Journal of Industrial Information Integration 42 (2024) 100730

Contents lists available at ScienceDirect

Journal of Industrial Information Integration

journal homepage: www.elsevier.com/locate/jii

Full length article

Supporting business confidentiality in coopetitive scenarios: The B-CONFIDENT approach in blockchain-based supply chains

Simone Agostinelli [*,1], Ala Arman[2], Francesca De Luzi, Flavia Monti[3], Michele Manglaviti, Massimo Mecella[4]

Sapienza Università di Roma, Rome, Italy

ARTICLE INFO

Keywords:
Blockchain
Industry 4.0
Supply chain management
Coopetition

ABSTRACT

An important issue in *coopetitive* supply chains is ensuring business confidentiality when sharing sensitive information among partner actors. This challenge becomes even more complex in blockchain-based supply chains due to inherent transparency, conflicting with businesses' need to safeguard sensitive information and posing risks to proprietary data. In this paper, we propose an approach based on permissioned blockchains to support transactional business confidentiality in supply chains. The approach is implemented as an open-source platform and evaluated against five non-functional requirements.

1. Introduction

Supply chain management (SCM) aims at efficiently managing the flow of products from suppliers to end-users, ultimately enhancing customer satisfaction while maintaining costs in check [1,2]. To meet the demand for more practical models, cooperative competition structures have been developed within supply chains. These frameworks seek to align supply chain members by implementing strategies that effectively combine cooperation and competition (aka *coopetition*) that offer the potential for mutually beneficial outcomes. Coopetition thus refers to a strategic concept where actors engage in both cooperative and competitive behaviors simultaneously [3]. It typically involves competitors collaborating in certain areas while still competing in other aspects of their operations [4].

Blockchain technology is emerging as a standout solution for coopetitive scenarios [5,6]. In Section 2 we will provide an in-depth overview of this technology, highlighting its central role in our work and explaining why it was chosen based on specific requirements. As an example, the use of blockchain can ensure transparency of transactions within the supply chain even among competing actors — therefore not fully trusting each others, enabling companies to better manage the information flow about exchanged products, and efficiently tracking the usage of products to record the actual performance throughout the supply chain [7]. Traditionally, the management of confidentiality in supply chains has been handled using Certification Authorities-based systems (CA) [8], which authenticate and authorize the involved actors. However, these systems lack the programmability and flexibility that are offered by smart contracts, which can dynamically and conditionally manage access rights based on specific rules, making them particularly suitable for coopetitive environments, where relationships between parties often change. Although there is limited research on the adoption of decentralized technologies with a coopetitive approach at the industrial level, most of it focuses on supply chains [9] and challenges such as complex relationships between organizations, trust, engagement and knowledge exchange [10].

While the use of blockchain technology in the domain of SCM has enhanced data *transparency*, *traceability*, and *verifiability*, it has placed relatively less emphasis on a fundamental challenge in coopetition: ensuring *business confidentiality*.[5] Differently from data privacy [11–14], in which the transactions are completely obfuscated to third parties, business confidentiality aims at limiting access to sensitive, proprietary transactional data to authorized parties while preserving the transparency of the transaction itself, thereby reducing the risk of such data being used against stakeholders [15–22].

In coopetitive supply chains [23], there is often the need for selective information sharing. Hence, different actors may expect varying valuable information, such as pricing details, which must be protected

* Corresponding author.
E-mail addresses: agostinelli@diag.uniroma1.it (S. Agostinelli), arman@diag.uniroma1.it (A. Arman), deluzi@diag.uniroma1.it (F. De Luzi), monti@diag.uniroma1.it (F. Monti), manglaviti.1964287@studenti.uniroma1.it (M. Manglaviti), mecella@diag.uniroma1.it (M. Mecella).
[1] ORCID: 0000-0002-6500-9802.
[2] ORCID: 0000-0001-8418-2099.
[3] ORCID: 0000-0003-3349-7861.
[4] ORCID: 0000-0002-9730-8882.

[5] We mean as business confidentiality the property of some data to be kept confidential in a business-to-business scenario.

https://doi.org/10.1016/j.jii.2024.100730
Received 25 January 2024; Received in revised form 24 October 2024; Accepted 3 November 2024
Available online 22 November 2024

from actors not directly involved in a specific transaction. Therefore, it is important for an actor within the supply chain to safeguard sensitive information and only selectively disclose non-confidential data. Indeed, depending on the responsibilities of the parties involved in the supply chain processes, an approach is needed (in blockchain networks) to regulate access to crucial data, such as product information, shipping details, and quality control records.

Therefore, in a coopetitive scenario, blockchain technology introduces a layer of trust between competing entities [24]. This trust mechanism is essential in coopetitive environments, where collaboration is necessary despite inherent competition. When transactions are generated on a smart contract involving a set of actors, all participants must initially trust each other, as the transaction itself is visible to everyone. However, specific data can be obscured to ensure business confidentiality.

For example, consider three actors in coopetition but aiming for joint processes to achieve common business goals. The first actor, while collaborating, restricts access to its data from the other two, allowing the second actor to view all data (both confidential and non-confidential) and the third actor to view only non-confidential data. This approach leverages blockchain to enable collaboration, decentralize control, and ensure that sensitive business data remains confidential while transaction transparency is maintained among all involved parties.

While smart contracts offer flexibility in access control, they must be carefully designed and audited to avoid vulnerabilities that could compromise confidentiality or the security of transactions. In this context, ensuring the robustness of smart contracts is essential for maintaining the trust needed in coopetitive scenarios. Additionally, blockchain technology enhances trust and transparency not only in transactions but also in related processes like certificate generation and revocation, pivotal for maintaining accountability. This is particularly relevant in coopetitive environments where selective information sharing is key to preserving business confidentiality.

The ability of smart contracts to program specific rules for access to data based on relationships between parties marks a significant advancement over traditional CA-based authorization systems. This flexibility is vital for coopetitive environments. However, despite this advantage, smart contracts still face challenges related to scalability and performance, which stem from the computational resources required to execute them.

Our primary focus is on addressing the issue of business confidentiality in coopetitive supply chains. This led us to the following key research questions:

- **RQ1:** How can blockchain technology be leveraged to provide *business confidentiality* in coopetitive scenarios?
- **RQ2:** To what extent does the exclusive use of blockchain effectively achieve the *business confidentiality* of supply chains?

To address the research questions, we propose the B-CONFIDENT approach and its implementation leveraging smart contracts that can be configured to ensure that confidential data are accessible solely to specific actors with appropriate permissions, effectively addressing concerns regarding business confidentiality in coopetitive supply chain scenarios. Specifically, we developed B-CONFIDENT on top of Quorum [25], one of the reference permissioned blockchain technology derived from Ethereum [26].

The rest of the paper is organized as follows. Section 2 discusses background and related works. Section 3 describes the problem related to business confidentiality in coopetitive scenarios. Section 4 presents a motivating case study based on the agrifood supply chain. Then, Section 5 outlines the design of the proposed approach to ensure business confidentiality among the actors participating in a permissioned blockchain through three sequential steps. Section 6 describes the application of the proposed solution through the case study. Finally, Section 7 reports the evaluation of the proposed approach against several non-functional requirements, and Section 8 concludes the paper by discussing limitations and future works.

2. Background and related works

This section provides an overview of the technology used in our approach. Specifically, Section 2.1 introduces the concept of blockchain, the main architectural characteristics, the concept of smart contracts, and their classification. It also explains the reasons that led us to choose one of the most widely used platforms for their implementation, namely Quorum. In addition to this background, Section 2.2 examines related works that focus on SCM using blockchain technology, highlighting various approaches and their contributions to data confidentiality, transparency, traceability, and verifiability.

2.1. Background

As defined by [27], *blockchain* is a distributed ledger that can record transactions between two (or more) parties efficiently and in a verifiable and permanent way. *Blockchain*, in its original form, is a distributed database technology that utilizes a tamper-proof list of transaction records with timestamps. It finds applications in various domains, including cryptocurrencies like Bitcoin [28]. Its revolutionary potential lies in enabling secure transactions between untrusted parties over a computer network in which nobody is trusted. This is achieved through a combination of peer-to-peer networks, consensus mechanisms, cryptographic techniques, and market mechanisms. The name blockchain is derived from its essential data structure, which is a linked list of blocks. These blocks are distributed across a decentralized network, with each node maintaining the most up-to-date version and within which transaction details can be found. The transaction is an operation between users, represented by the transfer of value. When a new block is added to the blockchain, it is signed using cryptographic methods; a specific *hash function* is applied to the block's content, producing a unique output. Each block is connected to a hash value generated from its own content and the hash value of the previous block in the chain. As a result, hash values not only represent the transactions within blocks but also establish the sequential order of every block. This fundamental mechanism forms the basis of the blockchain's integrity. Any attempt to modify a transaction would alter the hash value of its corresponding block, thereby breaking the chain.

Blockchain offers an additional concept called *smart contract*, which holds great relevance for business processes [29,30]. Business processes often operate according to predefined rules, dictating how they should respond to specific conditions. Smart contracts serve as a means to express these business rules encoded in a programming language. Deploying the code of a smart contract involves a specific type of transaction, and like any other blockchain transaction, it becomes immutable once deployed. By leveraging blockchain technology, untrusted parties can establish trust in the truthful execution of the code.

Blockchains vary in terms of visibility and authorization mechanisms applied to their operations. *Public* (or *permissionless*), and *private* (or *permissioned*) blockchains are different types of blockchain networks, each having its unique characteristics:

- **Permissionless** blockchain: it is a decentralized distributed ledger open to anyone. Anyone can participate in the network, validate transactions, and access the ledger. Permissionless blockchains are typically based on a consensus mechanism like Proof of Work or Proof of Stake [31]. Bitcoin and Ethereum are examples of permissionless blockchains [28,32].
- **Permissioned** blockchain: it is a centralized distributed ledger that is only accessible to restricted users. These users are typically chosen by the organization running the network and access to participate must be granted. In addition, they allow also for selective access to transaction data. For instance, it can be set up so that only certain users are allowed to read some transaction data, while others not. This is useful in a variety of scenarios, such as

in finance, healthcare, SCM (which is exactly our case) and more, where transparency is important but sensitive information needs to be protected. Examples of permissioned blockchain platforms are: Quorum, Hyperledger Fabric, R3 Corda and MultiChain.

Among the sets of *permissioned* and *permissionless* blockchains, the B-CONFIDENT approach relies specifically on permissioned blockchains. This decision, supported by [22], stems from the need for selective network access to address concerns about business confidentiality without relying on public-key cryptography to keep data confidential, which is rather needed in public blockchains such as Ethereum, as it offers full transparency of transaction data.

2.2. Related works

A vast number of studies in the literature focus on the application of blockchain technology in SCM for various purposes such as enhancing security, promoting traceability, and improving data collection. The wide-ranging impacts and benefits of this technology, including reducing data duplication, enhancing supply chain visibility, and preventing counterfeiting, have been extensively discussed [33–39]. These studies highlight the use of blockchain to enhance data transparency, traceability, and verifiability through various mechanisms including smart contracts, Internet of Things (IoT) solutions, and certification authorities. Given the huge amount of studies on this topic, we focus on highlighting only the most recent ones.

Very recently, authors in [40] propose a holistic approach that provides full traceability and transparency by connecting both supply chain actors and product identifications using digital certificates using a blockchain to manage the traceability of the product and validate identities. To create and validate the certificates, and set up the chain of trusts, a public key infrastructure is designed as part of the proposal. Similarly, authors in [39] propose a solution ensuring confidential information sharing between the organizations in the supply chain via a multi-chain framework, which guarantees access to the implemented Hyperledger network via a certification authority access control. However, these solutions rely on a centralized and hierarchical trust model and a central entity for certificate validation. In another study, authors in [41] explore a multidimensional, blockchain-based platform integrating the Internet of Things (IoT) and Building Information Modelling (BIM) to enhance supply chain management in offsite manufacturing. They developed the IoT-BIM-BCT platform characterized by a three-layered SCM Model to address traceability issues and information exchange incompatibilities by focusing on real-time communication and interoperability from planning to installation. Further, authors in [42] propose a blockchain-based traceability model to ensure accurate traceability, transparent information transmission, and secure storage. They introduced a decentralized, tamper-proof blockchain system and a verifiable delegated proof of stake (VDPoS) scheme to tackle centralization and security concerns.

Current solutions do not place so much emphasis on ensuring *business confidentiality* among network participants. To fill this gap, authors in [43,44] have introduced a novel technique that integrates blockchain technology with Multi-Authority Attribute-Based Encryption (MA-ABE) to regulate data access in the scenario of multi-party business operations. The method also leverages the usage of IPFS (InterPlanetary File System) for preserving information artifacts, access regulations and metadata. Smart contracts are used here to manage user attributes, establish access grants to the process participants, and save the connection to IPFS files. The authors implemented such an approach in the CAKE [44] and MARTSIA [43] systems. Similarly, authors in [45] take advantage of permissioned blockchain technology to store traceability data. Particularly, they propose a solution providing transparency to all the partners involved while preserving the confidentiality of their respective critical data in the Multichain blockchain. To preserve business confidentiality, the confidential data are not directly inserted into the blockchain, but only the derived information (i.e., data encryption and the hash before encryption) allowing to prove the unicity, integrity and authenticity of the actual confidential data.

Also, authors in [46] propose a Confidentiality-Minded Framework (CMF) for secure Building Information Modeling (BIM) design collaboration. The CMF is built on two decentralized networks: an IPFS network for storing large-sized design files and a blockchain network to keep and exchange design information. They developed innovative modules to provide: *(i)* an access control model to prevent unauthored access to sensitive data in a transparent blockchain and *(ii)* strategies for design coordination.

All these methods depend on public-key cryptography to meet business confidentiality. The implementation of these approaches may be complex and not feasible in supply chain scenarios involving a high volume of data. In this direction, the next sections will present the B-CONFIDENT solution, which targets to ensure business confidentiality among blockchain participants without delving into public-key cryptography, while at the same time satisfying: *(i)* data transparency, *(ii)* traceability, and *(iii)* verifiability requirements.

3. Problem description

The application of permissioned blockchain technology in coopetitive environments, while upholding *business confidentiality*, is one of the greatest obstacles in supply chain selective information sharing. In the following, we formalize the scenario under study.

In a coopetitive supply chain scenario using a permissioned blockchain, we define U as the set of the involved actors. Among the involved actors in U, we determine different classes of actors K. The set U is therefore defined as $U = \{\bigcup_{\forall k \in K} A_k\}$, where A_k is the set of actors of the same class k. In this scenario, we identify groups of actors $A \subseteq U$ involving actors of different classes $K_A \subseteq K$ who cooperate together on a specific activity of the process, i.e., $A = \{\bigcup_{\forall k \in K_A} A_k\}$. During their interactions, the actors A exchange data fields D also including confidential data.

We then formalize a smart contract as follows:

$$SC = (A, D, CFD) \tag{1}$$

where $A \subseteq U$ denotes the set of actors involved, D refers to the set of data fields, and CFD refers to the set establishing the subsets of accessible confidential data fields. Specifically, A is characterized by *(i)* the subset $W \subseteq A$, which includes actors authorized to write data and are usually of the same class $k \in K_A$, and *(ii)* the subset $R \subseteq A$, which includes the actors authorized to read data. D is divided into non-confidential data NC and confidential data C, with $D = NC \cup C$ and $NC \cap C = \emptyset$. Finally, for each different class of actor $k \in K_A$, subsets of confidential data $C^k \subseteq C$ are defined to characterize the set $CFD = \{\bigcup_{\forall k \in K_A} C^k\}$, which contain different sets of confidential data fields for each different class.

Given a smart contract $SC = (A, D, CFD)$, a transaction T generated after its execution is defined as follows:

$$T = (w, \mathcal{I}(D), \mathcal{CF}) \tag{2}$$

where, $w \in W \subseteq A$ is the actor generating the transaction (the one that writes data), $\mathcal{I}(D)$ is an instance of the data fields D, and $\mathcal{CF}$ is the set defining access of the actors to confidential data. Each data field $d \in D$ has its own domain $\mathcal{D}(d)$ and $\mathcal{I}$ represent an assignment of each data field d to a value $v \in \mathcal{D}(d)$. The set $\mathcal{CF} = \{(r, C^k) \mid \forall k \in K_A \wedge r \in R^C \subseteq A_k \wedge C^k \in CFD\}$ defines and regulates access to confidential data fields C, where $C^k \in CFD$ represents the subset of confidential data field accessed by the actor r of class k, and $R^C \subseteq A_k$ is the set of actors of the same class k authorized to access confidential data fields C. Thus, CF contains tuples specifying that each authorized actor $r \in R^C \subseteq A_k$ of class k can access their designated subset of confidential data C^k.

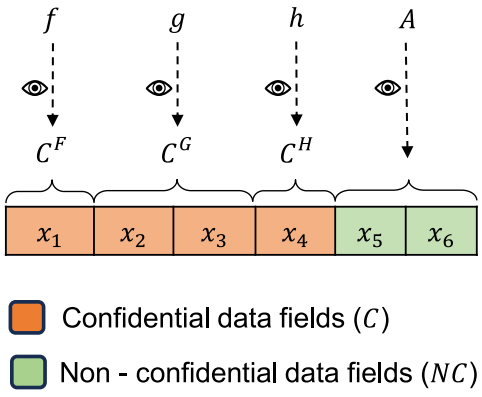

Fig. 1. An example of instantiation of the business confidentiality property.

Definition 3.1 (*Business Confidentiality*)**.** Given a transaction $T = (w, \mathcal{I}(D), \mathcal{CF})$ generated by the execution of a smart contract $SC = (A, D, \mathcal{CFD})$. *Business confidentiality* is defined as the property that guarantees that:

- each actor $a \in A$ is allowed to access the non-confidential data fields NC and their associated values $\mathcal{I}(NC)$;
- each actor $r \in R^C$ of class k is restricted to access its designed subset of confidential data fields C^k as specified by $\mathcal{CF}$, and their associated values $\mathcal{I}(C^k) \subseteq \mathcal{I}(C)$.

Fig. 1 shows, as an example, an instantiation of the business confidentiality property. In this example, $A = \{e, f, g, h, i, j\}$ represents the set of actors of six different classes $K_A = \{E, F, G, H, I, J\}$ involved in the smart contract SC, $D = \left\{\bigcup_{i=1}^{6} x_i\right\}$ is the set of managed data, and $\mathcal{CFD} = \left\{C^F = \{x_1\}, C^G = \{x_2, x_3\}, C^H = \{x_4\}\right\}$ is the set defining the subsets of accessible confidential data and $R^C = \{f, g, h\}$ is the set of actors accessing confidential data. As defined in Definition 3.1, if a transaction $T = (e, \mathcal{I}(D), \mathcal{CF})$ with $\mathcal{CF} = \{(f, C^F = \{x_1\}), (g, C^G = \{x_2, x_3\}), (h, C^H = \{x_4\})\}$ exists, it means that:

- f is the only one that can access x_1 and read its stored value $\mathcal{I}(x_1)$;
- g is the only one that can access both x_2 and x_3, thus also reading their stored values $\mathcal{I}(x_2)$ and $\mathcal{I}(x_3)$;
- h is the only one that can access x_4 and read its stored value $\mathcal{I}(x_4)$;
- i and j can access only the non-confidential data fields x_5 and x_6, thus also reading their stored values $\mathcal{I}(x_5)$ and $\mathcal{I}(x_6)$. This is also true for f, g, and h.

In general, a transaction T generated by a given actor $a \in A$ by executing SC will guarantee access to (in other words, will guarantee the possibility to read) the instance of non-confidential data fields $\mathcal{I}(NC)$ to all the actors A, and the access to the instance of confidential data fields $\mathcal{I}(C)$ to the actors R^C, regulated by $\mathcal{CF}$.

4. The case study

In this section, we present a motivating case study from the agrifood industry to illustrate how permissioned blockchain technology can be leveraged to provide business confidentiality in coopetitive scenarios and to frame our key research question (**RQ1**) introduced in Section 1. The application of blockchain in agrifood industry supply chains, particularly in the wine sector, deserves special attention for the following reasons:

- *Economic contribution*: the agrifood industry, including the wine sector, is a significant driver of the global economy. It contributed approximately USD 1.264 trillion to the U.S. economy in 2021, about 5.4% of the total GDP [47]. In 2018, the European Union (EU) did really well in selling agricultural and food products to other countries, making a surplus of EUR 21 billion. Especially the Italian wine sector holds a prominent position on the global stage, with exports amounting to 6.3 billion Euros and 21.3 million hectoliters in 2019 [48];
- *Complex supply chain*: the agrifood industry is intricate and involves various stakeholders. A wine supply chain, in particular, includes several actors including grape growers, wine producers, certifying authorities, agronomists, distributors, fillers, resellers, and consumers, each with distinct responsibilities. This diversity of roles underscores the necessity for precise tracking and tracing of elements throughout the supply chain. Our focus is specifically on the agrifood supply chain of the wine, using it as a case study (see Fig. 2 for a high-level example);
- *Coopetitive context*: in the wine sector, coopetitive environment demands a balance between collaboration and competition, emphasizing the role of business confidentiality. Alongside the increasing importance of transparency, protecting proprietary information is vital for wine producers [49].

Fig. 2 presents a BPMN (Business Process Modeling and Notation)[6] choreography diagram showing the interaction in the wine supply chain serving as a running example, between the following participants: grape grower, wine producer, filler, distributor, certifying authority, agronomist, resellers, and consumers. A BPMN choreography diagram defines how participants coordinate their interactions. The focus is therefore not on the work being done but rather on the exchange of information between the involved parties. This means that a BPMN choreography acts as a contract between all involved parties. Once this contract is defined, each party can transform it into their private process, or all parties can work together to transform the choreography into a collaboration diagram. Such characteristics perfectly fit the requirements needed for the B-CONFIDENT approach.

Below, we describe the interactions between the aforementioned participants.

- **Agronomist**: The agronomist interacts with both the grape grower and the certifying authority. In the first case, the agronomist sends all collected environmental data to the grape grower. This data is essential for assessing the quality and suitability of the vineyard. The agronomist also communicates the results of quality controls to the certification authority.
- **Grape Grower**: The grape grower collaborates with the wine producer by sending data on the grape harvest and also sends destination data to the filler. For verification purposes, the grape grower interacts with the certification authority to confirm that the provided harvest data complies with the standards.
- **Wine Producer**: The wine producer records the list of products used and communicates it to the filler. Additionally, they collaborate with the certification authority to obtain certification for the list of products used in production.
- **Certifying Authority**: The certification authority plays a crucial role in maintaining the quality and authenticity of the wine production process. It interacts with all actors in the supply chain to receive quality control data from the agronomist, harvest data from the grape grower, wine product lists from the wine producer, batch numbers from the reseller, and transport data from the distributor.

[6] Business Processing Modeling Notation is a standard language, proposed by the Object Management Group (OMG), to design business processes. We refer here to the last release of BPMN, namely BPMN v2.0 – http://www.omg.org/spec/BPMN/2.0/.

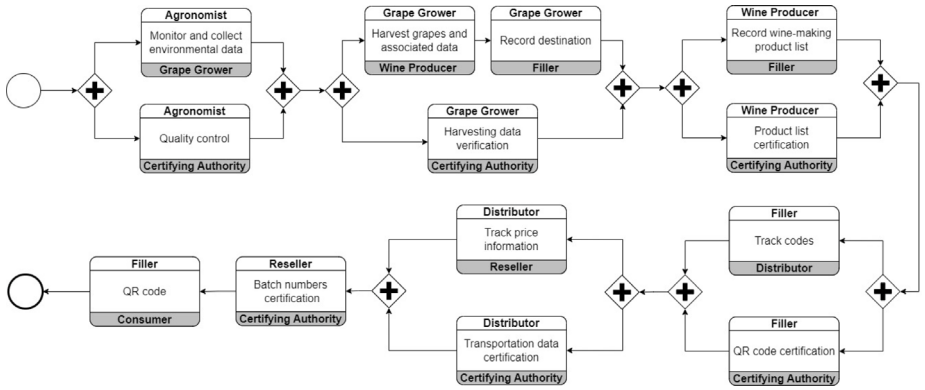

Fig. 2. The BPMN choreography of a wine supply chain.

- **Distributor**: The distributor collaborates with the reseller by providing pricing information, interacts with the filler to receive tracking codes, and sends transport-related data to the certification authority to ensure that the wine is transported while maintaining product integrity. Considering the business confidentiality issue (cf. Section 3), in a situation where a single wine distributor sells his wine to multiple resellers, it may expect a different sale price among the different resellers. It is crucial to hide the price information to the resellers not involved in a particular sale. Let us consider an example scenario involving the distributor class interactions (cf. the BPMN choreography in Fig. 2). Suppose we have a set of actors involving two distributors (d_1 and d_2), two resellers (r_1 and r_2), and a certification authority ca_1 (i.e., $A = \{A_{dist} = \{d_1, d_2\}, A_{res} = \{r_1, r_2\}, A_{cert} = \{ca_1\}\}$), which are involved in a smart contract SC managing data D regarding sales (i.e., $salePrice$, $productName$, $amount$, $resellerName$ and $saleDate$) and transportation (i.e., $batchNumber$ and $destination$), with accessible confidential data defined as $CFD = \{C^{res} = \{salePrice, resellerName\}, C^{cert} = \{batchNumber, destination\}\}$. When the distributor d_1 generates a transaction, it assigns values to the data field depending on the specific domain D of the various data, i.e., $\mathcal{I}(D) = \{salePrice = 50€, productName = chardonnay, amount = 7, resellerName = Caprigliano, saleDate = 23/01/2024, batchNumber = 32, destination = borgogna\}$. Also, it defines the CF set s.t. $CF = \{(r_1, C^{res}), (ca_1, C^{cert})\}$, meaning that only the reseller r_1 can access the *price* data and only the certification authority ca_1 can access the $batchNumber$ and $destination$. The other involved actors, i.e., d_2 and r_2, can only access the non confidential data, i.e, $productName$, $amount$ and $saleDate$. Such a condition is pivotal in the context of blockchain technology and B-CONFIDENT proposes a solution (detailed in Section 5) to maintain sensitive data, such as sales information, confidential.
- **Filler**: The filler generates and tracks the codes, which are then communicated to the distributor. Additionally, they collaborate with the certification authority to certify the QR codes. Finally, they also provide the QR codes to the consumer, enabling product monitoring.
- **Reseller**: The reseller interacts with the certifying authority by transmitting batch numbers for their certification and with the distributor, from whom they receive pricing information.
- **Consumer**: The consumer is the final recipient and interacts only with the filler through the QR code, which allows tracing the product through the entire production process.

B-CONFIDENT envisions a tailored smart contract for each actor involved in the wine supply chain, as described in Section 6.

5. The B-CONFIDENT approach

The proposed approach has been conceptualized and designed to enable *business confidentiality* among actors participating over a permissioned blockchain while at the same time ensuring data transparency, traceability, and verifiability. The approach consists of 3 operational stages to be applied in sequence: *(i)* Smart Contract Deployment, *(ii)* Transaction Initiation, and *(iii)* Supporting Business Confidentiality, as depicted in Fig. 3. Notably such steps are useful not only to tackle the technical requirements but also to serve as our answer to **RQ1**.

Smart contract deployment. A BPMN choreography diagram serves as the starting point of this stage. Indeed, the total number of deployed smart contracts depends on the number of different classes of actors involved in the BPMN choreography diagram (i.e., $|K|$). When a smart contract is deployed, the identifiers of actors of different classes participating in the choreography (i.e., $A \subseteq U = \{\bigcup_{\forall k \in K} A_k\}$) are placed in a list called *PrivateFor*. The identifiers of the actors are their wallet addresses. In case a smart contract is deployed as public among all the blockchain participants, the set A coincides with the set U. Given a smart contract $SC = (A, D, CFD)$ as defined in Section 3, we foresee the use of both getter and setter functions to read and write both NC and C data fields. Specifically, for confidential data C, the number of get and set functions is determined by:

(1) $\forall C^k \in CFD, \exists w \in W \implies \exists\ set(C^k) : w \rightarrow \mathcal{I}(C^k)$
(2) $\forall C^k \in CFD \implies \exists\ get() : \forall r \in R^C \subseteq A_k \rightarrow \mathcal{I}(C^k)$

While, for non-confidential data NC, it holds:

(3) $\forall nc \in NC, \exists w \in W \implies \exists\ set(nc) : w \rightarrow \mathcal{I}(nc)$
(4) $\forall nc \in NC \implies \exists\ get() : \forall r \in R \rightarrow \mathcal{I}(nc)$

Transaction initiation. Actors utilize their smart contracts to initiate transactions, securely updating the blockchain state through predefined logic and representing specific actions within the supply chain. Transactions can happen in both reading and writing modes, but only writing transactions will be recorded on the blockchain, as they modify its state. When a transaction T is initiated by an originator $w \in W$ and broadcasted to actors A within the *PrivateFor* list, it is initially represented as a block and consequently validated. It is worth noticing that transactions occurring on a smart contract, are visible only by explicitly declared actors of the *PrivateFor* list.

Supporting business confidentiality. Concerning transaction T generated from the execution of a smart contract SC, to address the business confidentiality concern outlined in Section 3, two lists are considered, i.e., *PrivateFor* and *ConfidentialFor* lists. The *PrivateFor* list is derived from the SC definition and consists of actors $A \subset U$ involved in the smart contract, while the *ConfidentialFor* list encompasses the subset of actors $R^C \subset R$ that can access confidential data C of transaction T as

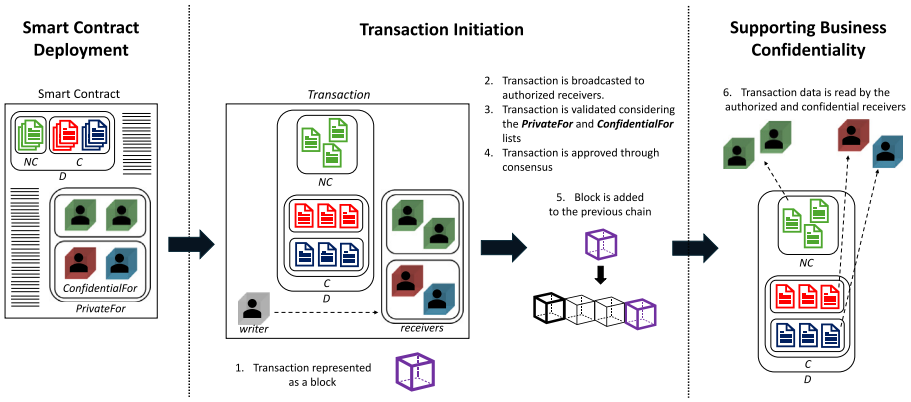

Fig. 3. Implemented approach with transaction data: C and NC fields.

dictated by $\mathcal{CF}$. During the validation of transaction T of SC with the set D of data fields, the *PrivateFor* list is checked to ensure that only the actors A access the NC fields in T while $U \setminus A$ cannot. Then, the *ConfidentialFor* list is examined to guarantee that solely the confidential R^C actors have access to the confidential data fields C in T. Access to NC and C fields, in both writing and reading mode, is regulated by the getter and setter functions as defined above.

6. B-CONFIDENT in action

In this section, we provide some implementing details. We then show the technical steps enacted to develop the B-CONFIDENT approach as a real implemented platform, then instantiated on the use case scenario of Section 4. This also serves for demonstration purposes of our artifact.

B-CONFIDENT is entirely built on the GoQuorum[7] (a version of Quorum) blockchain. Quorum is a permissioned blockchain platform that is based on a modified version of the Ethereum protocol, which allows the use of smart contracts and the execution of decentralized applications. The smart contracts are implemented using Solidity,[8] an Ethereum native programming language. Due to these reasons, we opted to use Quorum which aligns perfectly with our specific needs. GoQuorum particularly, is a lightweight fork of the Geth[9] client and implements Proof of Authority (PoA) consensus mechanisms, specifically through the IBFT[10] (Istanbul Byzantine Fault Tolerance) variant, which operates when participants know each other and have a certain level of mutual trust, such as in a permissioned consortium network. PoA consensus protocols have faster block times and a much greater transaction throughput than the Proof of Work (PoW) or Proof of Stake (PoS) protocols, with a group of nodes in the network acting as validators in the GoQuorum PoA consensus (QBFT and IBFT). The *privateFor* transaction field is managed by Tessera,[11] a private transaction manager that guarantees privacy by encrypting transaction data and securely exchanging payloads between Tessera nodes. Tessera operates independently of private keys and can utilize an enclave for cryptographic functionality. In our implementation, GoQuorum is configured as a zero-gas network. While gas is the unit that measures the computational effort required for transaction execution, GoQuorum removes the need for gas fees unless explicitly enabled. Specifically, transactions consume computational resources (with associated costs), where the cost unit is

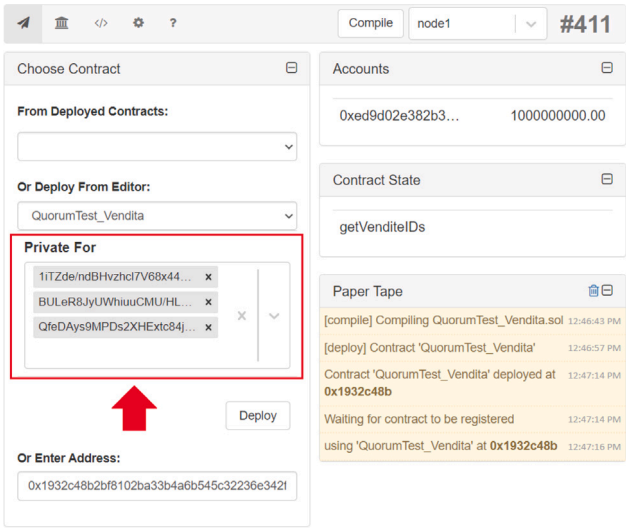

Fig. 4. Deployment of a private smart contract in the Quorum-based implemented approach.

gas, priced at the gas price per unit. The transaction cost is the gas used multiplied by the gas price. In public Ethereum networks, the transaction cost is paid by the sender in Ether and received by the miner or validator, whereas in private networks like GoQuorum, validators are network participants and no gas incentive is required. This is beneficial for private networks where validators are known and there is no need for economic incentives to process transactions. However, this does not exclude an estimate in terms of gas for transactions, to assess the effort required by the network to execute and validate them. In our case, it was not relevant since no tokens were planned for the network.

The first step of the approach is the **Smart Contract Deployment**. Since the number of different classes of actors involved in the BPMN choreography diagram depicted in Fig. 2 is eight (i.e., $|K| = 8$), the total number of smart contracts deployed will also be eight. The details of these smart contracts are discussed as follows.[12]

The agronomist actor engages with its smart contract in supply chain activities for wine health, cultivation data and quality control data. The grape grower actor uses its smart contract for grape-related rules and vineyard data. The wine producer actor oversees wine transformation via smart contracts, monitoring production with the filler.

[7] Cf. https://docs.goquorum.consensys.net/concepts/consensus.
[8] Cf. https://soliditylang.org/.
[9] Cf. https://geth.ethereum.org/.
[10] Cf. https://docs.goquorum.consensys.io/concepts/consensus.
[11] Tessera is an open-source private transaction manager developed under the Apache 2.0 license and written in Java, cf. https://docs.tessera.consensys.io/.

[12] All the artifacts (platform and smart contracts of the case study) are accessible at https://tinyurl.com/bconfident-code, in order to guarantee complete repeatability of the research outcomes.

```
1  function setSaleData(string memory _salePrice, string memory _productName, string memory _amount, string memory _resellerName,
   ↪  string memory _saleDate, address[] memory confidentialFor) public {
2          //check if msg.sender is in privateFor list
3          require(authorized[msg.sender]); //msg.sender = address of the caller
4          //set sale data fields
5          sales[idSaleSerial].salePrice = _salePrice;
6          sales[idSaleSerial].productName = _productName;
7          sales[idSaleSerial].amount = _amount;
8          sales[idSaleSerial].resellerName = _resellerName;
9          sales[idSaleSerial].saleDate = _saleDate;
10         //each address specified in the confidentialFor array is authorized to get the sale data
11         for(uint i=0; i<confidentialFor.length; i++){
12                 if(confidentialFor[i] != address(0)){ //address(0) = null address
13                         allowedAddressesSales[idSaleSerial][msg.sender] = true;
14                         allowedAddressesSales[idSaleSerial][confidentialFor[i]] = true;
15                 }
16         }
17         idSaleSerial++; //increments the serial number of the sales made.
18 }
```

Listing 1: Solidity code snippet for the `setSaleData` function.

Fig. 5. Initiating a *write* transaction: schema (a) and values (b).

Distributor, filler, and reseller actors utilize their smart contracts for shipping, filling, and distribution processes, ensuring data integrity and quality control. In addition, the certifying authority engages with all the smart contracts in the blockchain network to verify wine-related information authenticity, ensuring accurate and transparent certification processes. The consumer actor engages with the filler to retrieve product information, including lot number, origin, grape variety, and certifications, using a QR code. Fig. 4, as an example, illustrates the deployment of a smart contract over the Quorum blockchain, incorporating the *PrivateFor* list useful for the next stages of the implemented approach.

The **Transaction Initiation** comes into play when any actor engages with its respective smart contract, thus generating a transaction. This transaction will be visible to all the actors defined in the *PrivateFor* list of that specific smart contract, namely actors A. These actors can interact with their dedicated smart contracts through a designated user interface that allows to use getter and setter functions as regulated by CF.

For **supporting business confidentiality**, Listing 1 illustrates the process of a *write* transaction using as an example the *setSaleData* function of the smart contract between the distributor, reseller and certification authority actors. This function takes as input the sale data fields (i.e., *_salePrice*, *_productName*, *_amount*, *_resellerName*, *_saleDate*) and the *confidentialFor* list as an array. It begins by checking whether the transaction originator (i.e., *msg.sender*) is authorized to execute the transaction, verifying its inclusion in the *PrivateFor* list (line 3). The function proceeds only if the originator is authorized. Subsequently, lines 5 to 9 set the values of the sale data fields, while lines 11 to 16 grant read access to *confidential data* fields $C \subset D$ to each actor in the set $R^C \subset R$ of *confidentialFor* list. Fig. 5 provides an example of initiating a *write* transaction in the `setSaleData` function through a dedicated user interface.

In a similar way, Listing 2 shows how a *read* transaction is managed when the *getSaleData* function is used. The function begins by checking if the function caller *msg.sender* is authorized to access the data (line 3), i.e., it is included in the *privateFor* list. It is then checked if the unique identifier of the caller is set in the *setSaleData* function as *allowedAddressesSales* (line 5), to ensure that only actors $R^C \subset R$ can read the confidential data C (line 6). Conversely, an error is returned to the function caller (line 8).

In the following, for the sake of understanding, we instantiate the example scenario of Section 4 in which we have two distributors (d_1 and d_2), two resellers (r_1 and r_2), and a certification authority ca_1 where d_1 generates the following transaction $T = (d_1, I(D), CF)$ by executing the smart contract $SC = (A, D, CFD)$:

- $A = \{A_{dist} = \{d_1, d_2\}, A_{res} = \{r_1, r_2\}, A_{cert} = \{ca_1\}\} \implies$ the *privateFor* list contains the wallet addresses of actors $A = \{d_1, d_2, r_1, r_2, ca_1\}$; specifically $W = \{d_1, d_2\}, R = \{r_1, r_2, ca_1\}$, $R^C = \{r_1\}$.
- $CFD = \{C^{res} = \{salePrice, productName, amount, resellerName, saleDate\}\}$[13];

[13] We omitted C^{cert} from the set CFD to improve understandability of the example and to maintain consistency with the presented section.

```
function getSaleData(uint256 _saleId) public view returns(string memory, string memory, string memory, string memory, string memory) {
        //check if msg.sender is in privateFor list
        require(authorized[msg.sender]); //msg.sender = address of the caller
        //returns the sale's data if the msg.sender address was previously authorized
        if (allowedAddressesSales[_saleId][msg.sender] == true) {
                return (sales[_saleId].salePrice, sales[_saleId].productName, sales[_saleId].amount,
                    sales[_saleId].resellerName, sales[_saleId].saleDate);
        } else {
                revert("User not authorized");
        }
 }
```

Listing 2: Solidity code snippet for the `getSaleData` function.

  - applying (1): $C^{res}, d1 \implies setSaleData(C^{res}) : d_1 \rightarrow \mathcal{I}(C^{res})$
  - applying (2): $C^{res} \implies getSaleData() : r_1 \rightarrow \mathcal{I}(C^{res})$

- $CF = \{(r_1, C^{res})\} \implies$ the *confidentialFor* list contains the wallet address of actors $R^C = \{r_1\}$;

This means that:

- when d_1 call *setSaleData* function for a sale, r_1 is granted access to data fields of that sale $\mathcal{I}(C^{res})$;
- when r_1 call *getSaleData* function of a particular sale it can read the data fields only if it was previously authorized. Since r_1 was granted access in the previous step, therefore the sale data fields $\mathcal{I}(C^{res})$ are safely returned to r_1;
- when r_2 call *getSaleData* function of the sale between d_1 and r_1, an error is returned because it was not previously authorized.

It is worth noticing that we just focused on accessing C fields, as this is the primary objective of the paper. However, the implemented approach encompasses additional functions specifically designed for writing/reading also NC fields, which are not included here for the sake of simplicity (but still accessible in the source code of the smart contracts). B-CONFIDENT has been implemented as a standalone platform available for download at https://tinyurl.com/bconfident-code.

7. Evaluation

In this section, we aim to understand the general quality of the B-CONFIDENT approach (and its implementation) in addressing **RQ2**. To this end, we analyze five non-functional requirements of the artifact: effectiveness, feasibility, reliability, scalability, and robustness. Specifically, to evaluate the *effectiveness* of our approach, we compare it with CAKE [44], an approach with similar objectives as ours. Then, to assess its *feasibility*, we perform a controlled experiment involving real users exploiting the use case of Section 4. Lastly, we perform many synthetic experiments to assess the *reliability*, *scalability*, and *robustness* of B-CONFIDENT evaluating the performance of the underlying permissioned blockchain network.

Evaluating the effectiveness. To investigate the *effectiveness* of the B-CONFIDENT approach, we compared it with the CAKE approach [44] since it achieves business confidentiality in the context of multi-party business processes. Specifically, it combines blockchain technology and Attribute-Based Encryption (ABE) to control data access among network participants which we would like to neglect. Differently from us, whereas B-CONFIDENT is equipped with a graphical user interface (GUI), CAKE operates entirely through a command-line interface (CLI). In addition, CAKE is openly available on Github and provide two implementations (one for the EVM and one for the Algorand Virtual Machine (AVM)), distributed within Docker containers. Notably, since the difference in interface types might have influenced user responses, we implemented two key actions to mitigate potential confounding effects:

- *Preliminary training*: although the selected users were unaware of both B-CONFIDENT and CAKE before the experiment, we provided equivalent training sessions for both interfaces to ensure participants had a proper understanding of the features and could use each interface without bias due to unfamiliarity.
- *Qualitative feedback*: during the testing, we collected qualitative feedback to capture user experiences with the interfaces, which helped us in identifying any difficulties related to the interface type. At this stage, we have not encountered any feedback indicating that the interface type systematically influenced the responses.

To assess the effectiveness, we conducted a user study based on the use case presented in Section 4 with 40 different MSc students enrolled in the Enterprise Information Systems (EIS) course at Sapienza Università di Roma. Specifically, we denote two distinct groups of users made up of 20 people each: *p1* and *p2*. It is worth noticing that all the EIS students involved in the user study can be considered expert users in business process modeling and security.

We investigated the following experimental hypothesis **H1**: *Employing the B-CONFIDENT approach, thus neglecting asymmetric cryptographic functionalities, is easier than employing the CAKE approach which requires manually ciphering confidential data among actors to achieve business confidentiality*.

Then, to support or reject **H1**, a *between-subject* approach was used, i.e., each user in *p1* and *p2* respectively was assigned to a different experimental condition, related to the exclusive use of B-CONFIDENT (*c1*) or CAKE (*c2*) to perform the required steps for the enactment of the use case. Any user in *p1* (*p2*, respectively) was preliminarily instructed about the functionalities of B-CONFIDENT (CAKE, respectively) through a short training session, as described above. Notice that we selected users who were unaware of the use of both B-CONFIDENT and CAKE before the start of the experiment. We evaluated the validity of **H1** by asking any student who completed the user study the following three questions:

- *Q1*: The process of writing data in the blockchain is a complex task. Do you agree?
- *Q2*: Making data confidential among actors involved in a smart contract is a complex task. Do you agree?
- *Q3*: Once a transaction has been generated over the blockchain, reading data stored in it is a complex task. Do you agree?

Questions are rated with a 7-point average numerical scale structured as follows: 1 (“Strongly Disagree”), 2 (“Disagree”), 3 (“Somewhat Disagree”), 4 (“Neither Agree nor Disagree”), 5 (“Somewhat Agree”), 6 (“Agree”), 7 (“Strongly Agree”). We kept the same difference (numerical 1) between subsequent points of the scale, as suggested by [50]. The choice to employ a 7-point scale (rather than a 5-point scale) is supported by the findings of Sauro [51], which states that in case of a questionnaire consisting of few questions *“having seven points tends to be a good balance between having enough points of discrimination without having to maintain too many response options”*.

Table 1
Effectiveness of B-CONFIDENT: *p-values* associated to each question.

Q1		Q2		Q3	
B-CONF	*CAKE*	*B-CONF*	*CAKE*	*B-CONF*	*CAKE*
1	1	1	3	1	1
1	1	2	3	1	1
1	1	2	3	1	2
1	2	2	3	1	2
1	2	3	3	2	3
1	2	3	3	2	3
1	3	3	3	2	3
2	3	3	3	2	3
2	3	3	3	2	3
2	3	3	4	2	3
2	3	3	4	2	3
2	3	3	4	2	3
2	3	3	4	3	3
2	3	4	4	3	3
3	4	4	4	3	3
3	4	4	4	3	3
3	4	4	5	3	3
4	4	4	5	3	4
4	4	4	5	3	4
4	4	5	6	4	4
p-value	0,0305558	p-value	0,0304094	p-value	0,0282627

To validate *Q1*, *Q2* and *Q3* we performed a comparison of the rates obtained from the questionnaire, respectively in the cases of *c1* and *c2*. Specifically, for each question, we employed a *2-sample t-test* with a 95% confidence level to determine whether the means between the two distinct populations (i.e., independent groups *p1* and *p2*) involved in *c1* and *c2* differ. Before running the 2-Sample t-test, we first exploited the Kolmogorov Smirnov Statistic (KS Test) to establish the normality of the distribution of the collected data [52], and then we checked that the variances and standard deviations in both groups were approximately equal [51].

Finally, we measured the level of statistical significance by analyzing the resulting p-value. We remind that a p-value ≤ 0.05 is considered to be statistically significant, while a p-value ≤ 0.01 indicates that there is substantial evidence in favor of the experimental hypothesis. The results of the analysis are summarized in Table 1.

It appears evident that the experimental hypothesis *H1* is supported for both *Q1*, *Q2*, and *Q3* since p-values are lower than 0.05, but since they are greater than 0.01, it means that there is not too much statistical evidence in the difference of the means of both populations.

Q1 and *Q3* results revealed that users noted a distinction in the way data are stored between B-CONFIDENT and CAKE, and also in the access procedure. This distinction arises from the fact that in the context of B-CONFIDENT, the payload data of a transaction is directly stored on the permissioned blockchain, whereas in CAKE, it is stored on the IPFS (InterPlanetary File System). Smart contracts are indeed utilized in CAKE to generate links to IPFS files for reducing gas expenses associated with public blockchains.

In addition, concerning *Q2* users observed a distinction in handling business confidentiality. More precisely, CAKE offers a fine-grained specification for granting access permissions to process actors, in contrast to B-CONFIDENT, which employs a more general or coarse-grained specification. This means that users in CAKE can establish access controls with a high degree of accuracy and granularity, enabling them to encrypt individual data fields for precise and detailed management of who can access specific data and under what conditions. In contrast, B-CONFIDENT may offer simplicity in most situations.

Regarding the experiments' findings, we claim that their validity is bound to the experiment's settings. For instance, performing a further experiment that includes more users and the application of a second confidence level (e.g., set to 99%) could support more substantial evidence of the results.

Evaluating the reliability, scalability and robustness. To assess the *reliability*, *scalability*, and *robustness* of our solution in a controlled and reproducible manner we leveraged Hyperledger Caliper,[14] an open-source blockchain benchmarking tool specifically designed to evaluate the performance and the scalability of blockchain networks, including Ethereum-based blockchains [53], in which Quorum is comprehended. Using it, we measured the network's throughput for reading and writing transactions.

Our evaluation involved conducting tests on a local Quorum network consisting of 7 nodes, all deployed as Docker containers on a single machine. We specifically select the smart contract between the distributor, reseller and certifying authority actors as the target smart contract due to its complexity and the varied range of operations it executes. In Fig. 6, we can observe the performance of the Quorum blockchain under different rounds, each consisting of 5000 transactions but with varying *send rates* (representing the number of transactions performed per second).

The throughput parameter indicates the number of transactions per second that Quorum successfully managed without any failures. Note that an empty column in the chart represents the point at which the network becomes congested, indicating a failure in guaranteeing reliability under a certain load. We observed that increasing the number of transactions per round led to a decrease in the send rate at which the blockchain failed. This suggests that the load of transactions during a round represents the primary bottleneck for the network. We also observed similar behavior when benchmarking reading transactions, as shown in Fig. 7. Additionally, during our tests, we pointed out that the blockchain tended to reach failure states when more than 100,000 consecutive transactions were performed within a short time frame (with intervals between rounds not exceeding 5 s).

In practical contexts, the collected results are not worrisome. Indeed, we need to take into account we are working in a coopetitive environment where processes rely on human involvement, and tasks often take minutes or even hours to complete. These durations do not pose any limitations in the employment of our solution. However, it is worth noting that the results we found might be problematic if we were operating in a real-time constrained context.

Evaluating the feasibility. To evaluate the *feasibility* of the B-CONFIDENT approach, we conducted a user study with Caprigliano,[15] an Italian company specialized in grape cultivation and winemaking. We actively involved 10 employees from Caprigliano to provide feedback on our system. We demonstrated the running platform based on the use case of Section 4 to the employees and conducted several runs to gather their insights. The results of the user study confirmed that the transaction times were not critical and, importantly, highlighted the potential benefits that the users could experience through the system, such as improved *collaboration* and *interoperability*.

8. Conclusion

In this paper, we have introduced the B-CONFIDENT approach and its implementation, designed to interact with permissioned blockchains to offer simultaneous and shared management of (confidential) information in coopetitive supply chain scenarios.

A limitation of our approach lies in its reliance on permissioned blockchain since it prioritizes access control by restricting participation to known actors in contrast with the transparent nature of public blockchains. Therefore, the B-CONFIDENT approach cannot be directly employed on public blockchains to address business confidentiality concerns.

As future work, the adoption of blockchain technology in the agricultural sector represents an exciting opportunity to establish a

[14] Cf. https://hyperledger.github.io/caliper/.
[15] Cf. https://www.caprigliano.com/.

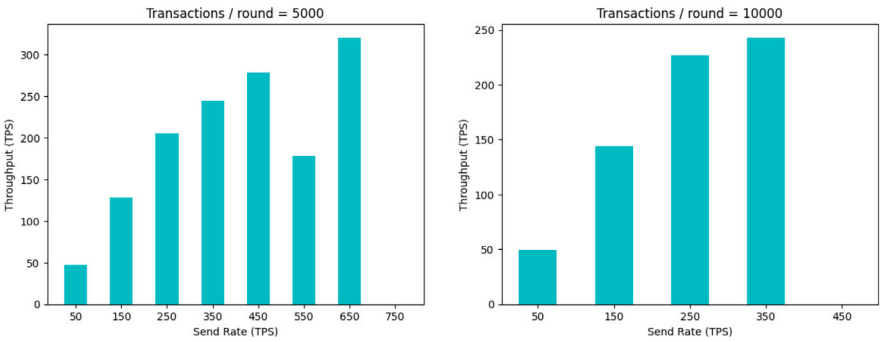

Fig. 6. Throughput of the distributor smart contract for 5k and 10k writing transactions/round.

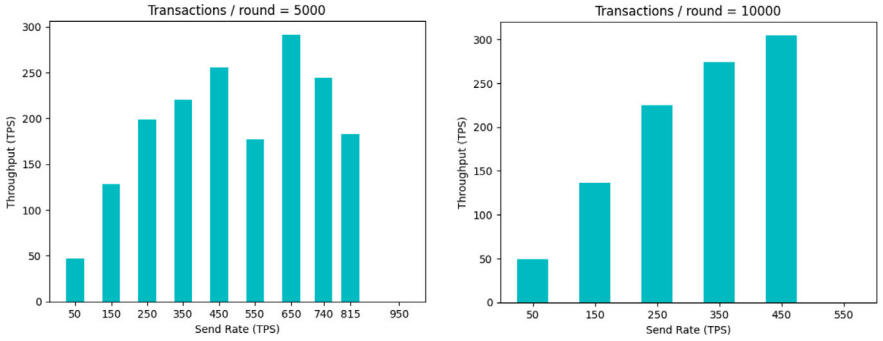

Fig. 7. Throughput of the distributor smart contract for 5k and 10k reading transactions/round.

robust and trustworthy framework for facilitating collaboration between humans and robots in the field of precision agriculture [54]. Precision agriculture involves the utilization of advanced technologies, including *artificial intelligence* (AI), to enhance various agricultural practices, such as optimizing grape harvesting and branch pruning operations, thereby contributing to the goals of B-CONFIDENT. The utilization of AI algorithms within precision agriculture can further enhance decision-making processes, enabling the system to analyze and interpret data collected from blockchain-enabled sensors and devices, ultimately facilitating data-driven insights and informed actions.

CRediT authorship contribution statement

Simone Agostinelli: Writing – original draft, Validation, Project administration, Methodology, Investigation. **Ala Arman:** Writing – original draft, Methodology, Investigation, Formal analysis. **Francesca De Luzi:** Writing – original draft, Formal analysis, Resources, Data curation. **Flavia Monti:** Writing – original draft, Formal analysis, Resources, Data curation. **Michele Manglaviti:** Software. **Massimo Mecella:** Methodology, Investigation, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Acknowledgments

The work of S. Agostinelli has been supported by the PNRR MUR project, Italy PE0000013-FAIR. The work of A. Arman is funded by MICS, Italy (PE00000004) Extended Partnership (CUP B53C22004130001) funded by the EU - NextGenerationEU PNRR MUR. The work of F. De Luzi, M. Manglaviti, and M. Mecella has been supported by Associazione CYBER 4.0, instrument "bando 2/2021 per la selezione e il cofinanziamento di progetti di innovazione, ricerca industriale e sviluppo sperimentale", project BINTRAWINE - Blockchain, Tracking and Tracing Solutions for Wine (CUP: C42C21002030008). The work of F. Monti is supported by the MISE agreement on Agile&Secure Digital Twins, Italy (A&S-DT).

Data availability

The authors do not have permission to share data. The software is available via the link provided in the paper.

References

[1] M. Cooper, L. Ellram, Characteristics of supply chain management and the implications for purchasing and logistics strategy, Int. J. Logist. Manage. 4 (2) (1993) 13–24.

[2] J. Hu, Y. Liu, T. Yuen, M. Lim, J. Hu, Do green practices really attract customers? the sharing economy from the sustainable supply chain management perspective, Resour. Conserv. Recycl. 149 (2019) 177–187.

[3] W. Czakon, P. Klimas, M. Mariani, Behavioral antecedents of coopetition: A synthesis and measurement scale, Long Range Plan. 53 (1) (2020) 101875, http://dx.doi.org/10.1016/j.lrp.2019.03.001, coopetition Strategies.

[4] J. Monticelli, J. Verschoore, I. Garrido, The emergence of coopetition in highly regulated industries: A study on the brazilian private healthcare market, Ind. Mark. Manag. 108 (2023) 35–46.

[5] M.-L. Marsal-Llacuna, Future living framework: Is blockchain the next enabling network? Technol. Forecast. Soc. Change 128 (2018) 226–234.

[6] S.E. Chang, Y. Chen, When blockchain meets supply chain: A systematic literature review on current development and potential applications, IEEE Access 8 (2020) 62478–62494, http://dx.doi.org/10.1109/ACCESS.2020.2983601.

[7] H. Jalali, A. Ansaripoor, V. Ramani, P. De Giovanni, Closed-loop supply chain models with coopetition options, Int. J. Prod. Res. 60 (10) (2022) 3078–3106.

[8] S. Wijethilaka, A.K. Yadav, A. Braeken, M. Liyanage, Blockchain-based secure authentication and authorization framework for robust 5 g network slicing, IEEE Trans. Netw. Serv. Manag. 21 (4) (2024) 3988–4005.

[9] J. Chan, R. Song, K. Schelhowe, Decentralised technology adoption with a coopetition approach, in: 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE, IEEE, 2023, pp. 1–6.

[10] M. Queiroz, S. Wamba, Blockchain adoption challenges in supply chain: An empirical investigation of the main drivers in india and the usa, Int. J. Inf. Manage. 46 (2019) 70–82.

[11] J. Polge, J. Robert, Y. Le Traon, Permissioned blockchain frameworks in the industry: A comparison, ICT Express 7 (2) (2021) 229–233, http://dx.doi.org/10.1016/j.icte.2020.09.002.

[12] L. Cristaldi, P. Esmaili, G. Gruosso, A. La Bella, M. Mecella, R. Scattolini, A. Arman, G.A. Susto, L. Tanca, The mics project: A data science pipeline for industry 4.0 applications, in: 2023 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering, MetroXRAINE, IEEE, 2023, pp. 427–431.

[13] M. Nour, J. Chaves-Avila, A. Sánchez-Miralles, Review of blockchain potential applications in the electricity sector and challenges for large scale adoption, IEEE Access 10 (2022) 47384–47418, http://dx.doi.org/10.1109/ACCESS.2022.3171227.

[14] P. Ray, Appendix C: Blockchain Technology, John Wiley & Sons, Ltd, 2018, pp. 337–346, http://dx.doi.org/10.1002/9781119331797.app3, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119331797.app3.

[15] C. Bai, J. Sarkis, A supply chain transparency and sustainability technology appraisal model for blockchain technology, Int. J. Prod. Res. 58 (7) (2020) 2142–2162, http://dx.doi.org/10.1080/00207543.2019.1708989, arXiv:https://doi.org/10.1080/00207543.2019.1708989.

[16] V. Venkatesh, K. Kang, B. Wang, R. Zhong, A. Zhang, System architecture for blockchain based transparency of supply chain social sustainability, Robot. Comput.-Integr. Manuf. 63 (2020) 101896.

[17] J. Sunny, N. Undralla, V. Madhusudanan Pillai, Supply chain transparency through blockchain-based traceability: An overview with demonstration, Comput. Ind. Eng. 150 (2020) 106895, http://dx.doi.org/10.1016/j.cie.2020.106895.

[18] G. Hastig, M. Sodhi, Blockchain for supply chain traceability: Business requirements and critical success factors, Prod. Oper. Manage. 29 (4) (2020) 935–954, http://dx.doi.org/10.1111/poms.13147, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/poms.13147.

[19] B.B. Sezer, S. Topal, U. Nuriyev, Tppsupply: A traceable and privacy-preserving blockchain system architecture for the supply chain, J. Inf. Secur. Appl. 66 (2022) 103116.

[20] X. Yang, M. Li, H. Yu, M. Wang, D. Xu, C. Sun, A trusted blockchain-based traceability system for fruit and vegetable agricultural products, IEEE Access 9 (2021) 36282–36293, http://dx.doi.org/10.1109/ACCESS.2021.3062845.

[21] Y. Wang, A. Kogan, Designing confidentiality-preserving blockchain-based transaction processing systems, Int. J. Account. Inf. Syst. 30 (2018) (2017) 1–18, http://dx.doi.org/10.1016/j.accinf.2018.06.001, Research Symposium on Information Integrity & Information Systems Assurance.

[22] H.D. Bandara, S. Chen, M. Staples, Y. Sai, Modeling multi-layer access control policies of a hyperledger-fabric-based agriculture supply chain, in: 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA, 2021, pp. 355–364, http://dx.doi.org/10.1109/TPSISA52974.2021.00039.

[23] K. Katsaliaki, S. Kumar, V. Loulos, Supply chain coopetition: A review of structures, mechanisms and dynamics, Int. J. Prod. Econ. 267 (2024) 109057, http://dx.doi.org/10.1016/j.ijpe.2023.109057, URL https://www.sciencedirect.com/science/article/pii/S092552732300289X.

[24] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, J. Mendling, Untrusted business process monitoring and execution using blockchain, in: Business Process Management: 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings 14, Springer, 2016, pp. 329–347.

[25] J. Morgan, Quorum: A permissioned implementation of ethereum supporting data privacy, 2016.

[26] Vitalik Buterin, Ethereum whitepaper, 2014, https://ethereum.org/en/whitepaper, (Accessed 3 December 2023).

[27] M. Correia, From byzantine consensus to blockchain consensus, Essent. Blockchain Technol. 41 (2019) 2019.

[28] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized Bus. Rev. (2008).

[29] N. Szabo, Formalizing and securing relationships on public networks, First Monday (1997).

[30] J. Mendling, I. Weber, W. Aalst, J. Brocke, C. Cabanillas, F. Daniel, S. Debois, C. Di Ciccio, M. Dumas, S. Dustdar, et al., Blockchains for business process management-challenges and opportunities, ACM Trans. Manag. Inf. Syst. (2018).

[31] B. Sriman, et al., Blockchain technology: Consensus protocol proof of work and proof of stake, Adv. Intell. Syst. Comput. (2021).

[32] V. Buterin, et al., A next-generation smart contract and decentralized application platform, White Pap. 3 (37) (2014).

[33] T. Alladi, V. Chamola, R. Parizi, K. Choo, Blockchain applications for industry 4.0 and industrial iot: A review, Ieee Access 7 (2019) 176935–176951.

[34] M. Brookbanks, G. Parry, The impact of a blockchain platform on trust in established relationships: a case study of wine supply chains, Supply Chain Manag.: Int. J. (2022).

[35] P. Danese, R. Mocellin, P. Romano, Designing blockchain systems to prevent counterfeiting in wine supply chains: a multiple-case study, Int. J. Oper. Prod. Manage. (2021).

[36] G. Dicuonzo, F. Donofrio, S. Ranaldo, M. Turco, The impact of blockchain on smes' sustainability. the case of an apulian wine company, Small Bus. (2021).

[37] F. Matos, T. Alcobia, A. Matos, Blockchain technology and traceability in the wine supply chain industry, in: ECIAIR, 2021.

[38] T. Dasaklis, T. Voutsinas, G. Tsoulfas, F. Casino, A systematic literature review of blockchain-enabled supply chain traceability implementations, Sustainability 14 (4) (2022) 2439.

[39] C. Chou, N. Hwang, C. Li, T. Wang, Y. Wang, Implementing a multichain framework using hyperledger for supply chain transparency in a dynamic partnership: A feasibility study, Comput. Ind. Eng. 175 (2023) 108906.

[40] P. Azevedo, J. Gomes, M. Romão, Supply chain traceability using blockchain, Oper. Manag. Res. 16 (3) (2023) 1359–1381.

[41] R. Brandín, S. Abrishami, Iot-bim and blockchain integration for enhanced data traceability in offsite manufacturing, Autom. Constr. 159 (2024) 105266.

[42] H. Zhao, K. Hu, Z. Yuan, S. Yao, L. Feng, Bctmssf: a blockchain consensus-based traceability method for supply chain in smart factory, J. Intell. Manuf. (2024) 1–17.

[43] E. Marangone, C.D. Ciccio, D. Friolo, E.N. Nemmi, D. Venturi, I. Weber, MARTSIA: enabling data confidentiality for blockchain-based process execution, in: 27th International Conference on Enterprise Design, Operations, and Computing, EDOC 2023, Vol. 14367, Springer, 2023, pp. 58–76, http://dx.doi.org/10.1007/978-3-031-46587-1_4.

[44] E. Marangone, C. Di Ciccio, I. Weber, Fine-grained data access control for collaborative process execution on blockchain, in: BPM (Blockchain Forum), 2022.

[45] V. Mullet, P. Sondi, E. Ramat, A blockchain-based confidentiality-preserving approach to traceability in industry 4.0, Int. J. Adv. Manuf. Technol. 124 (3–4) (2023) 1297–1320.

[46] X. Tao, Y. Liu, P. Wong, K. Chen, M. Das, J. Cheng, Confidentiality-minded framework for blockchain-based bim design collaboration, Autom. Constr. 136 (2022) 104172.

[47] Ag and Food Sectors and the Economy, United States Department of Agriculture (USDA) Economic Research Service, https://www.ers.usda.gov/data-products/ag-and-food-statistics-charting-the-essentials/ag-and-food-sectors-and-the-economy/.

[48] A. Corallo, M.E. Latino, M. Menegoli, F. Striani, The awareness assessment of the italian agri-food industry regarding food traceability systems, Trends Food Sci. Technol. 101 (2020) 28–37, http://dx.doi.org/10.1016/j.tifs.2020.04.022.

[49] I. Chrysakis, P. Papadakos, T. Patkos, G. Flouris, G. Samaritakis, D. Angelakis, N. Tsampanaki, N. Basina, P. Baritakis, A. Pratikaki, I. Loulakakis, B. Lyrarakis, The mib system: An interactive storytelling experience in the wine industry, in: 2022 13th International Conference on Information, Intelligence, Systems & Applications, IISA, 2022, pp. 1–8, http://dx.doi.org/10.1109/IISA56318.2022.9904362.

[50] A. Dix, Statistics for HCI: Making Sense of Quantitative Data, Morgan & Claypool Publishers, 2020.

[51] J. Sauro, J. Lewis, Quantifying the User Experience: Practical Statistics for User Research, Morgan Kaufmann, 2016.

[52] I. Chakravarti, R. Laha, J. Roy, Handbook of Methods of Applied Statistics, in: Wiley Series in Prob. and Math. Statistics (USA), 1967.

[53] W. Choi, J. Hong, Performance evaluation of ethereum private and testnet networks using hyperledger caliper, in: APNOMS, 2021.

[54] S. Kaszuba, et al., A preliminary study on virtual reality tools in human–robot interaction, in: Augmented Reality, Virtual Reality, and Computer Graphics, 2021.