Addressing Trust Issues in Supply-Chain Management Systems through Blockchain Software Patterns

Eddy Kiomba Kambilo[0009−0008−5623−3125], Irina Rychkova[0000−0002−1100−0116], Nicolas Herbaut[0000−0003−1540−2099], and Carine Souveyet[0000−0001−5677−9808]

Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, 75013 Paris, France
{firstname.lastname}@univ-paris1.fr

Abstract. Blockchain technology is a decentralized and distributed ledger that allows for secure, transparent, and immutable tracking of transactions. However, it is not a one-size-fits-all solution for addressing trust issues in the supply chain. In software engineering, design patterns provide a blueprint that developers can follow to solve a specific problem in a structured and efficient manner. In this paper, we identify and discuss the reusable blockchain software patterns that can be applied to design trustworthy solutions in supply chain management (SCM). Based on the literature analysis, we define a comprehensive taxonomy of SCM-specific trust issues. Then we apply requirement engineering technique to translate these issues into trust requirements and demonstrate how these requirements can be met by the specific blockchain software patterns.

Keywords: Trust · Blockchain · Software patterns · Supply Chain Management

1 Introduction

The supply chain is one of the most important economic systems[13] because it enables the production and delivery of goods and services to customers. In [15], the authors define supply chain as *"the network of organizations involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services delivered to the ultimate consumer"*. Trust is a vital aspect with far-reaching consequences across various domains. In supply chains, trust plays a key role in shaping relationships between stakeholders: timely identification and elimination of trust issues is crucial for successful collaborations. Gambetta[14] defines trust as *"the expectation that another person (or institution) will perform actions that are beneficial or at least not detrimental, to us regardless of our capacity to monitor those actions"*. Following this definition, a *trust issue* can be defined as a lack of trustor's belief that another party (trustee), for one reason or another, will actually meet these expectations.

In modern society, where interpersonal or inter-organizational relations are often mediated by technology, trust becomes multidimensional: Mayer [27] defines

trust between social entities (individuals or organizations), McKnight[10] specifies trust between humans and technology (Artificial Intelligence, Business intelligence), Andrew[31] discusses trust between humans depending on technology, Pietrzak et. al. [32] addresses digital trust as a determinant of interpersonal and inter-organizational relationships in the digital world. As a result, trust issues related to digital security, privacy of data, process transparency and performance gain a lot of attention.

Today, blockchain is considered a de facto technology to address trust issues in the supply chain. Blockchain is a distributed ledger system supported by a network of peers, each of whom maintains a copy of the ledger[28]. Blockchain is particularly attractive in supply chains due to its hacker-proof architecture and cryptographic algorithms(aspects), such as consensus algorithms that allow to verify and validate transactions on the network. In the context of SCM, this means that all parties involved in the supply chain can trust that the data recorded on the blockchain is accurate and has been agreed upon by the network. Additionally, blockchain can control access to information through smart contracts by defining specific conditions that must be met in order for the information to be accessed. These self-executing contracts, which are tamper resistant and traceable, can monitor the activity of each participant based on hash and signatures of each transaction[34].

Implementing blockchain-based solutions in SCM, organizations aim to address trust issues. However the success of this endeavor is contingent on the architectural model and implementation of blockchain technology.

Software patterns provide a blueprint that software engineers can follow to solve a specific problem in a structured and efficient manner. Bushman[4] defines software patterns as *"a function-form relationship that occurs in a context where the function is described in terms of unresolved trade-offs or forces in the problem domain. The form is a structure described in the solution domain that achieves a good and acceptable equilibrium among those forces."* Software patterns focus on capturing and systematizing successful experiences and techniques used in software development.

Blockchain software patterns are discussed in the literature [35][39]. To the best of our knowledge, there is a lack of research exploring blockchain software patterns focusing on trust. In this paper, we investigate how trust issues expressed in the supply chain domain (problem domain) can be efficiently addressed by blockchain technology (solution domain) using specific blockchain software patterns.

In this work, we develop the following contributions: 1) We construct a taxonomy of trust issues in Supply Chain Management(SCM) based on the analysis of 18 research publications in the domain. 2) Following the guidelines of requirements engineering, we propose a technique for translating trust issues into trust requirements. 3) We define the mapping between the formulated trust requirements and the blockchain software patterns. This mapping can guide decision-making in the design of trustworthy solutions in SCM.

The paper is structured as follows: In Section 2, we provide the background on blockchain technology, software patterns, supply chain, and discusses the concept of trust. In Section 3, we define the taxonomy of trust issues in SCM based on related literature, than we translate these issues into trust requirements. In section

4, we discuss how the trust requirements from the previous section can be met by the specific blockchain software patterns. We illustrate our findings on the example of Section 5. This example also serves as a preliminary validation of our findings. In Section 6 we present our conclusions.

2 Background

2.1 Trust

In social sciences, trust is defined as *"the willingness of one party (trustor) to be vulnerable to the actions of another party (trustee), based on the expectation that the other party will perform the expected action"*[27]. *Social trust* reflects (subjective) trustor's beliefs that the trustee has suitable at- tributes for performing as expected in a specific situation. These attributes include ability, benevolence and integrity [14]. Zheng [40] stated that social trust is a product of experiences and perceived trustworthiness.

Advances in technology, such as Artificial Intelligence (AI) and robotics have led to the need for organizations to establish processes to regulate trust in technology [9]. *Trust in technology* can be defined as trustor's confidence in technology (trustee) to accomplish the task at hand. In [10], the authors present three essential elements that can help build trust in technology: reliability, functionality, and helpfulness. *Digital trust* emerges in interpersonal or inter-organizational relations where technology plays a role of mediator. Jeffrey(2020) defines digital trust *"as the confidence users have in the ability of people, technology, and processes to create a secure digital world"*[1]. Trust in technology is a precursor to digital trust, as people must trust technology before using it between them.

2.2 Blockchain in Supply Chain Management

Supply chain management (SCM) aims to ensure that goods and services are delivered to consumers promptly, cost effectively, and efficiently[15]. In SCM, trust plays an important role. Tradelens[19] is one example of blockchain in SCM. It provides transparency, efficiency, and accountability in global trade by digitizing and streamlining the flow of information and documents among supply chain participants. Another example of practical applications is traceability that can be provided through a blockchain solution [21].

Blockchain is a decentralized, distributed ledger technology widely recognized as a critical enabler for the secure, transparent, and immutable tracking of transactions [36]. Blockchain has several intrinsic features that make it relevant for supply chain management. It can create a *decentralized* and *tamper-proof* ledger of all transactions that occur throughout the supply chain. The ledger could track the movement of goods from their origin to their final destination, providing complete *transparency* and *traceability*. It can also help increase supply chain data's *integrity* by enabling *monitoring* and *auditing* of all transactions. Smart contracts can be

[1] https://www.techtarget.com/whatis/definition/digital-trust

used to *automate* specific processes within the supply chain. This can improve efficiency and reduce the risk of human errors. Finally, blockchain can be used for *the real-time identification* of goods, particularly for perishable goods with a limited shelf life.

2.3 Software patterns

Pattern-based design is widely adopted by the software engineering community since the mid-1990s. The resulting software patterns describe recurring designs used in software development [4]. A software pattern is considered as *"a function-form relation that occurs in a context, where the function is described in problem domain terms as a group of unresolved trade-offs or forces, and the form is a structure described in solution domain terms that achieve a good and acceptable equilibrium among those forces."*[4]

According to [39], software patterns play a vital role in addressing trust issues. Blockchain software pattern is a repeatable design solution to a recurring problem in blockchain development [35]. In SCM, blockchain software patterns can provide a systematic way of tackling trust-related concerns, such as ensuring data authenticity and integrity, promoting transparency and accountability, and maintaining the privacy and security of the system.

In [35], authors identify a set of 120 unique patterns. 104 of them have been classified as design patterns, 3 of them as architectural patterns, and 14 as idioms. These blockchain software patterns come from a range of fields, including agriculture and industries and address a number of generic issues. In this work, we review the software patterns in [35] and identify twelve patterns that can be used to address the specific trust issues in SCM.

3 From Trust Issues to Trust Requirements in SCM

A trust issue refers to a challenge, problem, or disagreement that affects the level of trust between individuals or parties[22]. They can be grounded on explicit evidence (frauds, contract violations, bad user experience) or on implicit beliefs. They are subjective and hard to grasp. In order to be explicitly analyzed and addressed by the software solutions, trust issues need to be translated into requirements. A requirement is a statement which translates or expresses a need and its associated constraints and conditions[1].

In this section, we define a taxonomy of trust issues based on our analysis of related literature and translate these issues into trust requirements.

3.1 Taxonomy of Trust Issues in SCM

The work in [30] presents a methodology for building a taxonomy and discusses problems associated with taxonomy development. To establish our taxonomy, we define the following research protocol:

1 Identification: We conducted the search for primary research publications in the two major databases: Scopus and Google Scholar. We used the following key words: supply chain, trust, issues, requirements(38 papers for Scopus and 52 google scholar).
2 Selection: We selected the articles on the literature that met the following criteria :
C1: Evoke the issues related to social, digital trust or trust in technology
C2: Propose a solution that addresses trust issues or aims to improve trust in SCM. We identified (non-systematically) 18 research studies published between 2018 and 2022 (we filter by date to ensure the research is consistent and avoid irrelevant or outdated papers.) by screening abstracts and full texts.
3 Extraction: We extracted two types of text evidences
(a) evidence evoking (explicitly and implicitly) trust issues and
(b) evidence on the proposed technological solutions, indicating technological, architectural, design choices.
4 Synthesis: The extracted data was revised and discussed by several researchers (authors of this paper) to reduce the interpretation bias. Eventually the 21 extracted trust issues were grouped into 7 categories to form a taxonomy (Fig 1).

We applied the protocol to identify and formulate trust issues in all selected sources. Here is an example: In [37], the authors highlight the difficulties faced in Supply Chain Management in verifying the authenticity of goods and conducting investigations into illegal activities. Three main issues are identified as "insufficient auditing", "opacity - lack of transparency" and "lack of oversight". Another study by FranCasino[6] sheds light on the problem of a single point of failure in traditional databases, which can compromise privacy and tamper-proofing. We identify issues of "storage" and "privacy," etc.

Figure 1 groups trust issues extracted in the literature into seven categories, *"Traceability"* focuses on the challenge of tracking and monitoring products in real time along the supply chain. *"Cost Control"* addresses the stakeholders' concern about reducing costs associated with blockchain transactions(cost limit and cost reduction). It is essential to minimize the cost of these transactions to ensure their practicality. *"Lack of Auditability"* highlights the difficulties in auditing blockchain transactions, and stakeholders need to be able to audit them at any time. *"Security"* emphasizes the importance of keeping data and transactions confidential, secure, and tamper-proof. *"Data Governance"* concerns the users' control over data shared with other institutions and the need to anticipate scalability to avoid additional fees. *"Lack of Accountability"* highlights the responsibility of stakeholders to be accountable for their actions, as it is essential for everyone to take responsibility for their decisions and actions. The final category, *"Acceptance"* that concerns the user acceptance.

3.2 Translating Trust Issues Into Trust Requirements.

Trust issues are subjective and sometimes stem from stakeholders' intentions. Defining requirements starts with understanding the stakeholders' intentions, needs,

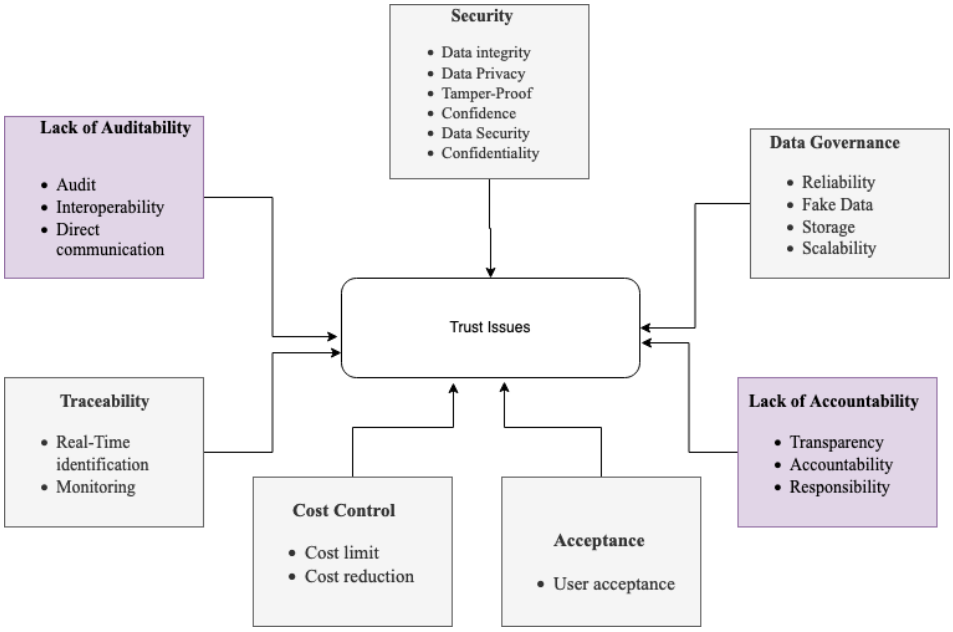

Fig. 1. Taxonomy of Trust Issues

goals, or objectives, as outlined in ISO[1]. In requirements engineering, a requirement is defined as a statement which identifies an operational, functional or design features or constraint of the product or process, which is unambiguous, testable or measurable, and necessary for the product or process to be accepted by consumers or internal quality assurance guidelines[1]. A set of explicit, clearly stated requirements facilitates communication between stakeholders: it justifies technological and design decisions and provides a basis for solution validation. When expressed in natural language, the statement of requirements should include a subject (e.g., system, software, etc.), an active verb and other elements necessary to specify the information content of the requirement.

The guidelines for writing requirements are specified by ISO/IEC standard[1]. Transforming trust issues into trust requirements involves thoroughly analyzing the needs and expectations of all stakeholders and examining current systems and practices. The subjective needs of stakeholders are then transformed into objective needs or objectives.

Requirements Engineering(RE) as a mediator between the acquirer and supplier domains, establishing and maintaining the requirements for the desired system, software, or service[33]. RE covers the discovery, elicitation, development, analysis, determination of verification methods, validation, communication, documentation, and management of requirements[1]. It is a crucial part of the software development

process and involves stakeholder collaboration to guarantee that the end product fulfills stakeholders' needs and adheres to project constraints.

In this paper, we adapt the RE process and follow the ISO/IEC standard [1] for transforming trust issues into trust requirements. Our process consists of the following steps:

(1) Elicitation of user trust issues: In this step, the evidences of trust issues has to be gathered. Various techniques defined in the fields of requirements engineering and knowledge management can be used to collect the empirical data, including interviews, case studies, workshops, action research, etc. The outcome is a collection of trust issues expressed by end users or stakeholders. In this work, the data about trust issues has been collected through the literature review.

For instance, in [17], an evidence of lack of accountability (I1) is expressed as follows: "it is important to listen to the interactions and responsibility between the suppliers and the OEM to maintain the transparency between the different vendors". In [29], the same issue is expressed as follows: "The use of accountability and incentive structures to punish and encourage dishonest or trustworthy individuals was a strategy to increase trust and confidence in the data".

(2) Analysis: The purpose of this step is to analyze each expressed trust issue in order to identify a subject of trust (actor, system, process, technological component), an object of trust (e.g., data, activity, function, etc.) and an expected relationship that must be established between the former and the latter in order to mitigate the issue. For example: Lack of accountability (I1) issue addresses a business partner in the supply chain (the subject) and the transaction data (the object). The issue expresses a trustor's belief that, in case of dispute, the partner can avoid responsibility for his actions unless the formal proof of such actions is provided. To mitigate the issue, the transaction has to be non-reputable (the relationship). In the field, such analysis has to be conducted iteratively, confirming and validating the results with users.

(3) Specification of requirements: In this step, the requirements are documented based on the analysis from the previous step and following the recommendations from [1]. The outcome of this step is a formalized requirement specification. Example: For the Lack of accountability (I1) issue, we formalize the corresponding trust requirement as follows: "System must guarantee non-repudiation of data".

The taxonomy of 21 trust issues and their corresponding trust requirements defined following the process above is presented Table 1. This taxonomy provides a decision-making support for requirements engineers and designers and guides the design of the prospective trustworthy SCM solution. The proposed process for translating issues into requirements can potentially support designers in identifying new issues and requirements.

Table 1. Taxonomy & Mapping trust issues into trust Requirements

Taxonomy of Trust Issues	**Trust issues to Trust requirements**		
Issue	**References**	*Requirement Specification*	*Requirement Name*
Lack of accountability (I1)	[16][29][25][37][7][5][24][22][8]	System must guarantee non repudiation of data.	accountability
Lack of auditing(I2)	[18][16][24][8]	System must have log files to facilitate auditing	Audit
Lack of security (I3)	[18][38][3][37][7][17]	System must guarantee a no intrusion of external stakeholders	Security
Lack of interoperability(I4)	[18][38][26][37][7][12][6][2]	System must be flexible to insert other modules or SI.	Interoperability
Lack of transparency(I5)	[16][11][25][24][17][22][2]	System must save and share data in readable and transparent manner for everyone.	Transparency
High cost of transaction(I6)	[11][38][3][37][12][12][2]	System doesn't cost much by utilization	Cost control
Lack of reliability(I7)	[16][29]	System must be reliable	Reliability
Lack of confidence(I8)	[29]	System must guarantee data confidence	Confidence
Lack of confidentiality(I9)	[11][22]	System must guarantee procesoricesss confidentiality	Confidentiality
Lack of privacy(I10)	[11][3][17][6][22][2][8]	System must guarantee data privacy	Privacy
Lack of responsibility(I11)	[25]	System must save a trace of each transaction	Responsibility
Lack of traceability(I12)	[26] [3][2]	System must guarantee tracking of data	Traceability
Lack of direct communication(I13)	[16][29]	System must be decentralised with Peer-to-Peer communication	Decentralized
Lack of RT identification(I14)	[26][7][5][17] [8]	System must track data on real time	R-T identification
Lack of data integrity(I15)	[26][37][7][5][22][8]	System must be guaranteed data integrity	Integrity
Lack of User acceptance(I16)	[3][5]	System must be guaranteed a User satisfaction.	User Acceptance
Lack of monitoring(I17)	[12][24][6][2]	System must give the possibility to monitor all processes	Monitoring
Fake data(I18)	[3][6]	System must guarantee data origin	Fake Data
Lack of scalability(I19)	[37][24][22][8]	System must guarantee a High availability of data	Scalability
Lack of tamper-proof(I20)	[37][12][24][22]	System must guarantee data tamper-proof	Tamper-proof
Lack of good storage(I21)	[5][24][17][6]	System needs Good support for storage of data	Storage

4 Use of Blockchain Software Patterns for Meeting Trust Requirements in SCM.

Blockchain is often considered a de facto trust enabler. We argue that, while offering a number of key features, stock blockchain may not provide a complete solution for the specific trust requirements in a given context. In this section, we discuss the current limitations of stock blockchain solutions and propose the use of blockchain software patterns to efficiently address the specific trust requirements in SCM.

According to the literature, intrinsic features of blockchain(public) technology address a number of requirements in SCM including trustworthiness. However, this technology also has limitations that can overshadow the benefits and have a negative impact on trust. These limitations have to be taken into account when making *design decisions*. Challenges related to privacy, scalability, trust and interoperability are some examples relevant to the SCM domain. These challenges can be efficiently addressed using specific blockchain software patterns. Here are some examples.

Blockchain is not suitable for storing and managing large amounts of data. Keeping images and other large data sets in blockchain can be expensive due to the high cost of transaction fees. This undermines the performance and credibility of the solution and negatively impacts user's trust in this technology. To overcome this, off-chain data storage (patterns) such as the Interplanetary File System (IPFS) can be implemented in conjunction with blockchain. Using blockchain to store hashes of the data and off-chain storage to store extensive data can reduce transaction costs and add scalability to the system.

Along the same lines, if a blockchain solution is not designed with security in mind, it can be vulnerable to cyberattacks or other forms of malicious activity. This can result in the loss of funds or the compromise of sensitive information. Additionally, if the consensus mechanism used in the blockchain is not well adapted, it could lead to issues with trust in the network. For privacy, if a blockchain is not designed with privacy in mind, it can lead to the exposure of sensitive information because ledger is public for all stakeholders. The use of encryption-on chain(patterns) data is recommended to address these challenges.

Table 2 presents the mapping between the trust requirements in SCM, the key blockchain features discussed in Section 2.2 that are recurrently used to meet these requirements, and the blockchain software patterns from [35], which we identify to complement the features. We consider three cases:

CASE 1 blockchain features provide a complete solution for specific requirements. For example: transparency, as the transparent ledger offered by blockchain inherently meets this requirement without the need for additional patterns (indicated with a ✔ symbol in BC features column in the table).

CASE 2 : blockchain features are insufficient to meet a requirement and blockchain software patterns are proposed. For example, a public blockchain's (transparency) cannot ensure data privacy. In this case, a private blockchain or encryption on-chain data patterns can encrypt data during transit to maintain privacy on the

blockchain (indicated with a − symbol in BC software patterns column in the table).

Table 2. Mapping of Trust requirements on BC features & Patterns
✔: Satisfied Req., !: partially satisfied Req.,−: Unsatisfied Req.

Requirements	**Blockchain Features**	**Blockchain Software Patterns [35]**
Accountability	✔Accountability	✔ Identifier Registry
Audit	✔ Audit	−
Interoperability	−	✔ Contract Observer
Transparency	✔ Transparency	−
Cost control	!Automating	! Minimize On-Chain data, Flyweight, Off-chain data storage
Reliability	✔ Immutability	−
Privacy	−	✔ Encryption on-chain
Responsibility	−	✔ Identifier Registry
Traceability	✔ Traceability	−
Decentralized	✔ Decentralized	−
RT identification	✔ RT identification	−
Integrity	! Hash, Integrity	! Hash secret
Monitoring	✔ Monitoring	! Event Log, Publisher-Subscriber
Scalability	−	✔ State channel[50%], off-chain data storage[50%]
Tamper-proof	✔ Tamper-proof	! Embedded permission
Storage	−	✔ Off-chain data storage, Limit-Storage

CASE 3 : blockchain features offer only a partial solution and can be complemented by using software patterns to meet a requirement. For example, the automation of information systems in blockchain can lead to increased costs if the data transit is extensive in terms of storage. To address this issue, minimize On-chain data or using Flyweight patterns can limit the data size in each transaction and reduce costs (indicated with a ! symbol in the table).

Implementing blockchain solutions by using software patterns can help to improve scalability, good storage, privacy and trust in the supply chain management

environment. In Table 2, we present the mapping of 16 out of 21 trust requirements in SCM . According to our analysis, Security, Confidence, Confidentiality, Fake data, and User acceptance requirements are not fully met by the current blockchain solutions (features and/or patterns). These complex problems require more research and development and need to be addressed by the blockchain community in the future.

5 Illustrative Example and Discussion

To illustrate our proposed mapping of trust issues / requirements into specific blockchain software patterns and to provide the initial validation of this mapping, we consider an example from [7]. This paper discusses the use of blockchain for supporting traceability in a food supply chain for food safety risk management and compliance. It describes in details the design and implementation and validates the approach.

5.1 Running the process on the illustrative example

Following the process defined in Section 3.2 , we identified the following trust issues from this case [7] and mapped them on our taxonomy in Table 1:

Lack of Interoperability (I4): According to the case, "The aim is to develop an interoperable, autonomous systems", where heterogeneous stakeholders can collaborate. *High cost of transactions(I6):* Stakeholders are preoccupied by the transaction costs: "Ipfs help to store large amount of data to reduce cost transaction". *Lack of Scalability(I19):*"We must use decentralized storage such as IPFS to guarantee the integrity of the information", stakeholders need systems with scalability of data. *Lack of Traceability(I12):* "Traceability-related information is not shared between participants, since they have their own traceability mechanisms and inevitably store their unique traceability records" *Lack of Data integrity and privacy (I15, I10):* "It is essential to guarantee the privacy of the transactions and the involved actors by using SC".

We map the identified trust issues on trust requirements. For exemple: "Stakeholders need to be reassured of the source of provenance and the authenticity of the products." correspond to Traceability requirement in Table 1. We analyzed the architecture and patterns proposed by the case authors and compared them to our recommendation based on the mapping in Table 2. Table 3 shows the comparison results.

5.2 Results

In this example, we were able to identify the trust issues from the case text and to map them on our taxonomy. Though the patterns indicated by the case authors are not expressed explicitly, we were able to match them with the patterns provided by [35]. The authors in the case use IPFS to store vast amounts of data, which generates a hash secret on a smart contract address to reduce transaction

oricess

Table 3. Blockchain features and Patterns resolving trust issues
✔: Satisfied Req., !: partially satisfied Req.,−: Unsatisfied Req.

Requirements (extracted from the case)	BC intrinsic features	Patterns (extracted from the case)	BC software patterns (our proposal)
Interoperability	−	−	Contract Observer
Cost control	!	Off-chain data storage	Limit Storage
Privacy	−	Encryption On-chain	Encryption On-chain
Traceability	✔	−	−
Integrity	!	Hash Secret	Hash Secret
Storage	−	Off-chain data storage	Off-chain data storage, Limit storage

costs and enable `Off-chain data storage`. The authors are also concerned with ensuring the privacy and traceability of data through encryption and the use of hash (`Encryption On-chain` & `Hash secret`).

The patterns used in the case correspond to our recommended blockchain software patterns for the following four requirements: Privacy, Integrity, Storage and Cost control. For the storage and Cost c requirement, we propose the `Limit storage` pattern to set a gas limit for transactions in addition to the `Off-chain data Storage`, already identified by the case authors.

The Traceability requirement does not require a specific pattern as it is directly provided by blockchain. This corresponds to our mapping in Table 2. The Interoperability requirement is not addressed in the paper. We propose the following blockchain software patterns to meet these requirements: the `Contract observer` pattern to guarantee interoperability and confirm that data written to the blockchain comes from a trustworthy source, and the `Identifier registry` pattern to track transactions and hold stakeholders accountable in the event of any problems.

In this example, we used a mapping proposed in the previous section to extend the solution proposed by the authors with two specific blockchain software patterns. This proposal completes the solution by addressing more trust requirements.

5.3 Discussion

Software patterns provide developers with technical best practices. However the trust implications of the patterns are not explicit. Our literature analysis shows that trust issues are not explicitly addressed in the design of the SCM solutions. Trustworthiness is often taken for granted by the mere use of a blockchain and cannot by validated.

The blockchain often meets trust requirements through a goal-oriented approach, which many researchers have explored. For example, authors in [23] focused on studying and identifying trust requirements in blockchain systems and created a trust engineering taxonomy to meet blockchain systems' trust requirements and goals. However, they did not provide evidence of how their taxonomy can be used or how to meet trust requirements using goals. In [20], a goal-oriented approach for business process reengineering is discussed. Here trustworthiness concerns are explicitly represented as (soft) goals and mapped to the relevant trust-enhancing features of blockchain, supporting business process reengineering. These works use a goal-oriented approach to focus on requirements and how specific blockchain features meet trust requirements.

The uniqueness of our approach resides in combining specific blockchain patterns from [35] and intrinsic blockchain features to address trust issues and to create trustworthy solutions in the field of SCM.

We make a first attempt to create a taxonomy of *trust issues* and *trust requirements* and define their design implications for blockchain solutions. This taxonomy will guide organizations to create trustworthy solutions in SCM.

Despite the promising results obtained from our study, it is important to note that there are several limitations and opportunities for improvement that should be addressed. These include:

- We identified trust issues using a sample of 18 research articles on SCM. We plan to conduct Systematic Literature Review to validate our findings and to extend our taxonomy.
- We limited our study to SCM, and more work is required to generalize this approach to other domains.
- Both trust issues and pattern were often not explicit in the literature, we had to rely on our expertise and interpretation to extract them. Our main effort is to promote standardization and the use of a common language (taxonomy of trust issues and requirements) to alleviate the interpretation bias in the future.

For the practical and effective use of of the proposed taxonomy and mapping, we plan to establish a recommendation system application where developers can select the issues encountered as input and have access to the blockchain software patterns they can leverage to ensure a design that inspires trust for collaboration.

6 Conclusion

While blockchain technology is often considered as the de facto trust enabler, some limitations persist. These limitations has to be systematically addressed by improved design practices. In this article we consider blockchain software design patterns to address the trust issues in SCM domain. First, we provided an overview and developed a taxonomy of trust issues based on literature in the supply chain. Following the recommendations from requirement engineering, we translated the trust issues into explicit trust requirements. We examined the existing solutions that address these trust requirements in the literature and identified their limitations: We argue that the use of intrinsic features of blockchain often provides only

a partial solution for trust issues in SCM. To complete this solution, we propose the use of blockchain software patterns.

We identified 12 patterns from the blockchain software pattern literature that can support the trust requirements in SCM and evaluated our proposal on one example from the literature. This preliminary evaluation shows the relevance of the trust issues taxonomy defined in this work. The proposed blockchain software patterns extend the solution from the case, demonstrating the potential interest and added value of our mapping.

This work aims at helping enterprises to better understand their trust-related requirements and to improve the design of their SCM systems.

References

1. Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering. ISO/IEC/IEEE 29148:2018(E) pp. 1–104 (2018). https://doi.org/10.1109/IEEESTD.2018.8559686
2. Baralla, G., Pinna, A., Tonelli, R., Marchesi, M., Ibba, S.: Ensuring transparency and traceability of food local products: A blockchain application to a smart tourism region. Concurrency and Computation: Practice and Experience **33**(1), e5857 (2021)
3. Biswas, D., Jalali, H., Ansaripoor, A.H., De Giovanni, P.: Traceability vs sustainability in supply chains: The implications of blockchain. European Journal of Operational Research **305**(1), 128–147 (2023)
4. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Software patterns (1996)
5. Caro, M.P., Ali, M.S., Vecchio, M., Giaffreda, R.: Blockchain-based traceability in agri-food supply chain management: A practical implementation. In: 2018 IoT Vertical and Topical Summit on Agriculture - Tuscany (IOT Tuscany). pp. 1–4 (2018). https://doi.org/10.1109/IOT-TUSCANY.2018.8373021
6. Casino, F., Kanakaris, V., Dasaklis, T.K., Moschuris, S., Rachaniotis, N.P.: Modeling food supply chain traceability based on blockchain technology. IFAC-PapersOnLine **52**(13), 2728–2733 (2019). https://doi.org/https://doi.org/10.1016/j.ifacol.2019.11.620, 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019
7. Casino, F., Kanakaris, V., Dasaklis, T.K., Moschuris, S., Stachtiaris, S., Pagoni, M., Rachaniotis, N.P.: Blockchain-based food supply chain traceability: a case study in the dairy sector. International Journal of Production Research **59**(19), 5758–5770 (2021). https://doi.org/10.1080/00207543.2020.1789238
8. Chang, S.E., Chen, Y.: When blockchain meets supply chain: A systematic literature review on current development and potential applications. IEEE Access **8**, 62478–62494 (2020). https://doi.org/10.1109/ACCESS.2020.2983601
9. Chopra, K., Wallace, W.A.: Trust in electronic environments. In: 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the. pp. 10–pp. IEEE (2003)
10. D. H. Mcknight, M. Carter, J.B.T., Clay, P.: Trust in a specific technology: An Investigation of its Components and Measures. ACM Transactions on Management Information Systems **2**(12), 1–25 (2011)
11. De Giovanni, P.: Blockchain and smart contracts in supply chain management: A game theoretic model. International Journal of Production Economics **228**, 107855 (2020)

12. Figorilli, S., Antonucci, F., Costa, C., Pallottino, F., Raso, L., Castiglione, M., Pinci, E., Del Vecchio, D., Colle, G., Proto, A.R., Sperandio, G., Menesatti, P.: A blockchain implementation prototype for the electronic open source traceability of wood along the whole supply chain. Sensors **18**(9) (2018)
13. Flores-González, L., Vargas Florez, J., Monteza-Valdivia, L., Cáceres-Cansaya, A., García-Salinas, J., Silva-Alarco, L.: Urban road network resilience assessment on freight logistics by simulating disruptive events. In: Production and Operations Management. pp. 427–450. Springer International Publishing, Cham (2022)
14. Gambetta, D., et al.: Can we trust trust. Trust: Making and breaking cooperative relations **13**(2000), 213–237 (2000)
15. Giannakis, M., Croom, S., Slack, N.: Supply chain paradigms. Understanding supply chains pp. 1–22 (2004)
16. Hameed, H., Zafar, N.A., Alkhammash, E.H., Hadjouni, M.: Blockchain-based formal model for food supply chain management system using vdm-sl. Sustainability **14**(21) (2022). https://doi.org/10.3390/su142114202
17. Hasan, H.R., Salah, K., Jayaraman, R., Ahmad, R.W., Yaqoob, I., Omar, M.: Blockchain-based solution for the traceability of spare parts in manufacturing. IEEE Access **8**, 100308–100322 (2020). https://doi.org/10.1109/ACCESS.2020.2998159
18. Imeri, A., Agoulmine, N., Feltus, C., Khadraoui, D.: Blockchain: Analysis of the new technological components as opportunity to solve the trust issues in supply chain management. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) Intelligent Computing. pp. 474–493. Springer International Publishing, Cham (2019)
19. Jensen, T., Hedman, J., Henningsson, S.: How tradelens delivers business value with blockchain technology. MIS Quarterly Executive **18**(4) (2019)
20. Johng, H., Kim, D., Park, G., Hong, J.E., Hill, T., Chung, L.: Enhancing business processes with trustworthiness using blockchain: a goal-oriented approach. In: Proceedings of the 35th Annual ACM Symposium on Applied Computing. pp. 61–68 (2020)
21. Kambilo, E.K., Zghal, H.B., Guegan, C.G., Stankovski, V., Kochovski, P., Vodislav, D.: A blockchain-based framework for drug traceability: Chaindrugtrac. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing. pp. 1900–1907 (2022)
22. Kamble, S.S., Gunasekaran, A., Sharma, R.: Modeling the blockchain enabled traceability in agriculture supply chain. International Journal of Information Management **52**, 101967 (2020). https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2019.05.023
23. Khalifa, D., Madjid, N.A., Svetinovic, D.: Trust requirements in blockchain systems: a preliminary study. In: 2019 Sixth International Conference on Software Defined Systems (SDS). pp. 310–313. IEEE (2019)
24. Kuhn, M., Funk, F., Zhang, G., Franke, J.: Blockchain-based application for the traceability of complex assembly structures. Journal of Manufacturing Systems **59**, 617–630 (2021). https://doi.org/https://doi.org/10.1016/j.jmsy.2021.04.013
25. Manning, L., Brewer, S., Craigon, P.J., Frey, J., Gutierrez, A., Jacobs, N., Kanza, S., Munday, S., Sacks, J., Pearson, S.: Artificial intelligence and ethics within the food sector: Developing a common language for technology adoption across the supply chain. Trends in Food Science & Technology **125**, 33–42 (2022). https://doi.org/https://doi.org/10.1016/j.tifs.2022.04.025
26. Masudin, I., Rahmatullah, B.B., Agung, M.A., Dewanti, I.A., Restuputri, D.P.: Traceability system in halal procurement: A bibliometric review. Logistics **6**(4), 67 (2022)
27. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. Academy of management review **20**(3), 709–734 (1995)

28. Mohanta, B.K., Panda, S.S., Jena, D.: An overview of smart contract and use cases in blockchain technology. In: 2018 9th International Conference on Computing, Communication, and Networking Technologies (ICCCNT). pp. 1–4 (2018)
29. N, J., Rampur, V., Gangodkar, D., M, A., C, B., N, A.K.: Improved block chain system for high secured iot integrated supply chain. Measurement: Sensors **25**, 100633 (2023). https://doi.org/https://doi.org/10.1016/j.measen.2022.100633
30. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. European Journal of Information Systems **22**, 336–359 (2013)
31. Patrick, A.S., Briggs, P., Marsh, S.: Designing systems that people will trust. Security and Usability **1**(1), 75–99 (2005)
32. Pietrzak, P., Takala, J.: Digital trust–asystematic literature review (2021)
33. Pohl, K.: Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated (2010)
34. Sahai, A., Pandey, R.: Smart contract definition for land registry in blockchain. In: 2020 IEEE 9th International conference on communication systems and network technologies (CSNT). pp. 230–235. IEEE (2020)
35. Six, N., Herbaut, N., Salinesi, C.: Blockchain software patterns for the design of decentralized applications: A systematic literature review. Blockchain: Research and Applications p. 100061 (2022)
36. Viriyasitavat, W., Hoonsopon, D.: Blockchain characteristics and consensus in modern business processes. Journal of Industrial Information Integration **13**, 32–39 (2019)
37. Wei, Y.: Blockchain-based data traceability platform architecture for supply chain management. In: 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,. pp. 77–85 (2020). https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00025
38. Xu, X., Choi, T.M.: Supply chain operations with online platforms under the cap-and-trade regulation: Impacts of using blockchain technology. Transportation Research Part E: Logistics and Transportation Review **155**, 102491 (2021)
39. Xu, X., Pautasso, C., Zhu, L., Lu, Q., Weber, I.: A pattern collection for blockchain-based applications. In: Proceedings of the 23rd European Conference on Pattern Languages of Programs. pp. 1–20 (2018)
40. Yan, Z., Holtmanns, S.: Trust modeling and management: From social trust to digital trust (2008)