, records all the trust scores of the network, which is directly available to other peers without the need for any central coordination. All the peers in the network are equal and they take an identical role except the miner peer within the CaPU. The new trust scores are updated and stored in a peer, once the peer is connected to the network. This means that the trust scores on the blocktrust cannot be lost or destroyed because to do so would mean having to destroy every single peer on the ecosystem.

Blocktrust network

The internal components of the blocktrust network are illustrated in Fig. 3. For simplicity, only three peers belong to DFU1, DFU2, and DFU3 are denoted as Peer1, Peer2, and Peer3 respectively. The internal actions of the system are accomplished by the CaPU and this unit is the miner (i.e.orderer) peer in the network. In CM, every DFU can have a consumer, provider, or both consumer and provider roles, so DFA might act as a provider, or consumer or both (e.g. DFA1 is a provider, DFA2 is a consumer and provider, and DFA3 is a consumer). Every peer holds one or some smart contract(s) and the ledger(s). For instance, Peer1 has a Trust as a Consumer (TaaC) smart contract for accessing the TaaC ledger, while Peer3 holds two smart contracts to access the Trust as a Provider (TaaP) and TaaC ledgers. The channels (e.g. Channel1, channel2) in the system are used to connect DFA to the peers and then to the ledgers.

Figure 4 represents the query and updating processes of a ledger, as shown in the figure, a DFA should send a transaction proposal to an appropriate channel. The transaction proposal flows to a peer through the channel and peers invoke a suitable smart contract to access the information available on the ledger and makes a query response to the proposal. This transaction can be added to the ledger if the DFA requests a mining node to mine the transaction. In blocktrust network, mining is the process of reaching an agreement on the next set of transactions (i.e. trust scores) which should be added as a block to the ledger. The consensus algorithm of the blocktrust is deterministic and inspired by the Hyperledge fabric framework (Androulaki et al. 2018). The consensus involves three distinct steps: endorsing the transaction, ordering transaction to a block, and commitment of the block to the ledger. The transaction is created on a DFA, where this software kit prepares and sends a transaction proposal to endorsers and then endorsers simulate the proposed transaction and generate the sets of the Reading set. The Read-set captures what was read from the current ledger. The Read-set is then signed by the authorization peer and returned to the DFA to be used in future steps of the transaction consensus flow. Next, the DFA submits the authorized transaction and the Read-set to the ordering service. The miner takes the authorized trust score transaction and Read-set data, orders this information into a block, and distributes the block to all committing peers. The ordering service does not process transactions, smart contracts, or maintain the shared ledger. It only accepts the endorsed transactions and specifies the order in which those transactions will be written to the ledger. The commitment peer validates the block by proving to make sure that the Read-set still matches the current trust score available in the ledger. When the commitment peer validates the transaction, the block is written to the ledger, and the trust transaction of the firm is updated. If the commitment peer finds that the Read-set does not match the current trust score, the transaction ordered into a block will not be updated in the ledger. Commitment peers are responsible for adding blocks of

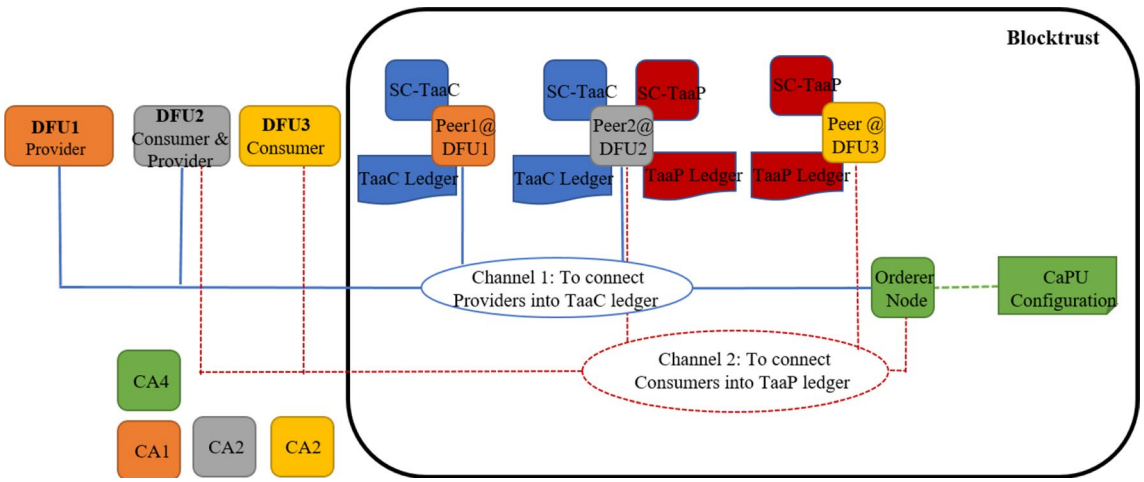

Fig. 3 The internal components of a blocktrust network

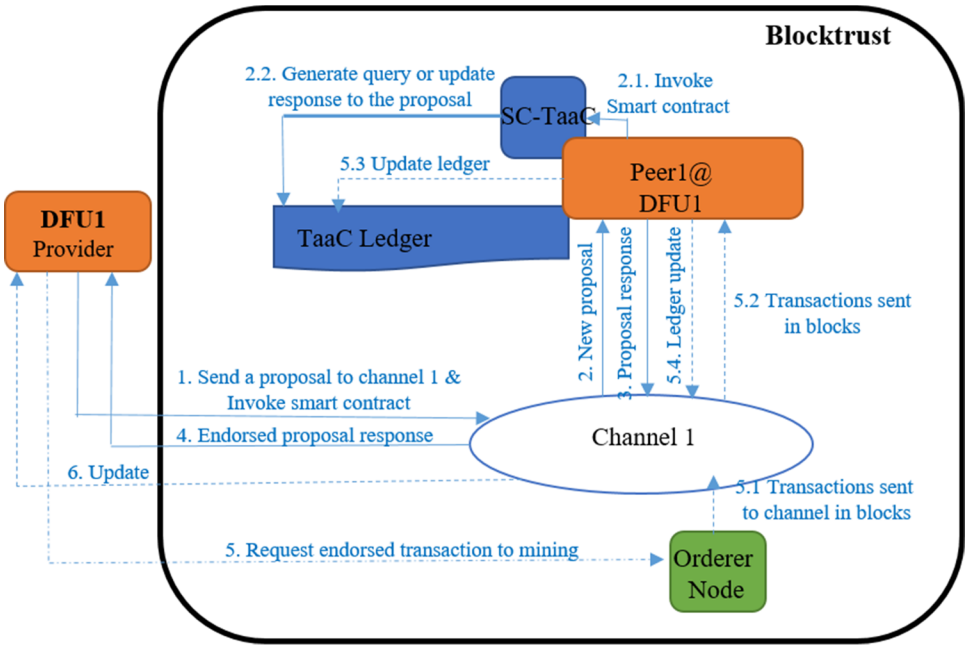

Fig. 4 The query and updating process to a ledger

transactions to the ledger on blocktrust network and updating the trust transactions using smart contracts. Finally, the commitment peer asynchronously notifies the DFA of the success or failure of the transaction. In blocktrust, a smart contract is a chaincode that runs on the peers and enables the peers to do the actions as endorse peer, orderer peer, or commitment peer. In endorse peer, smart contracts are used to generate the Read-set of the proposed transaction. Order peers use smart contract to create the block, and commitment peer use smart contract to validate the provided block and add the new block to the ledger of the blocktrust.

Ledger structure and trust scoring method

Blocktrust is a peer-to-peer network of consumers and providers. Every peer has a copy of a ledger to hold feedback values of the peers' interactions. Unlike the common file sharing P2P networks in blocktrust the quality of variety of services should be traced. However, the protocol for all is based on query-response architecture of the first generation P2P networks. This work asserted that, the trust/distrust for a firm can be based on feedback values of the past interactions. In the blocktrust ledger we should store the feedback value of the past interactions and these values should be used to extract the trust/distrust scores for the firm. The feedback values also should be considered for range of QoSs. Depends on the role of the peer, the specification of the QoSs would be changed. Trust matrix is designed in way that the columns represents the feedback values and rows are for every QoS. To extract the trust and distrust scores the approach proposed by Selçuk et al. 2004 is used since it is a well-known trust evaluation protocol for query-response P2P networks.

The ledger in the proposed system is a matrix called the "trust matrix". As shown in Fig. 5, the trust matrix is a matrix with M columns and N rows. Columns are binary vectors of M bits (e.g. 8) rows represents a particular type of quality of service (e.g. price, quality). A 1 bit represents satisfaction, a 0 represents dissatisfaction. An integer beside each row (K), specifies the number of significant bits in the particular row. After an interaction, the satisfaction/dissatisfaction results are written at the most significant bit and shifts the present bits to the right $\left(\text{e.g.}\frac{\text{Trust Vector}:11010000}{\#\text{ of significant bits}:4}\overset{\text{After collaboration}}{\longrightarrow}\frac{\text{Trust Vector}:11101000}{\#\text{ of significant bits}:5}\right)$. To convert a trust matrix to a "trust rating matrix", a row on the trust matrix with K significant bits is divided by 2 K for conversion into a row in the trust rating matrix in the (0,1) interval A distrust rating matrix is also computed from the complement of the trust matrix $\left(e.g.\frac{\text{Trust Vector}:11101000}{\#\text{ of significant bits}:5}\overset{\text{To}}{\Rightarrow}\frac{\text{Trust rating}=\frac{(11101000)2}{2^5}=0.90625}{\text{Distrust rating}=\frac{(00010000)2}{2^5}=0.0625}\right)$.

When trust and distrust rating matrixes regarding a firm is calculated, a trust and distrust score is calculated for the firm according to their trust and distrust rating matrixes. The threshold Z specifies the number of the rows (i.e. quality of services) to be considered for a trust and distrust rating matrix. First, trust/distrust ratings are sorted by their trust rating according to the min-distrust max-trust criterion. A ranking Z is selected and the trust and distrust score of the firm is determined as the average of the trust and distrust ratings.

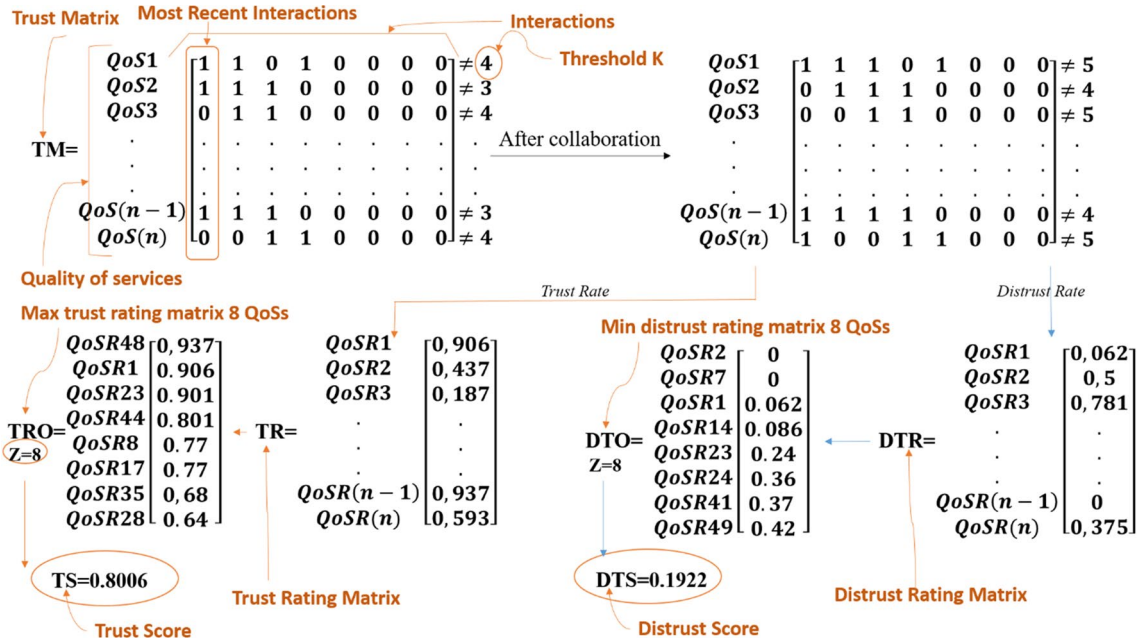

Fig. 5 Trust matrix, trust rating, and trust score

The internal interaction among the units of the framework can be separated into two main groups including the interactions before collaborations and, the interaction after collaboration. The first group of interactions includes interactions to add a new member to the network, and the interactions' need to do to find a suitable capability for a firm. The second group is interactions to provide trust transactions to the nodes of the collaboration.

Interaction before collaboration

Issuing a digital certificate for a new user

A new firm should have DC to become a DFU in the ecosystem. DC provides two sets of information to the firm including; firm identification, and initial trust scores of the firm. Figure 6 illustrates the sequence diagram of the phases to issue a digital certificate for an applicant firm. As shown in the figure, an applicant firm (DFA@1) must provide its general information, technical capabilities, and also a history of the past received/provided services to CPOS in CaPU. CPOS collects this information and request a DC for the firm from a DCIU by sending all the information of the applicant to the DCIOS. As soon as the information is received, the registration unit in DCIU uses this information and prepares a DC request proposal for the applicant. This proposal is directed to the DCIOS from the registration unit. The past interaction of the applicant firm after classification as provider and consumer in DCIOS is sent to the TaaC and TaaP units for the scoring process. TaaC and TaaP units calculate the trust scores for the firm according to the information provided by the applicant. If the provided information is not appropriate and or insufficient for scoring, TaaC and TaaP units send a query to the past interacted firms and based on their responses the trust scores can be calculated. The obtained scores return to the DCIOS and this unit issue a DC for the applicant. Finally, DC is referred to as the applicant and blocktrust through DFA of the DCIU and CPOS.

Request, receive a capability from the pool

A firm with a DC can request and/or provide capability from/to the pool. As soon as the interaction among the firms is finished, the interacted firms should provide trust scores to each other and this information should be recorded on blocktrust network. Figure 7 demonstrates the sequence diagram to obtain a capability from the pool. A digital firm looking for a capability (DFA@1) should enter the system through CPOS and login/logout units by providing the entrance information available on DC. This DFU can request a service from the pool by providing the specification of the capability, acceptable price, and suitable TaaP scores. The received request is analyzed by CPOS and a list of the existing matched capabilities that exist in CPOS server is prepared and returned to the CPOS. This unit then asks the Planning/scheduling, Pricing, and Blocktrust units to check and provide information related to accessibility, price, and TaaP scores. As soon as the information is received to CPOS, this unit provides a list of the complemented capabilities including its availability, price, and TaaP scores to the requester. The consumer

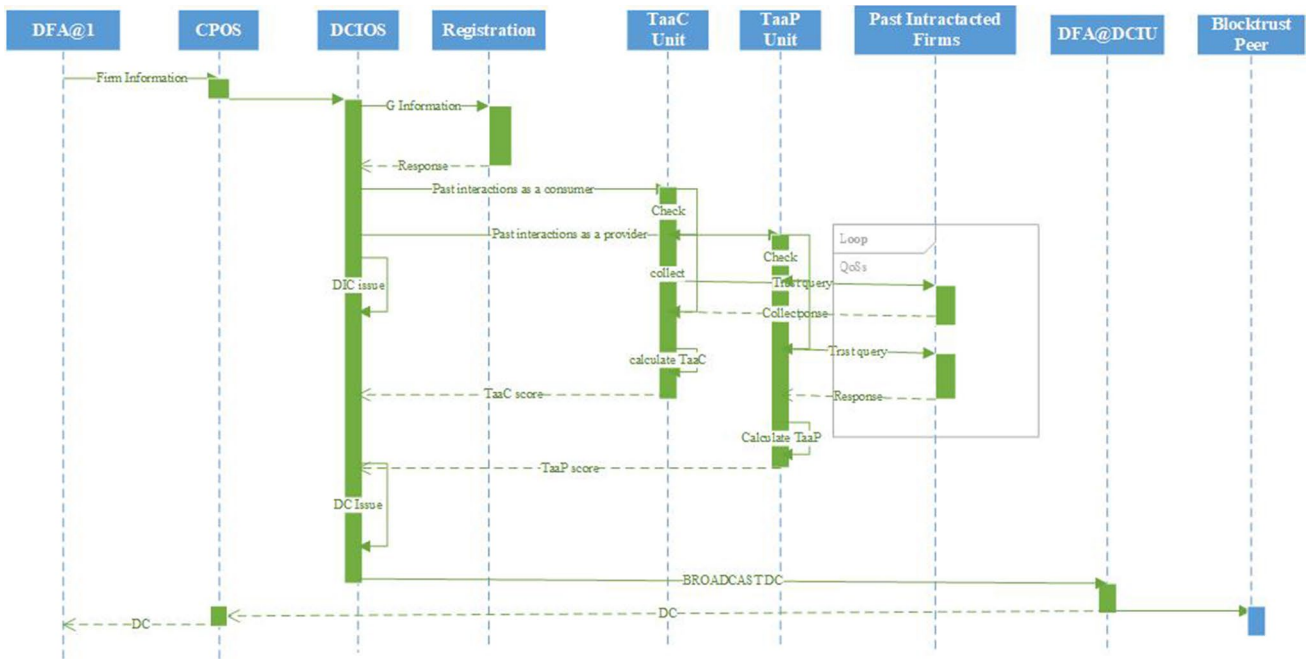

Fig. 6 Sequence diagram for issuing a digital certificate to an applicant firm

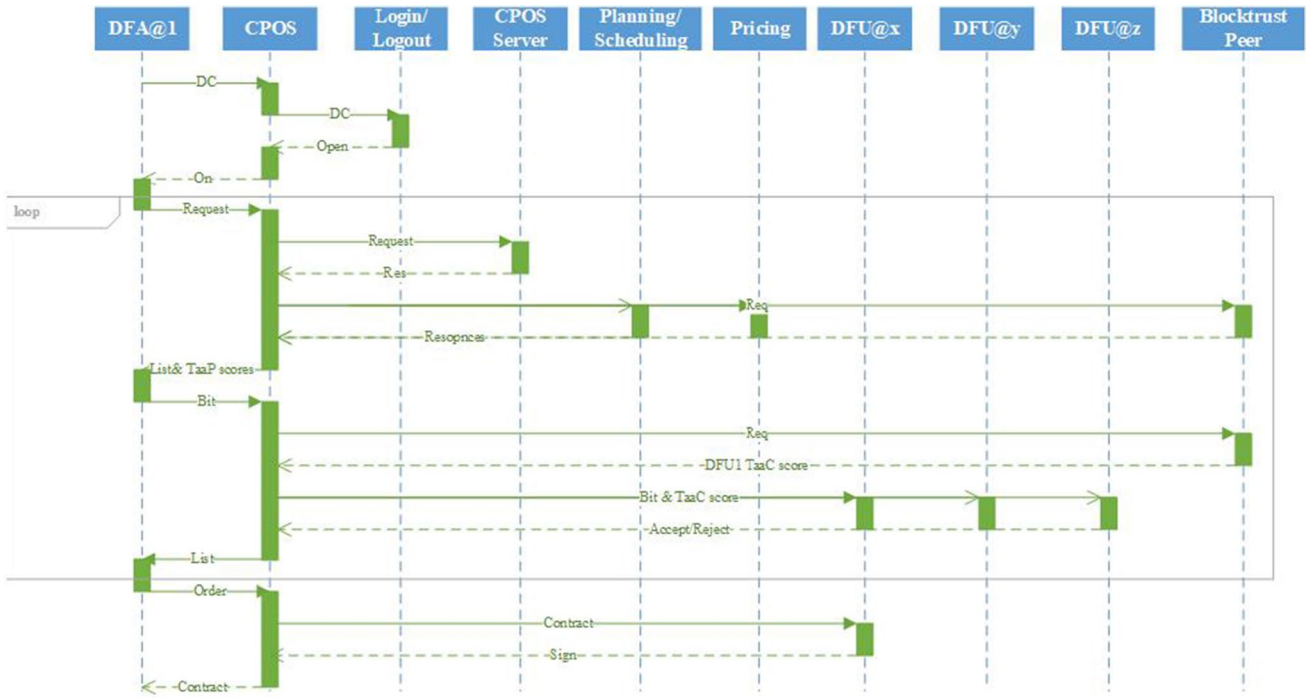

Fig. 7 Sequence diagram for request a service from the system

may select some capabilities in the list and place an initial offer to the CPOS. The CPOS connects to the blocktrust to find the latest TaaC score of the requester and then provide the TaaC score of the requester to the providers and asks the selected provider to decide to accept or deny the offer. According to the response provided by the provider DFAs, CPOS prepare a list of the response to the consumer DFA. The consumer DFA may make an order by accepting the list.

A contract is sent to the provider and consumer DFAs when the list is approved.

Figure 8 demonstrates the sequence diagram of the interaction among the units to add a new capability to the pool. As shown in the figure, a DFA should use DC to enter the system through the CPOS and login/logout units. This DFA must also provide full technical information on the provided capability and the availability, and price information to the CPOS. This unit shares technical information on the capability with CPOS server, availability, and price information related to the capability to be stored on Planning/scheduling and pricing/payment units respectively. As soon as a response configuration is returned to the CPOS, this unit will return a confirmation message to the DFA and the capability will be available to the consumers.

Interaction after collaboration

Trust scores for a completed interaction

As shown in Fig. 9, once the interaction between two firms is completed, the CPOS asks the provider and consumer DFAs to prepare TaaC and TaaP transaction proposals. As a fact provider will arrange a TaaC and the consumer makes a TaaP

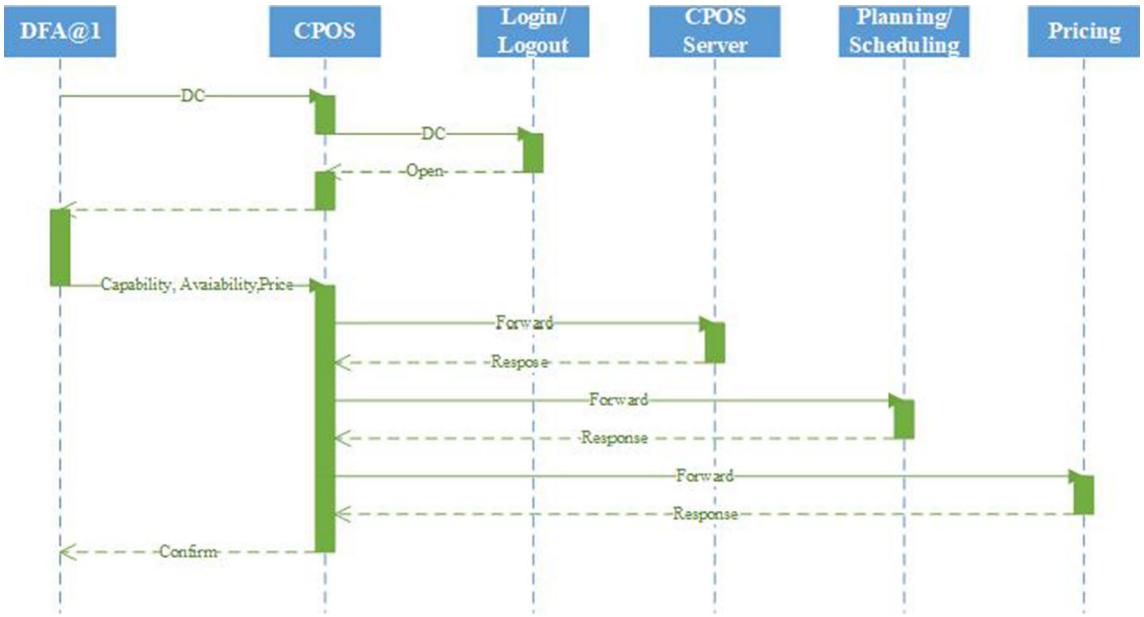

Fig. 8 Sequence diagram to provide a capability to the system

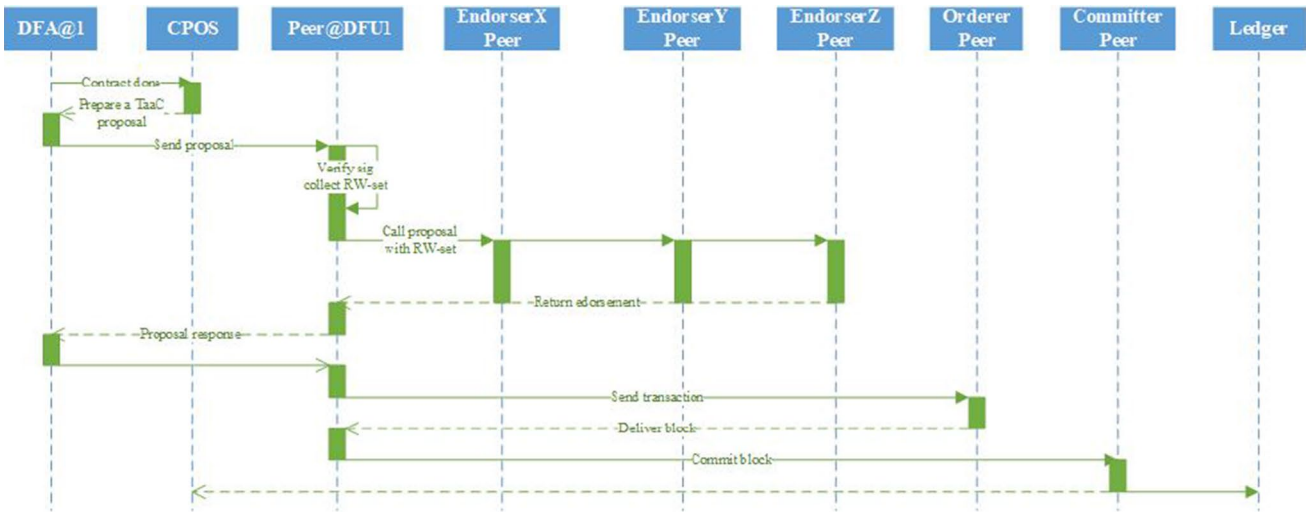

Fig. 9 Sequence diagram of the system units after collaboration

proposal. For the sake of simplicity, only TaaC is explained and represented in Fig. 9 and the approach is similar to the TaaP. The provider DFA prepares a TaaC transaction proposal and sends this proposal to its local blocktrust peer. Local peer verified the DFA's signature and then invoke the chaincode and create an RW-set. The local peer submits the proposal to the endorser peers. The smart contract on the endorsers peers executed and the endorser node simulates the transaction, signs or rejects it, and returns the response to the local peer. The local peer forwards the endorsed transaction to the DFA. The DFA collects the endorsements and sends the endorsed transaction to the orderer through the local peer. The orderer peer runs the consensus algorithm on the endorsed transaction and creates a block of transaction. Then, orderer sends the block of the transaction to the local peer node. The local peer requests a committer peer to write the block of the transaction to the ledger. The committer will inform the CPOS as soon as the scoring is finished.

Case study and implementation

The simulation test platform reported in (Barenji et al. (2016)) is employed to demonstrate the effectiveness of the proposed CM framework embedded with blocktrust. We used Petri-nets for demonstrating the interactions among the units and data integrity in the framework. For examining the performance of the blocktrust in the framework, a prototype peer-to-peer network is developed and tested under different working scenarios. In modeling, by Petri-nets we used a process-oriented approach (Barenji et al. 2017) where the actions of the units are represented by a place and the start and end of the actions are shown by transactions. We developed a Petri net model for each of the units according to their roles in the framework these models perform the essential actions of the units as soon as their actions are completed the Petri net model of the unit generates a token in the final transaction and this token acts as an input token for the next unit Petri net model. Using this approach, the internal actions of the units are examined, and also the accomplishment of the entire system is tested.

As shown in Fig. 10, five types of models including visitor, CaPU, consumer firm, blocktrust, and DCIU is developed and connected. There exist fifteen providers connected to the CaPU. For each of these firms, a trust matrix with sixteen columns (to represent historical interactions) and one row (to represent the quality of the service) is defined and stored in blocktrust. For the sake of simplicity, only one kind of capability is available on the system. The DCIU creates the trust matrix for the visitor by randomly assigning numbers. The interactions within and among the units are examined by controlling by tokens flows on the model and among the models. The interactions among the models are realized by firing the appropriate transaction(s). When a transaction is fired in the model depend on the number of the output arcs and weight of the arcs the tokens is flowing in the system and the flow of the tokens simulates the interactions (Fig. 10).

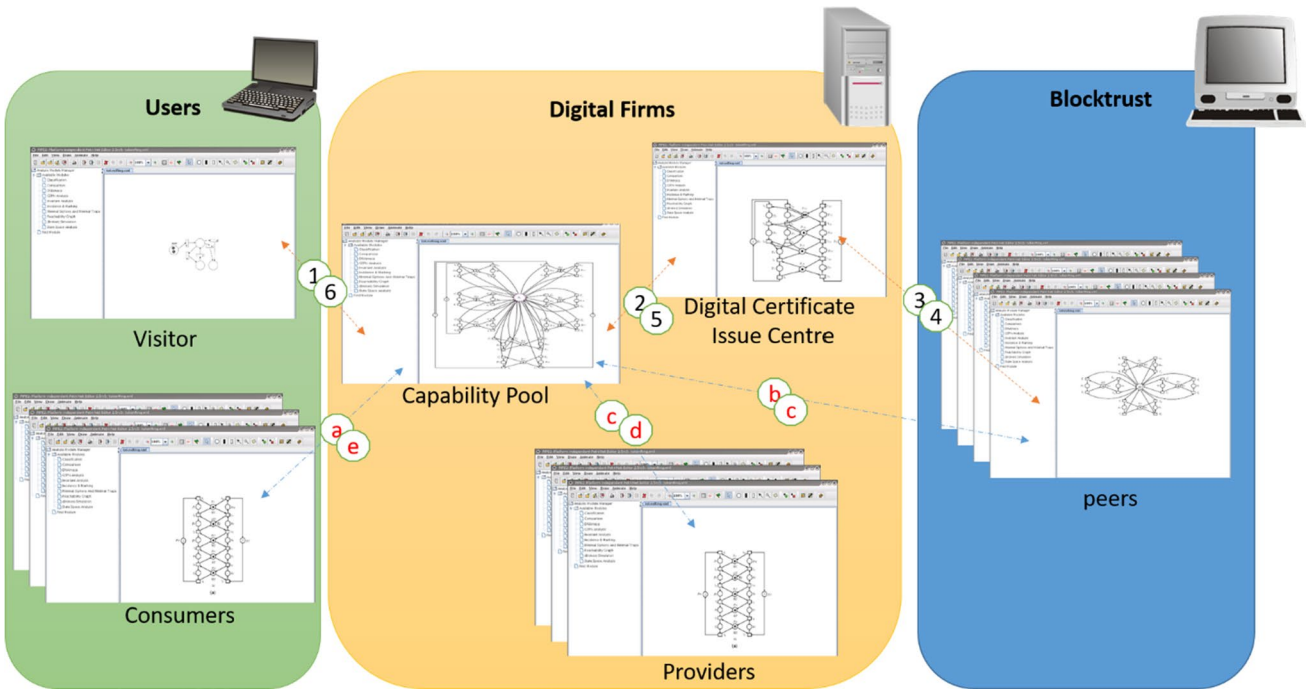

Fig. 10 Interaction among the units and blocktrust

To test the interactions among the models two testing scenarios are employed: a. interaction between the consumer and CaPU for reading a trust score, and b. interactions between a visitor firm, CaPU, and DCIU for writing a trust score.

In the first test scenario, the consumer demands a capability from the CaPU (i.e. fire a transaction hold an output arc with one weight), the CaPU provides a list of the available capabilities including the trust scores, price, and available capacity of the providers. At this stage, the CaPU asks the consumer to provide the domain for an acceptable price, quantity, and trust score. Based on the provided domain the CaPU delivers a list of providers and consumers' fees to select the provided capabilities and start the collaboration (select a capability by entering a token on availability resource).

In the second test scenario, a visitor firm asks CaPU to join the system by providing the general, technical information of the firm and records of interactions (three tokens). The CaPU connects with DCIU to issue a certificate to the firm. This unit provides a trust matrix to the blocktrust. Finally, the DCIU provides a token for the visitor firm indicating that the interactions are finished. The experience gained from the simulation indicates that the developed framework embedded with trust is working as expected and it is robust in both the read and write working scenarios.

Blocktrust performance measurement

Throughput and latency are used as indicators for measuring the performance of the blocktrust network. Throughput is the rate at which transactions are committed to the ledger. Latency is the time taken for sending a transaction proposal from application to transaction commitment. An instance of blocktrust is built on the Hyperledger Fabric open BCT platform. The detailed fundamental implementation of BCT on a Hyperledger Fabric has been illustrated by (https://github.com/IBM/build-blockchain-insurance-app.) for an insurance application. The approaches for measurement of the performance of the fabric is reported in (Thakkar et al. (2018); Nasir et al. 2018). The same approaches are used here to implement and measure the performance of the blocktrust. As shown in Fig. 11, the case blocktrust contains an orderer node (Kafka-Zookeeper) and four digital firms, each with two peers. All the peers and orderer are implemented on the IBM SoftLayer data center as an X86-64 virtual machine. Four DFA is used to generate load with 54vCPUs and 128 GB memory. Nodes are connected to the 3Gbps datacenter network.

The detailed specifications of the developed blocktrust are given in Table 1. In the network of a channel with and GoLevelDB StateDb is used. First, the impact of the received transaction rate and block size (number of the transaction in a block) on throughput and latency is studied. Then, the

Table 1 The used configurations for measuring the performance

Number of channels	1
StateDb database	GoLevelDB
Peer resource	32 vCPU, 3Gbp link
Endorsement policy	OR {AND(a,b); AND(a,c); AND(a,d); AND (b,c); AND(c,d)}
Block size	10, 30, 50, 70
Transaction arrival rate	25, 50, 75, 100, 125, 150, 175

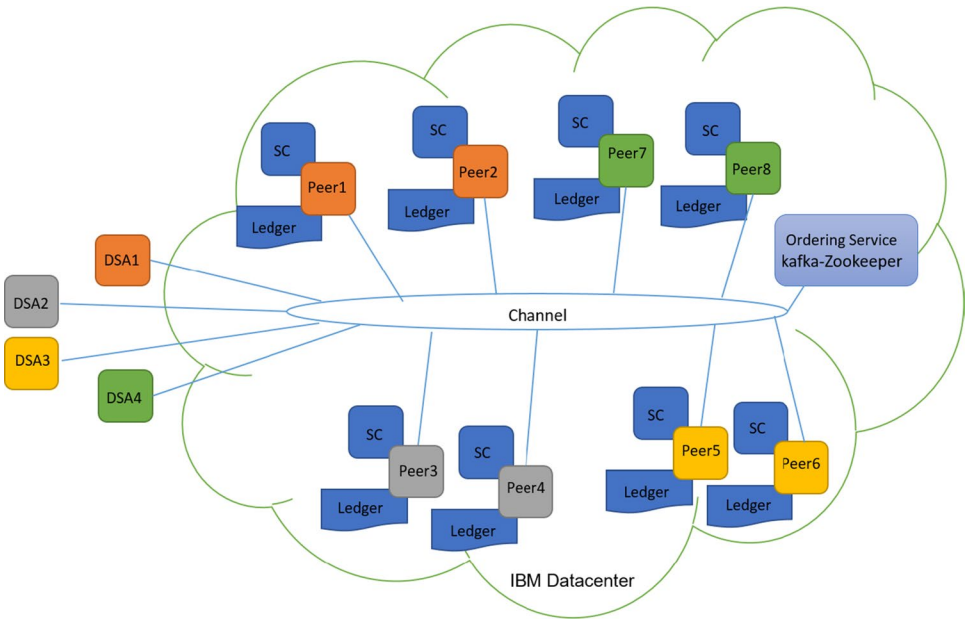

Fig. 11 The case blocktrust configuration

Table 2 Impact of block size and transaction receive rate on throughput

Throughput (Tx/s)	10 TX in a block	30 TX in a block	50 TX in a block	70 TX in a block
25 Tx/s	22	25	28	30
50 Tx/s	51	54	56	57
75 Tx/s	74	77	79	80
100 Tx/s	98	100	113	105
125 Tx/s	122	123	125	130
150 Tx/s	137	138	139	140
175 Tx/s	139	140	140	141

effect of the number of QoSs in transaction throughput and latency is examined.

All the possible configurations based on block size and transaction arrival rate is developed. For the size of the blocks in the ledger, four-block sizes are considered as 10, 30, 50, and 70 transactions on a block and for the transaction arrival rate, six rates are considered as 25,50,75,100,125,150,175 per second. Each piece of the configurations is run for a duration of 100000 s as warmup and then throughput and latency of the network is examined. Table2 denotes the impact of the received transaction and block size on throughput. As shown in the table, it is observed that with an increase in the rate of the arrival transactions, throughput is linearly increased in all the configurations. In all the arrival rates with the increase in the block size, the throughput is also increased. This amount is growing sluggishly by increasing the receive rates.

Tables 3 and 4 denotes the latency of the blocktrust network on each piece of the configurations. The latency of the configurations with 25 Tx/s up to 125 Tx/s arrivals is rising steadily, however, in the configurations with 150 Tx/s and 175 Tx/s arrivals the latency of the network is sharply raised. Among the considered configurations 125Tx/s arrival can be deliberated as an optimal acceptable arrival rate for the case blocktrust network.

In all the prior configurations the transaction holds a trust matrix with only one quality of service as a column and sixteen records as rows. To examine the effect of the transaction size on latency and throughput of the blocktrust new configurations with 5, 10, 15, 20, 25 columns on the trust matrix are developed and some of the previously developed configurations are tested using the trust matrixes.

The 125&70; 100&70; 125&50; and 100&50 (transaction arrival rate & transactions per block) configurations are employed in the experiments. Table 3 demonstrates the throughput and latency of the tested configurations. As shown in the table in the 125 &70 configuration latency of the network is feasible with a maximum of 15 columns in a trust matrix. The 100 &70 configuration is feasible up to 20 columns in the trust matrix while the 125&50 and 100 &50 configurations are feasible up to 25 columns in the trust matrix.

Table 3 Impact of block size and received transaction rate on latency

Latency (ms)	10 TX in a block	30 TX in a block	50 TX in a block	70 TX in a block
25 Tx/s	400	890	960	974
50 Tx/s	384	759	995	1022
75 Tx/s	368	689	755	1248
100 Tx/s	344	687	769	1301
125 Tx/s	310	694	761	1422
150 Tx/s	15,589	13,484	12,326	10,145
175 Tx/s	44,481	41,801	37,489	33,786

Table 4 Impact of trust matrix size on throughput and latency

# of column in a trust matrix	125 Tx/s &70 Tx in a block		100 Tx/s &70 Tx in a block		125 Tx/s &50 Tx in a block		100 Tx/s &50 Tx in a block	
	Throughput	Latency	Throughput	Latency	Throughput	Latency	Throughput	Latency
1 column	130	1400	105	1300	125	760	113	760
5 columns	128	1413	104	1380	121	883	111	840
10 columns	126	1440	98	1440	112	963	107	867
15 columns	124	1476	91	1844	99	1380	101	907
20 columns	21	10,344	84	2340	91	1888	94	1386
25 columns	14	44,810	23	28,480	21	32,475	82	3404

Implication and limitation

Research implication

The current COVID-19 crisis has affected the operations of firms as well as the rest of the economy. For policymakers, however, production continuity forms the bedrock of any arrangement and it should be certainly having been adaptable with social distancing restrictions. Joining a cloud-manufacturing platform is an opportunity for industries to share their capabilities and resources. The overriding principle where being in this platform is concerned is that the provider and consumer remain responsible at all times for its activities and performance of tasks. In the context of CM, capability pool unit, and some extent, service providers also who will need to prepare for ad-hoc requests and support, should consider trust issues concerning their outsourcing arrangements. The proposed blocktrust network for CM can offer the participating firms to control their trust assets, potential tax saving, a long-term plan based on their needs and privacy all in one place. It also incorporates the firm with others in a much-convinced way.

Limitation and future works

The study is perhaps limited by the focus on the CM, however, we articulated this position as the specific gap in the literature that has not been addressed within previous studies. We advocate future research within some specific areas that in our view could yield the greatest contribution to literature. As a direction for future research, other existing and feasible types of trust matrixes can be used to improve the proposed approach. This approach also can be examined in other contexts (e.g. supply chain management), the proposed approach can be revised and implemented on Ethereum and the performances can be compared.

Conclusion

Cloud Manufacturing (CM) is a service-oriented business model to share manufacturing capabilities and resources on a cloud platform. Trust is often talked about as the bedrock of a company's concerns to use the CM model. Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. It is a digital ledger of transactions that is duplicated and distributed across the entire network of computers (i.e. Peer). Blockchain technology changes the models of trust from humans into technology. In the CM model, the collaborator parties never meet each other and the only way of trust is believing in the platform, however, the platform commonly doesn't stay behind the partners' form trust points of view. Similar to the other service provider platforms (e.g. Book Hotel, Rent car) one possible way of trust to the other parties is screening the trust scores of the firm that may be available by the platform to the users. However, these scores are only available to the services providers and there are under the platform control. In CM, the trust scores of the consumers as well as the providers should be recorded and available to facilitate the selection process and bring the choice opportunities to the providers. to avoid manipulations, we also prefer to store, control, manage, and disseminate these trust scores in a distributed way not by a central authority. A distributed blockchain-based trust system called blocktrust is developed in this paper. This trust system is a private peer-to-peer network for preserving the trust scores (TaaC, TaaP) of the involved companies in the CM platform, and it is developed and implemented on Hyperledger fabric. A CM framework embedded with blocktrust is developed to test the applicability of the Blucktrust. Unlike conventional CM frameworks that have only two types of units (One for consumer and provider and another for manager), this framework holds three main digital units and all the units act as a peer of the blocktrust network. The third unit is to act as a digital certification issuing center in the ecosystem, where all new members of the platform must first be forwarded to this unit which acts as a digital firm to inter the ecosystem. The feasibility of the developed framework and blocktrust is tested under different testing scenarios, and the results indicate a robust and credible system.

Form technological perspective, the interaction and integration among the units of the proposed framework should be further verified and evaluated considering more providers, and digital firms. It is also need to consider the trust matrix with more historical interactions and QoS scores. Furthermore, the accomplishment of the entire system with more number of capabilities should be tested. In the blocktrust, we only consider one orderer and four digital firms. However, the more orderer and digital firms exist in real-life ecosystems that might affect the internal and external actions performances of the blocktrust, including throughput and latency. The block size and received transaction rate should be further explored to obtain optimal block size and also optimal range for transaction rate. In the pilot implementation, only four block size and seven transaction arrival rate is considered. Therefore, it is important to consider the scalability and compatibility of the proposed framework and blocktrust itself.

Acknowledgements The author would like to thank HATİCE DENİZ TEMURTAŞ, MELİKE TAKIL, and MERT OLACAK who did a part of this study as their course contribution.

References

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A & Muralidharan, S. (2018). Hyperledger fabric: A distributed operating system for permissioned blockchains. In: *Proceedings of the Thirteenth EuroSys Conference* (p. 30). ACM.

Balta, E. C., Lin, Y., Barton, K., Tilbury, D. M., & Mao, Z. M. (2018). Production as a service: A digital manufacturing framework for optimizing utilization. *IEEE Transactions on Automation Science and Engineering, 15*(4), 1483–1493.

Barenji, A. V., Barenji, R. V., & Hashemipour, M. (2016). Flexible testing platform for employment of RFID-enabled multi-agent system on flexible assembly line. *Advances in Engineering Software, 91,* 1–11.

Barenji, A. V., Li, Z., Wang, W. M., Huang, G. Q., & Guerra-Zubiaga, D. A. (2019). Blockchain-based ubiquitous manufacturing: A secure and reliable cyber-physical system. *International Journal of Production Research, 58*(7), 2200–2221.

Barenji, R. V. (2013). Towards a capability-based decision support system for a manufacturing shop. In: *Working Conference on Virtual Enterprises* (pp. 220–227). Springer, Berlin.

Barenji, R. V., Hashemipour, M., & Guerra-Zubiaga, D. A. (2015). A framework for modelling enterprise competencies: From theory to practice in enterprise architecture. *International Journal of Computer Integrated Manufacturing, 28*(8), 791–810.

Barenji, R. V., Ozkaya, B. Y., & Barenji, A. V. (2017). Quantifying the advantage of a kitting system using Petri nets: A case study in Turkey, modeling, analysis, and insights. *The International Journal of Advanced Manufacturing Technology, 93*(9–12), 3677–3691.

D'Aniello, G., De Falco, M., & Mastrandrea, N. (2020). *Designing a multi-agent system architecture for managing distributed operations within cloud manufacturing* (pp. 1–8). Ahead-of-Print: Evolutionary Intelligence.

Ghasempouri, S. A., & Ladani, B. T. (2019). Modeling trust and reputation systems in hostile environments. *Future Generation Computer Systems, 99,* 571–592.

Helo, P., Phuong, D., & Hao, Y. (2019). Cloud manufacturing–scheduling as a service for sheet metal manufacturing. *Computers & Operations Research, 110,* 208–219.

Khaqqi, K. N., Sikorski, J. J., Hadinoto, K., & Kraft, M. (2018). Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application. *Applied Energy, 209,* 8–19.

Laili, Y., Lin, S., & Tang, D. (2020). Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment. *Robotics and Computer-Integrated Manufacturing, 61,* 101850.

Li, R., Chen, T., Lou, P., Yan, J., & Hu, J. (2019). Trust mechanism of cloud manufacturing service platform based on blockchain. In: *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 15–19). IEEE.

Li, S., Zhang, H., Yan, W., & Jiang, Z. (2020b). A hybrid method of blockchain and case-based reasoning for remanufacturing process planning. *Journal of Intelligent Manufacturing, 9,* 1–11.

Li, T., He, T., Wang, Z., & Zhang, Y. (2020a). SDF-GA: a service domain feature-oriented approach for manufacturing cloud service composition. *Journal of Intelligent Manufacturing, 31*(3), 681–702.

Li, Z., Barenji, A. V., & Huang, G. Q. (2018). Toward a BCT cloud manufacturing system as a peer to peer distributed network platform. *Robotics and Computer-Integrated Manufacturing, 54,* 133–144.

Liu, X. L., Wang, W. M., Guo, H., Barenji, A. V., Li, Z., & Huang, G. Q. (2020). Industrial BCT based framework for product lifecycle management in industry 40. *Robotics and Computer-Integrated Manufacturing, 63,* 101897.

Lu, Y., Wang, H., & Xu, X. (2019). ManuService ontology: A product data model for service-oriented business interactions in a cloud manufacturing environment. *Journal of Intelligent Manufacturing, 30*(1), 317–334.

Nakamoto, S. (2019). Bitcoin: A peer-to-peer electronic cash system. *Manubot*, pp. 1–24.

Nasir, Q., Qasse, I. A., Abu Talib, M., & Nassif, A. B. (2018). Performance analysis of hyperledger fabric platforms. *Security and Communication Networks, 2018,* 1–15.

Selçuk, A. A., Uzun, E., & Pariente, M. R. (2004). Reputation-based trust management for P2P networks. In: *Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid (CCGrid'04)* (pp. 251–258).

Simeone, A., Deng, B., & Caggiano, A. (2020). Resource efficiency enhancement in sheet metal cutting industrial networks through cloud manufacturing. *The International Journal of Advanced Manufacturing Technology, 107,* 1345–1365.

Talhi, A., Fortineau, V., Huet, J. C., & Lamouri, S. (2019). Ontology for cloud manufacturing based product lifecycle management. *Journal of Intelligent Manufacturing, 30*(5), 2171–2192.

Thakkar, P., Nathan, S., & Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric BCT platform. In: *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)* (pp. 264–276). IEEE.

Valdeolmillos, D., Mezquita, Y., González-Briones, A., Prieto, J., & Corchado, J. M. (2019). BCT technology: A review of the current challenges of cryptocurrency. In: *International Congress on BCT and Applications* (pp. 153–160). Springer, Cham.

Viriyasitavat, W., Da Xu, L., Bi, Z., & Sapsomboon, A. (2018). Blockchain-based business process management (BPM) framework for service composition in industry 4.0. *Journal of Intelligent Manufacturing, 31,* 1737–1748.

Wang, Y., Wang, S., Yang, B., Gao, B., & Wang, S. (2020). An effective adaptive adjustment method for service composition exception handling in cloud manufacturing. *Journal of Intelligent Manufacturing, 1,* 1–17.

Yang, X., Wang, S., Yang, B., Ma, C., & Kang, L. (2019). A service satisfaction-based trust evaluation model for cloud manufacturing. *International Journal of Computer Integrated Manufacturing, 32*(6), 533–545.

Yi, H. (2020). A post-quantum secure communication system for cloud manufacturing safety. *Journal of Intelligent Manufacturing, 17,* 1–10.

Yuan, M., Cai, X., Zhou, Z., Sun, C., Gu, W., & Huang, J. (2019). Dynamic service resources scheduling method in cloud manufacturing environment. *International Journal of Production Research, Ahead-of-Print, 4,* 1–18.

Zhang, Y., Xi, D., Yang, H., Tao, F., & Wang, Z. (2019). Cloud manufacturing based service encapsulation and optimal configuration method for injection molding machine. *Journal of Intelligent Manufacturing, 30*(7), 2681–2699.

Zhu, X., Shi, J., Huang, S., & Zhang, B. (2020b). Consensus-oriented cloud manufacturing based on blockchain technology: An exploratory study. *Pervasive and Mobile Computing, 62,* 101113.

Zhu, X., Shi, J., Xie, F., & Song, R. (2020a). Pricing strategy and system performance in a cloud-based manufacturing system built on blockchain technology. *Journal of Intelligent Manufacturing, 31,* 1985.