Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 149, Issue 26, March/April 2021, Pages 26–45
https://doi.org/10.7250/csimq.2021-26.02

Using i* and UML for Blockchain Oriented Software Engineering: Strengths, Weaknesses, Lacks and Complementarity

Anne Sofie Vingerhoets[1], Samedi Heng[2], and Yves Wautelet[1⋆]

[1]KU Leuven, Oude Markt 13, 3000 Leuven, Belgium
[2]HEC Liège, Université de Liège, Rue Louvrex 14, 4000 Liège, Belgium

annesofie.vingerhoets@gmail.com, samedi.heng@uliege.be,
yves.wautelet@kuleuven.be

Abstract. New blockchain-based projects do appear every day. The technology has indeed been popularized by cryptocurrencies but is now gaining interest in various domains and new types of applications are evaluated constantly. Understanding the impact of blockchain adoption on the organization and the internals of blockchain-related behavior nevertheless remains a challenge for managers but also for IT professionals. This article studies how two existing organizational and software modeling languages can be fit to document a blockchain development project in Supply Chain Management (SCM) at its earliest stages. These two frameworks are i* on the one side and the Unified Modeling Language (UML) use case and sequence diagrams on the other side. The real life project used as a case study in this application is 'Farm-to-Fork' where a blockchain solution for the Supply Chain (SC) of farm animals is developed. The application of the frameworks is intended to identify their strengths and weaknesses. An extension of i* is proposed to deal with blockchain privacy issues as well as laws and norms. We finally point to the complementarity of i* and UML use case and sequence diagrams in a Blockchain-Oriented Software Engineering (BOSE) context. The i* framework indeed supports early requirements to understand the impact of the project on stakeholders while UML use case and sequence diagrams support the late requirements and the design by depicting the use of blockchain and some of its behavioral mechanisms.

Keywords: i* Framework, Blockchain, Blockchain-Oriented Software Engineering, Conceptual Modeling, Supply Chain Management, Distributed Ledger.

1 Introduction

Blockchain, through its decentralized nature, is seen nowadays as a very promising technology with applications that go far beyond the domain of cryptocurrencies. We have indeed seen

⋆ Corresponding author

Reference: A.S. Vingerhoets, S. Heng, and Y. Wautelet, "Using i* and UML for Blockchain Oriented Software Engineering: Strengths, Weaknesses, Lacks and Complementarity," Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 26, pp. 26–45, 2021. Available: https://doi.org/10.7250/csimq.2021-26.02

Additional information. Author ORCID iD: S. Heng – https://orcid.org/0000-0002-6037-0914 and Y. Wautelet – https://orcid.org/0000-0002-6560-9787. PII S225599222100149X. Received: 13 February 2021. Accepted: 15 April 2021. Available online: 30 April 2021.

applications in healthcare, finance, land registry, Supply Chain Management (SCM), etc. The field of blockchain is nevertheless still cumbersome to a lot of organization leaders that are in trouble to understand the added value of applications, the impact on stakeholders, the use cases, the infrastructure that has to be deployed, etc. Modeling the various stakeholders, their dependencies but also the blockchain's use cases and the dynamic between the actors then proves to be of value. There is nevertheless no commonly agreed upon modeling standard for Blockchain-Oriented Software Engineering (BOSE) adoption [1] so that existing modeling notations can be further investigated to overview the extent to which they fit the purpose. This article studies two existing modeling frameworks that could be used for such a purpose with a specific focus on the Supply Chain (SC) domain. These frameworks are also further refined to better fit the needs of blockchain-based modeling.

The i* framework [2] is a goal-oriented graphical requirement modeling notation [2]. It allows an early requirement engineering analysis in environments where social actors depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished [2]. Previous researches proved the relevance and utility of i* to model organizational requirements of a "multi agent system" [3] facilitating stakeholders' interactions by depicting their dependencies and hence providing a mean for coordination. i* was previously used to model several organizational settings [4] such as online stores [5], hospital beds management [6], [7], health care [2], SCs and more specifically outbound logistics [8], production support in the steel industry [9], and also for the development of higher education platforms like collaborative learning software [10] and MOOCs [11]. The i* framework is divided in two parts providing each a different level of abstraction: the Strategic Dependency (SD) and the Strategic Rationale (SR) model [2]. Figure 1 provides the core elements of the i* framework as well as their graphical representations. The SD model shows dependencies and the SR model depicts internal intents.

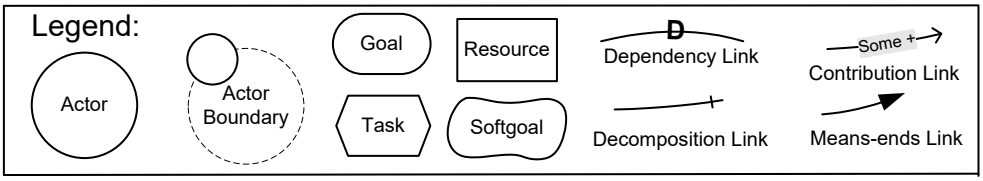


Figure 1. Relevant i* concepts and their graphical representations

The Unified Modeling Language (UML) [12] has for years been a reference in the field of object-oriented development. UML use case diagrams are well known to describe the cases in which a system can be used. We will thus apply it to depict the uses cases of a blockchain system. Figure 2 provides the core elements of the UML use case diagrams as well as their graphical representations. UML sequence diagrams are popular to describe the interactions between the actors and objects part of a centralized system. This is useful for blockchain in SC Requirements Engineering (RE) because the sequence diagram can depict a specific order of system operations, which corresponds very well to the nature of the SC flow. This similarity makes sequence diagrams a well-established candidate to model blockchain initiatives in the SC domain. Figure 3 provides the core elements of the UML sequence diagrams as well as their graphical representations.

The contribution of this article to the existing literature is threefold:

Firstly, this article applies a combination of two modeling techniques (i.e., the i* framework, as well as some UML models) to a relevant case study. The case study is based on the Farm-to-Fork initiative. The Farm-to-Fork solution is a blockchain prototype for the end-to-end food SC of farm animals. Interviews were conducted with 2 consultants including a validation of the produced representations.

Secondly, the ability of i* and some UML models to represent a blockchain-related problem in a SC context are evaluated against an extended set of criteria. The assessment criteria are based on existing literature and on the interviewee's expert opinions. This appraisal demonstrates

that both modeling techniques have their merits and deficiencies, and none of the two techniques outperforms the other for modeling blockchain in a SC context.

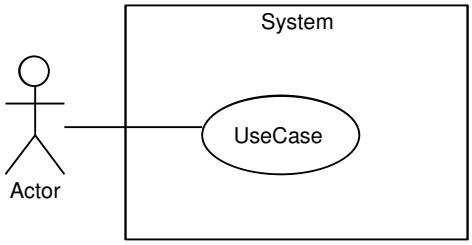


Figure 2. Relevant use case diagram concepts and their graphical representations

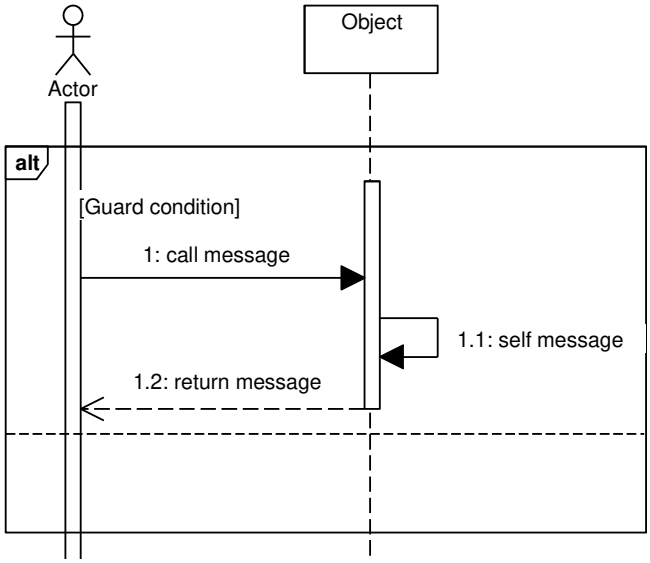


Figure 3. Relevant sequence diagram concepts and their graphical representations

Thirdly, a series of graphical extensions for the i* framework initially proposed by Ben Hamadi *et al*. [13] but not validated yet have been applied to make these models more tailored to describe blockchain in SCM. The enhancements include privacy concepts and the ability to model the compliance with laws or norms.

The article is structured as follows. Section 2 briefly discusses the benefits of blockchain and the related work. Section 3 explains the research paradigm, question and methodology. Section 4 discusses the case study on which we apply the frameworks, namely Farm-to-Fork. Section 5 describes the application of the i* framework on the case study while Section 6 describes the application of the UML use case diagram and sequence diagrams on the same case. Finally, Section 7 reports on the strengths and weaknesses of both frameworks and Section 8 concludes the article.

2 Background

2.1 Benefits of Blockchain

The main benefits of blockchain technology lie in its increased transparency and immutability [14], [15], [16]. Because the network is more accessible, transparency, and hence reliability, are increased. Trust is created in a trustless system [15]. Furthermore, blockchain

technology provides a highly secure method of dealing with transactions by using asymmetrical cryptography and hash functions [15]. While these are all primary benefits of blockchain, [16] describes the decentralized approach as probably the biggest advantage, because intermediaries are made completely redundant through a consensus mechanism in which data is verified by all participants, distributed and stored across different locations. Cutting out the middleman has the advantage of reducing overhead costs. Additionally, storing the database at different places reduces the likelihood of hacking and loss of data in case the system goes down. This results in a highly available system, where every node always has the same up-to-date version of the truth. Taking away several nodes will not affect the integrity of the system on its own [15], [17]. Speed is also commonly described as a major benefit of the blockchain technology [16]. Speed is defined in terms of transactional velocity, since blockchain removes all intermediaries who will only slow down the transaction process because they often need to undertake lengthy verification and approval procedures [14]. Data is immediately distributed and agreed upon [15].

2.2 Related Work

While previous literature has touched upon the adoption of blockchain technology for SCM, it has failed to conceptually model these processes for software engineering. For instance, Niranjanamurthy *et al*. [17] discuss how blockchain can meet SC objectives and present a few small case studies to demonstrate how this technology is already used in businesses. The paper nevertheless includes only superficial process descriptions. Other research articles, like Saberi *et al*. [18] and Apte & Petrovsky [14] discuss the use of blockchain in SC and its benefits and challenges but without providing a case study or conceptual model. Bettín-Díaz *et al*. [16], Roa [19] and Casado-Vara *et al.* [20] provide exemplary flowcharts, but these only describe a generic implementation of blockchain in the SC of virtually any company in any industry. Furthermore, Rocha *et al.* [1] and Marchesi *et al.* [21] have tried to model blockchain implementations for a fidelity point program and for the workings of a university group, by using different UML techniques. However, both these cases were mostly fictional and limited.

This article extends on previous research by Ben Hamadi *et al.* [13]. The latter paper studied the use of the i* modeling language for blockchain technology in SC for Blockchain-Oriented Software Engineering (BOSE), based on a case study of a Belgian retail giant. The study in this paper further investigates and elaborates on this notably by applying extensions to i* as proposed by Ben Hamadi *et al.* [13] but not developed in there; this has been done on a genuine case study. Moreover, the present research additionally applies UML as a modeling technique. The latter is widely adopted in businesses for specification stages in software engineering.

3 Research Paradigm, Question and Methodology

Research Paradigm. The research presented here takes roots in the Design Science paradigm [22]; the latter aims to deliver generic solutions for known (or not yet considered) problems. The result of a design science research problem can be a solution in the form an artifact, terminology, methodology, engineering tool, and so forth. In the present research, we have enriched the i* framework to better match with the problematic of blockchain as well as applied i* and UML models for BOSE. Strengths and weaknesses of the models are explored, and a comparison between the frameworks is presented, based on a set of criteria.

Research Question. *Are extended i* models and UML use case/sequence diagrams appropriate modeling techniques to visualize the organizational structure of blockchain ecosystems and how can these two frameworks be extended and integrated to better fit this purpose?*

Research Methodology. To answer the research question, a case study is required [23]. The chosen case study is a Farm-to-Fork project. Farm-to-Fork is a SC tracking prototype that uses blockchain to digitize the food SC and make it more transparent. The Farm-to-Fork project does not

have any technical documentation available, so all information was gathered through interviews. Interviews have been conducted with two experts of blockchain working at the consultancy company to gather the domain knowledge.

A first interview was conducted in February 2020 with Interviewee 1 (I1) a blockchain consultant who has worked on, among others, blockchain projects for the Belgian government. A second interview was conducted in April 2020 with Interviewee 2 (I2), another blockchain expert working at the same company. He provided some additional insights. Out of the information gathered from the interviews, we have elaborated several conceptual models using both i* and UML techniques. A third, final validation session was organized in May 2020 with I2; the latter then validated and confirmed the case study description as well as the associated representations. After applying both modeling techniques to the case study, they have been compared at the light of a set of criteria. As can be seen in Table 1, the criteria are based on three intakes: *generic modeling criteria based on existing literature* [24,25]; *blockchain-specific criteria defined in consultation with blockchain expert* (I2), and *other criteria based on findings from applying both modeling techniques*. Comparing both techniques is useful to determine the pros and cons but also the complementarity of each technique.

Table 1. Evaluation criteria to assess modeling techniques for blockchain in supply chain

Criterion	Description	References
Generic		
Coverage of elements	Whether certain things are difficult or impossible to express. This is about the completeness of the available elements.	[24], [25]
Reusability	Whether models can be reused in a different context.	[25]
Guidelines and tool-support	Whether clear guidelines and tools for the model are available.	[24], [25]
Widespread in different areas	Whether the modeling technique is standardized and generally adopted.	[24]
Expert opinions		
Restricting access and privacy concepts	Whether the model can include privacy concerns.	I1
Scalability	Whether the model is scalable.	I1
Ability to express workflow patterns	Whether a flow or structure can be defined in the model.	I1, [24]
Norms	Whether the model can include the compliance of norms and regulations.	I1
Other criteria		
Social focus	Whether the model can represent the actor's intentions and internal reasoning.	
Dual granularity	Whether the model allows for both a high-level and a more detailed view of the system.	
Flexibility in modeling	Whether the model is not 'deterministic' but allows to model different scenarios.	
Technical concepts	Whether the model has notations to introduce technical concepts.	

4 Case Study

The case study, called Farm-to-Fork concerns a blockchain solution made to track farm animals throughout the SC process, from "their birth to your plate". The solution also includes an easy to use app that gives an overview of the stages of the SC process, including QR-codes to track animals. Every participant in the network, and therefore every node in the SC, can quickly check the origin of the animal, the quality and the different previous steps that the animal has gone through.

4.1 Farm-to-Fork

The Farm-to-Fork prototype was created to meet the increasing expectation levels for improved transparency in the food industry. This solution provides an answer to many of those struggles. I2 suggests that the most important benefits of this implementation are the traceability and the liability aspects. Traceability ensures the ability for the SC participants to closely monitor the animals and allows them to know the exact state and quality of the animal (product). Therefore, it becomes much easier to detect contaminated batches, and to identify any such batches before they can reach the final SC node (such as supermarkets) where they may create a health hazard to unknowing consumers. This also helps to reduce waste. Additionally, I2 remarks that, even if a contaminated product manages to get to consumers, it is much easier to trace down the specific faulty batches, since all product information is meticulously and individually stored in the blockchain. Therefore, in case a contaminated batch would still reach the end-consumers, the health associated consequences will be much less severe.

The liability aspect that I2 mentions refers to knowing all the actions of the SC nodes, including their consequences. For instance, fragile chicken eggs that are transported from node to node throughout the SC can break at any stage. However, disputes can arise between the participants of the SC network about who is responsible for this. With blockchain, these disputes can be settled very quickly as the database can tell when and where every individual egg broke. Furthermore, the advantages do not only apply to the producers, but also to the consumers, since the idea is also to expose a part of the blockchain to them. Consumers can view information about a specific animal product in the supermarket by scanning a QR-code. This enables consumers to verify the origin and all the process steps that the animal has gone through. As consumers become more and more critical about their food intake, it is also becoming more and more important for them to be able to check the authenticity of their food. In this case, consumers could, for instance, check whether chicken eggs come from free range chicken farms or what kind (and how much) of antibiotics they have gotten.

However, it is important to mention that consumers should only have access to restricted, but relevant information. If a consumer would also be able to see exactly how many chicken eggs they have bought from a specific farmer, they might want to skip some parts of the SC cycle and go straight to that farmer for eggs, leaving the rest of the SC nodes redundant and unprofitable. I2 underlines the importance to carefully assign specific access rights to each participant.

4.2 Farm-to-Fork Blockchain Type

The most appropriate blockchain type for the Farm-to-Fork solution is the permissioned enterprise (or private) blockchain. Indeed, such type is only accessible for the responsible participants of the SC. The roles and rights of every node are decided in consultation with all the participants at the start of the blockchain implementation (I1). I2 nuances this by adding that the roles and rights can still be changed afterwards. This can happen through a voting procedure. The rules for such a voting are usually agreed at the start of the project and may, for instance, specify that there needs to be a two thirds majority (2/3) in agreement before a rule can pass. Another example is that all rule changes require unanimous votes. In case the voting is not successful, nothing will change. In any case, in the context of Farm-to-Fork, consumers have fewer rights than the SC actors. All the nodes' identities are known and there is no problem in upscaling the network (I1).

4.3 Farm-to-Fork Raft Consensus

The network works by using the Raft consensus [26]. This is a consensus mechanism that assigns one of three possible roles to every node in the network: *follower*, *candidate* or *leader*. At the start, every node is a follower. Then, an election is held to vote which node becomes a leader. A majority vote is conducted, every node is a possible candidate and the chosen leader is the one responsible

to validate the transactions in the blockchain (I1). I2 adds that every other node can (and will) double-check the transactions afterwards for security, in order to avoid that the leader will validate incorrect information. The election continues and when another majority vote is found, a new leader is chosen. If the newly voted leader does not respond within a given timeframe (usually around 300 milliseconds), then the leader is timed out and a new leader will be elected as well. This mechanism offers to all parties the opportunity to become a leader, and thus becoming responsible for the validation of the blockchain transactions (I1; [26]).

4.4 Overview of the Farm-to-Fork Blockchain Participants

The Farm-to-Fork solution is used in a context of farm animals that go through the SC. For this article, the case study takes an in depth look at the logistic flow of chicken meat from farmer to consumer.

The possible participants for the blockchain project, their respective roles within the SC and their minimum required input into the blockchain are listed in Table 2. This represents a generic model of how the solution works in this context. More (or less) participants could be involved, depending on the needs and context of each specific SC's structure.

Table 2. Farm-to-Fork blockchain network participants

Blockchain participant	**Task**	**Input**
Farmer	Raising chickens.	Characteristics of chickens, the kind of poultry feed used, the sicknesses of chickens and their antibiotics, confirmation of number of caught chickens.
Catcher	Catching the chickens.	Which chickens were caught and in which order.
Transporter to butcher	Transports the chickens from the farm to the butcher.	The conditions of the transportation such as the humidity, the temperature, the shipment status.
Butcher	Prepares the chicken meat.	Number of chickens (slaughtered), treatments, treatment conditions, storage, storage conditions, sizes and volume of the pieces, quality control.
Packager	Packs the meat according to needs of the supermarket.	The amount of meat, the size and volume.
Transporter to supermarket	Transports the chicken meat from the packager to the supermarket.	The conditions of the transportation such as the humidity, the temperature, the shipment status.
Supermarket	Sells the chicken meat to consumers.	Amount of chicken meat and volume sold, stock data, waste data, quality control.

The first participant in the network is the farmer. He is the starting point of the blockchain. The farmer is responsible for raising the chickens on his farm, feeding them, and taking care of them. This obliges him to insert data about the chickens into the blockchain. This data includes the kind of chicken, its age, whether it is free range or not, whether it is biological, as well as the type of poultry feed that is given to the chicken. Additionally, the data should also contain serious events like chicken diseases (more precisely: what kind of sickness, illness period, type and quantity of antibiotics). I1 remarks that some chicken characteristics can be derived from a small IoT machine that is used already by many farmers. Such devices allow to inspect the blood of the chickens and can be linked to the blockchain to automatically register this data [27]. The catcher then catches the chickens of the farmer. The required data from this node describes which chickens were caught (individual identification) and in which order. The order of the catching is important. I1 mentions that during the process of catching chickens (to send them to the butcher), the last chickens have more stress because they realize they will be captured. This increased stress level leads to a lower quality of chicken meat (compared to those chickens that were caught first). So, the order is indeed

important to distinguish the higher quality chickens from the lower quality ones. The farmer will confirm the number of chickens that were caught.

The node responsible for the transportation from the farm to the butcher will need to feed data about the transport conditions into the blockchain. This includes, among other data elements, the humidity and the temperature of the transport vehicle. Furthermore, the blockchain can be helpful to monitor and control the loss or damage of the chickens. Additionally, shipment status information is also required to permit the other nodes to track the chickens. I1 notes that, here too, some information can be sent directly to the blockchain, via interfaces with IoT devices. Such IoT devices can accurately capture the humidity and temperature levels in the transportation vehicle [27].

The butcher who prepares the chicken meat will confirm the number of chickens received and the number of chickens slaughtered. He will slaughter the chickens, apply treatments to the meat and store it. All the treatments and all the conditions of treatments and storage must be recorded into the blockchain database as well. The individual pieces or packs of chicken should be stored into the blockchain by size and volume. Additionally, a thorough quality control is performed.

After the butcher, the packager will pack the meat in conformance with the supermarket's specifications and will record data about the different packages, such as the weight and the number of packages.

The transporter will then transport the chicken meat from the packager to the final node of the SC: the supermarket. Similar data as for the transport from the farmer to the butcher, is required.

Finally, in the supermarket, the meat should all be qualitative enough to sell to consumers. A final quality control is performed. As a general remark, all nodes of the network must identify themselves when entering data. This way, each process step is linked to the organization that is responsible for it. Along every step of the SC, all nodes must perform quality control and report their findings for maximal transparency. Nodes should also enter price data into the blockchain database (I1). I2 adds that some data, like pricing, may be considered as sensitive information. Therefore, he recommends to carefully restrict access to pricing data to those participants who really need this type of information, without making it publicly available.

5 Using i* to Model the Farm-to-Fork Blockchain

5.1 Proposed Extensions for i* for Modeling Blockchains

Two types of extensions of i* are proposed in this article: *privacy*, and *laws and norms*.

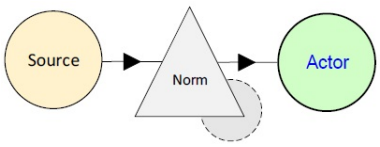
Bashir [15] and Bettin-Diaz *et al.* [16] note that, from the various blockchain hurdles, the privacy issue might be one of the most challenging. The privacy of all nodes in the network must be respected by restricting access to certain data. The nodes themselves should be able to determine which information can be accessed by whom and what information should be anonymized. The importance of privacy should not be overlooked. Therefore, it is recommended that these privacy requirements are explicitly modeled when visualizing blockchain in SCM. Ben Hamadi *et al.* [13] proposed to extend the i* framework by adding the following concepts: *access control*, *privacy accountability*, *confidentiality* and *anonymity* but did not implement it, this is done in this article.

Next to the privacy issues, I1 also stresses the importance of regulations. As blockchain is still a relatively young technology, new regulations that limit the working of the blockchain and/or smart contracts might become applicable. The legal binding of smart contracts in a court of law is often a subject for debate [15]. I1 specifically refers to the repercussions of the GDPR regulations on blockchain. Under GDPR, personal data should remain within the EU. This imposes restrictions on public blockchains because there is no control on where the nodes are located. I1 mentions that this is less of a problem with private blockchains. Additionally, the 'right to be forgotten' conflicts with blockchain as an immutable chain of historical transactions, although this rule lacks a real, strict definition. Currently, a workaround exists whereby personal data is stored

'off-chain', outside the blockchain database. A reference and a hash of this data is then registered in the blockchain. However, this destroys the purpose of the blockchain, since transparency is diminished, data-ownership becomes vague, one need to find a new way to integrate data from other participants and the data is more vulnerable to cyber-attacks. Siena *et al.* [28] and Siena, Perini, Susi, & Mylopoulos [29] have introduced an i* extension to model laws and norms. Siena *et al.* [28] revolves around the application of such extensions specifically for European food traceability systems. This is particularly interesting for blockchain in SCM, as it is important for system developers to understand how the blockchain should be compliant with which regulations. Because blockchain is a technology which steadily becomes more widespread in the IT-landscape, new regulations will emerge to control it legally.

Table 3 provides an overview of all suggested extensions, including their proposed graphical notations. The i* extensions for privacy concepts and for laws and norms are taken over from the literature.

Table 3. Proposed extensions of i* for BOSE

Concept	Graphical notation	Description	References
Generic			
Access control		Access to data in the chain is restricted to certain nodes.This notation can be used on data elements. The annotation allows to specify who has access or who has not.	[13]
Privacy accountability		The notation is used on a dataelement and allows to make third parties accountable for data manipulation under privacy requirements.	[13]
Confidentiality		The data-owner can hide certain data from the other nodes. This notation can be used on data elements.	[13]
Anonymity		An actor wants to anonymize his data partially or completely.The notation is used on an actor element and allows to specify what data should be anonymized.	[13]
Laws and norms			
Norms		An actor should be compliant with a certain norm. This norm also has a source (e.g., EU).	[28]

5.2 Strategic Dependency Diagram – Farm-to-Fork with Blockchain

It should be apparent that, in case of a blockchain adoption for the Farm-to-Fork process, all actors will become connected to each other through the blockchain system. To visualize such a process, the blockchain system itself should also be represented as an actor, alongside the other

participants in the network. The relationship between the nodes in the SC and the blockchain is indeed a dependent one. The blockchain network depends on the farmer, the catcher, the transporters, the butcher, the packager and the supermarket for data. The data is validated and saved into the blockchain. Certain nodes, like the transporter, can depend on the use of IoT devices to automatically capture and send data to the blockchain. For the transporter, this data can include the transportation conditions such as the humidity and the temperature. Based on this data, the blockchain system can also verify whether the contractual terms are fulfilled. The (execution of the) smart contracts therefore depend(s) on the data in the blockchain database. The system can automatically execute the contract through these smart contracts. Because these smart contracts depend on the input data of the SC nodes in the blockchain database, they are also represented as an actor.

Additionally, the blockchain depends on the supermarket to specify the attributes that must be collected by the various stakeholders as input for the blockchain database. On the other hand, the supermarket node also depends on the blockchain data itself, to permit an analysis of the optimal quality requirements (via business intelligence techniques on this data). After the optimal conditions are estimated by the supermarket, the smart contracts need this list of quality requirements to adjust the contract specifications. Moreover, consumers can check the product's history and origin by (partially) viewing the blockchain's data. The SD model of such a SC process is shown in Figure 4. The legend for the elements of this figure are represented in Figure 1.

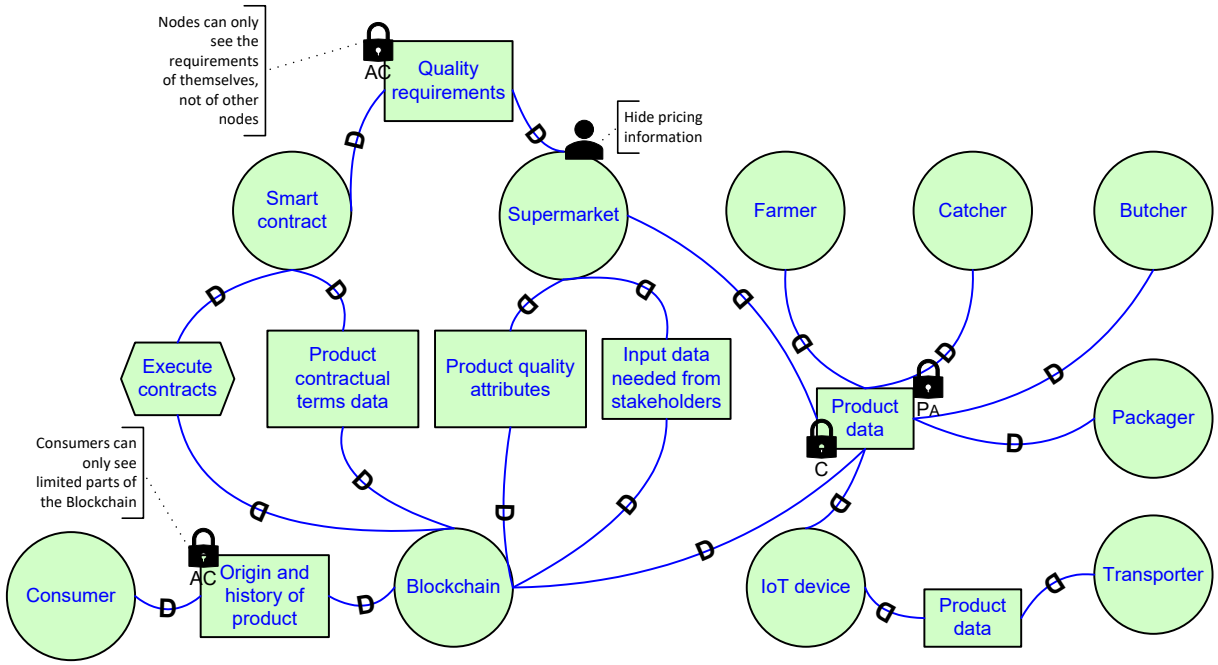


Figure 4. Farm-to-Fork blockchain SD diagram using blockchain and smart contracts

Figure 4 also contains the extensions for privacy concepts. Consumers are only allowed to see a limited part of the blockchain data and process. Next, the quality requirements imposed by the supermarket are only distributed to the relevant nodes, depending on their respective responsibilities within the overall process. The supermarket can also hide its own price data, since this is classified as sensitive information. All nodes can specify what data they want to hide from other nodes, and third parties should be held accountable when given access to manipulate this data.

5.3 Strategic Rationale Diagram – Farm-to-Fork with Blockchain

The SR model focuses more on the internal rationale or reasoning of all the nodes, related to the dependencies between actors.

In addition to the interaction between the different SC participants, the supermarket's ability to specify the quality requirements for each stakeholder is also important and is therefore depicted with the SR model in Figure 5. The legend for the elements of this figure are represented in Figure 1. The SR model focuses on the interdependencies between the supermarket, the blockchain, the smart contracts, and the consumer. The supermarket is an especially important node as the final product arrives here and is sold to consumers. Hence, the chicken meat must be of the best quality in order to sell it to consumers. It is likely that most benefits of the blockchain adoption are experienced in this stage of the SC: no more wastage because of higher quality and avoidance of contaminated products, contaminated products can no longer get into the hands of consumers which limits health risks, and consumer awareness is higher because they can scan the QR-code on the packaging of the chicken meat to check the history of the product. Given these four important actors (the supermarket, the blockchain, the smart contracts and the consumer), the SR model can understand the 'why' of interdependencies.

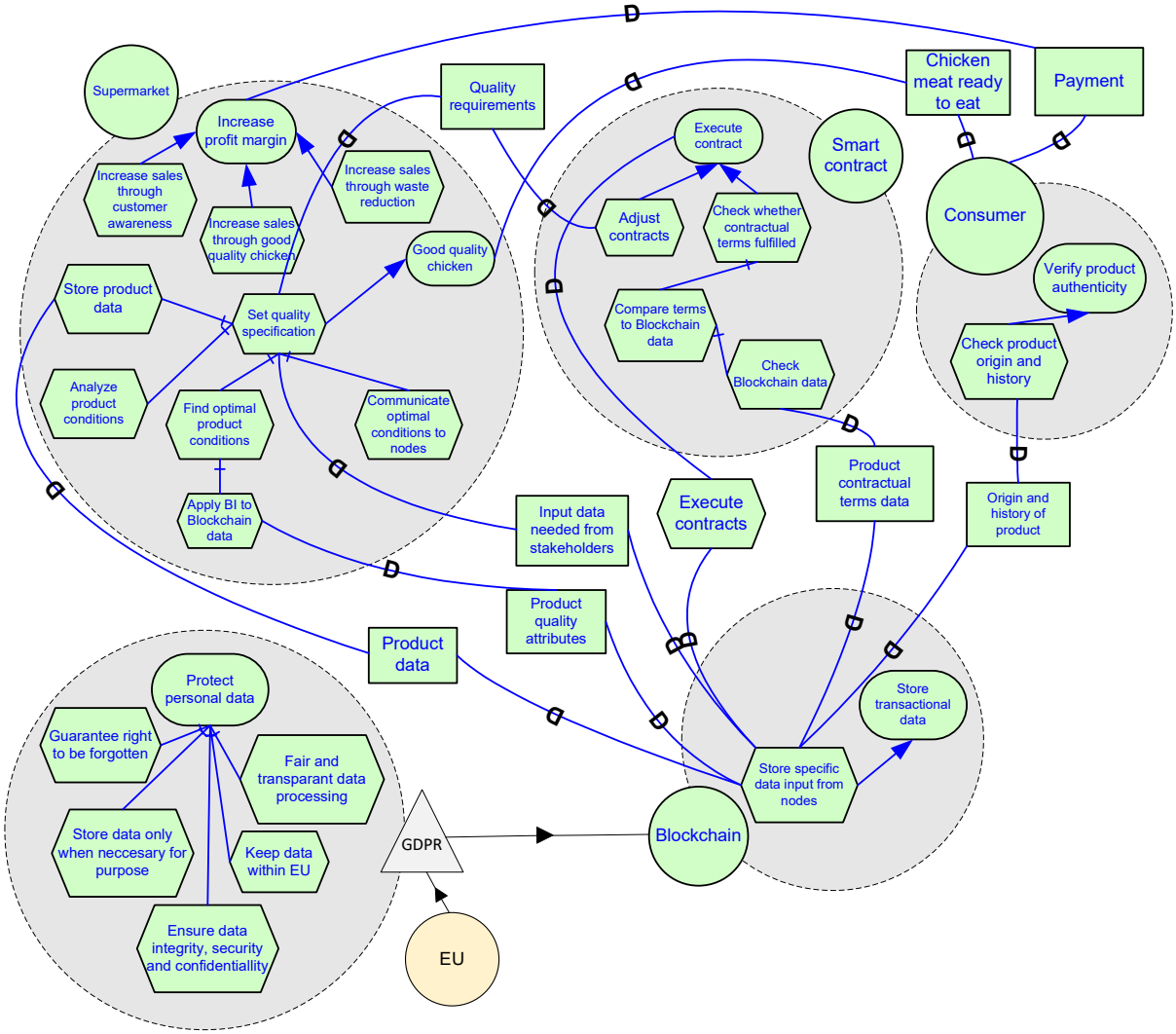


Figure 5. Farm-to-Fork blockchain SR diagram to specify quality requirements

The original i* extension to describe regulatory compliance was specifically targeted towards the SR type of models in i*. Figure 5 shows the integration of the EU GDPR law in the SR model. The overall aim of the regulation is to protect personal data. As shown in Figure 5, this can be achieved through guaranteeing the 'right to be forgotten', keeping data processing transparent, only recording data when necessary, keeping the data within the EU and ensuring data integrity, security and confidentiality.

6 Using UML to Model the Farm-to-Fork Blockchain

6.1 UML Use Case – Farm-to-Fork

The use case diagram is depicted in Figure 6. The legend for the elements of this figure is represented in Figure 2. All network nodes can input, store and verify data. The verification of data can only be fulfilled when a leader is appointed in the Raft consensus mechanism, although every node will double check the verification of the leader (I2). Additionally, the supermarket can provide quality specifications that must be adhered to by all parties. Here, smart contracts are shown as an actor even though they are an integrated part of the blockchain system. This is done to show the possible actions of the smart contracts (i.e., checking whether contractual terms are fulfilled or not and automatically executing the smart contracts). Moreover, consumers can check the history and origin of products by scanning a QR-code.

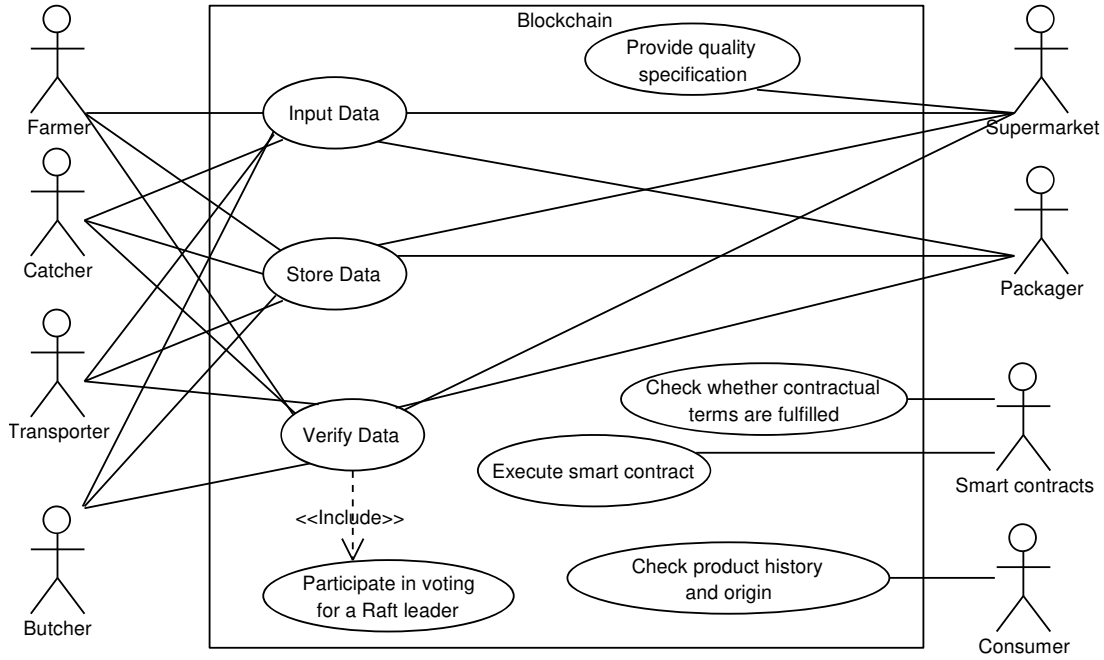


Figure 6. Farm-to-Fork blockchain use case diagram

6.2 UML Sequence Diagram – Farm-to-Fork

The sequence diagram shows the order in which the activities occur. As already mentioned, this is especially useful for SC processes. The Farm-to-Fork sequence diagram is depicted in Figure 7. The legend for the elements of this figure are represented in Figure 3 and details on the supported process can be found in Section 4.

With every blockchain return message 'Verification of data', an alternative fragment should take place, which defines what happens if the verification is successful and what happens if it is not. However, for simplicity reasons, in Figure 7, this alternative (or alt-) fragment is only explicitly modeled for the first occurrence where the blockchain wants to verify data (i.e., at the farmer's data entry). Thus, although not explicitly modeled, this alt-fragment takes place every time the blockchain wants to verify data.

As mentioned before, the transporter can use an IoT device to automatically save and send transportation data to the blockchain. This proposed IoT device is also included in the sequence diagram to show the effects of the implementation.

Finally, at the bottom, a loop is included. This represents the quality requirement updates that the supermarket can repeatedly implement whenever a new (local or global) optimum is found for the process conditions (e.g., by using business intelligence tools).

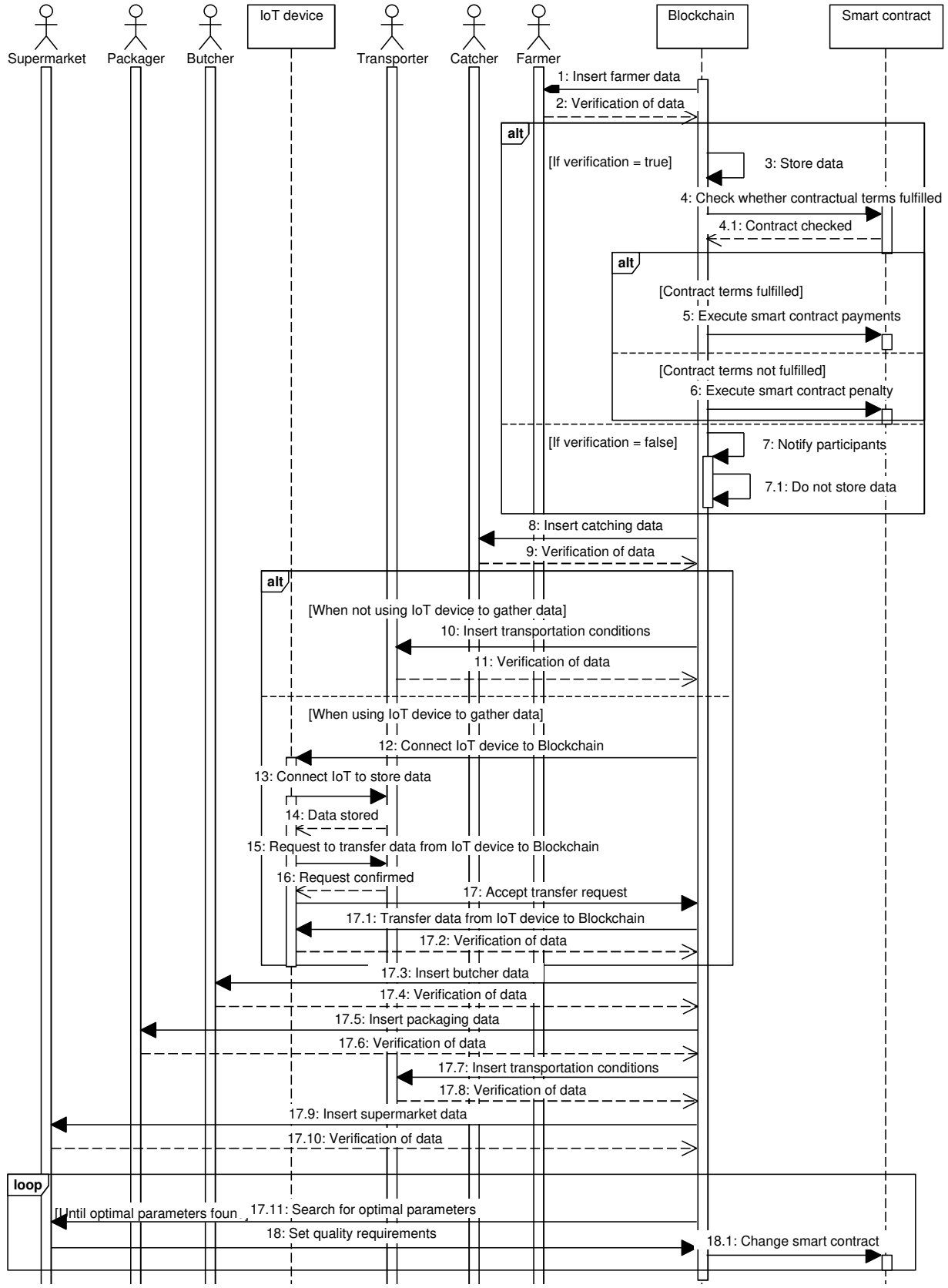


Figure 7. Farm-to-Fork blockchain sequence diagram

Next to the overall sequence diagram in Figure 7, also the Raft consensus mechanism (previously discussed in Section 4.3) can be represented by a sequence diagram. Figure 8 shows the model for this blockchain consensus mechanism. The whole election process loops until a

candidate with the highest term or majority vote becomes leader. A new leader is appointed through a new election when a new majority vote has been found or when the appointed leader times out.

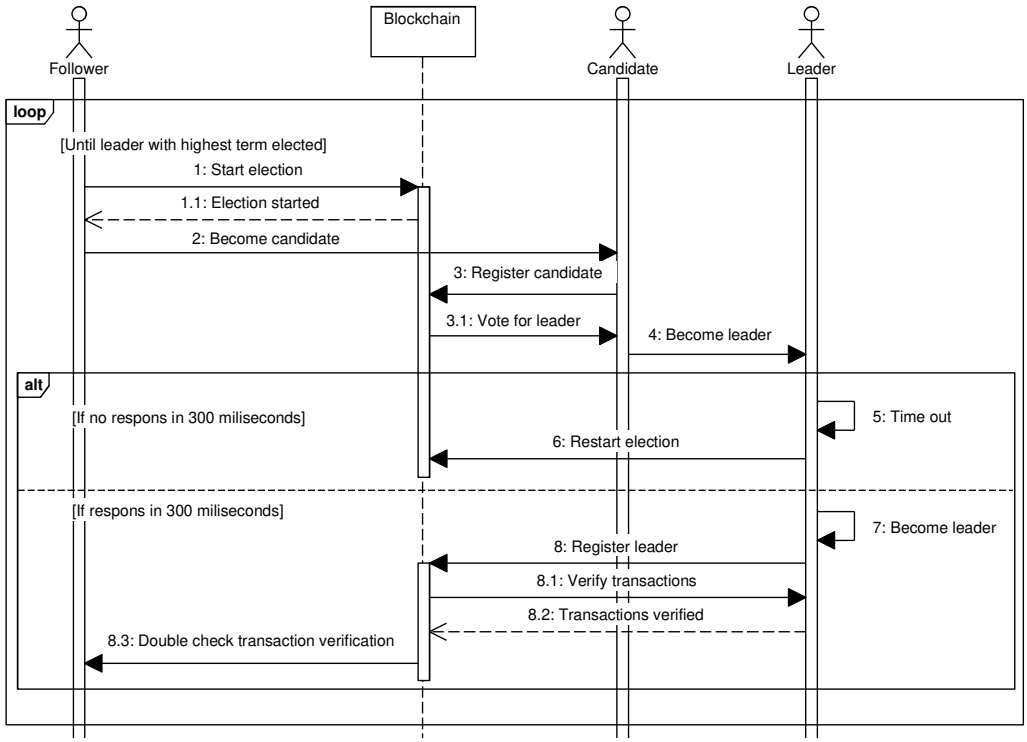


Figure 8. Farm-to-Fork blockchain sequence diagram for raft consensus mechanism

Finally, Figure 9 depicts that consumers can access a limited part of the product history in the blockchain. This can be done by scanning the label of the product in the supermarket with their phone. Using a mobile app, consumers can view some of the processes of the SC.

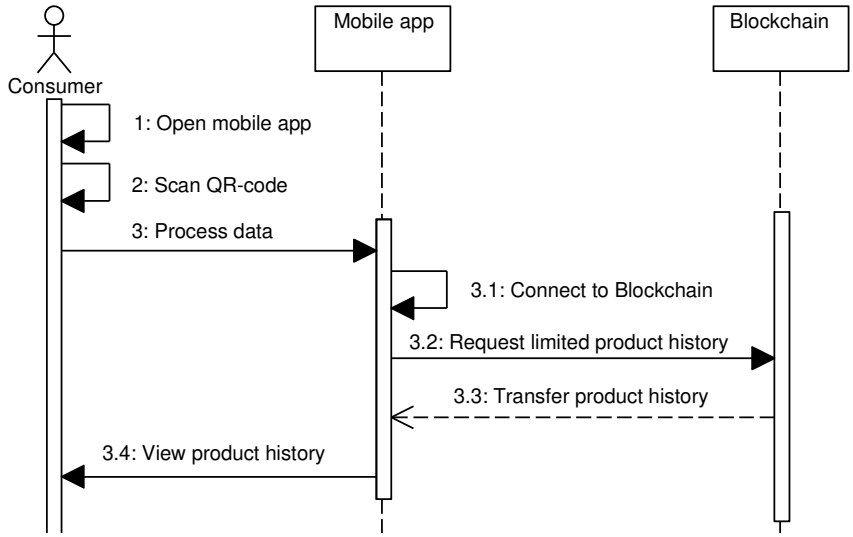


Figure 9. Farm-to-Fork blockchain sequence diagram for consumer access

7 Evaluation of i* and UML for Modeling BOSE in SCM

This section compares the pros and cons of the SD and SR models (i* framework) versus the use cases and sequence diagrams (UML) as modeling techniques for BOSE. The evaluation criteria

have been presented in Section 3. Both frameworks are evaluated using these criteria, then a summary is given.

7.1 Evaluation of the i* Framework to Model Blockchain in SCM

This section maps the characteristics of the i* framework against the set of criteria described in Section 3.

Generic criteria:

- **Coverage of elements.** The internal behavior of the system cannot be represented with i*. Also, both the Raft consensus mechanism and the execution of smart contracts cannot be explicitly represented;
- **Reusability.** Reuse of i* models can be envisaged for different blockchain situations;
- **Guidelines and tool-support.** No CASE-tool provides the explicit extensions for blockchain yet;
- **Widespread in different areas.** The framework has been applied in various areas but barely in the field of blockchain.

Expert opinions:

- **Restricting access and privacy concepts.** The i* representations lack a way to include the blockchain's (here private) typing. Consumers clearly should have restricted access and restricted rights when accessing the blockchain. There is no way to represent restricted access in i*. Moreover, not only consumers, but also other network SCM nodes may express the need to keep certain information private or even to anonymize specific data;
- **Scalability.** A major downside of i* to represent a blockchain-based problem is its growing complexity when the network expands. This disadvantage is not specifically tied to the blockchain field but rather to the nature of the SC. In the Farm-to-Fork case, only seven SC nodes (including the consumer) were identified. In many SCs, the number of nodes can be higher which would increase the complexity of the representation;
- **Ability to express workflow patterns.** The i* framework lacks the notion of structure or sequence which can be seen as a drawback in the context of SCs;
- **Norms.** The i* framework does not allow the modeling of laws or norms that should be respected when adopting blockchain in SCM.

Other criteria:

- **Social focus.** In the context of SCM, the i* framework is particularly suited because every SC involves a network of depending actors. The social focus is here very appropriate. This is also valid for the blockchain system itself and for the smart contracts: interactions of the blockchain within the SC process can be modeled, making it clearer what role the system plays in the SC;
- **Dual granularity.** While the SD model allows for a high-level and more abstract view of the network, the SR one provides a more detailed understanding of the reasoning behind the actions in the process [2]. These two levels permit the modeling of blockchain and smart contracts to be depicted at a high level as well as on a more detailed one. This allows a better understanding of the tasks of the blockchain and smart contracts, and to what goals they contribute;
- **Flexibility in modeling.** The means-end links in the SR model can depict how a certain goal can be achieved through different tasks or different means [2]. Conceptually, this gives room for different options and scenarios to be modeled. The use of blockchain technology in SC offers many options to make processes more efficient so it provides new opportunities for actors to achieve their goals. Being able to flexibly model these means is as a plus of i*;
- **Technical concepts.** The i* framework fails to entirely capture blockchain as a technology, because it is oriented on social concepts. The blockchain database is represented as an actor

in the diagrams. i* lacks dedicated stereotypes to represent specific system components. The inability to depict the blockchain system technically is a major disadvantage of i*. Within the i* framework, it is thus unclear that the actors use the blockchain database to store information about their SC processes using cryptographic means to verify and validate data from other nodes.

7.2 Evaluation of the UML Use Case and Sequence Diagrams to Model Blockchain in SCM

Similarly as for i* in the previous section, this section gives an overview of the pros and cons of using the UML use cases and sequence diagrams for BOSE in the SCM domain. Because the UML use cases and sequence diagrams complement each other, and are often used together, the evaluation of the modeling criteria applies to the combination of both diagrams. The set of criteria for evaluation is described in Section 3.

Generic criteria:

- **Coverage of elements.** UML models provide a broad coverage because they can depict the Raft consensus mechanism, the verification of data and the functions of the smart contracts. This is in line with findings of a study by [1]. Also, the interaction between the blockchain database and other devices such as IoT devices to gather data automatically can be represented with UML diagrams. [21] also points to the sequence diagram as one of the best diagrams to accurately visualize the workings of smart contracts;
- **Reusability.** The UML diagrams are not bound to a specific situation but can be modeled for various purposes and in different contexts;
- **Guidelines and tool-support.** Many sources of explanation such as guidelines about the UML models exist online. Furthermore, many CASE-tools support UML representations;
- **Widespread in different areas.** [1] points out that UML is a widely known and adopted modeling technique, which makes it easy to use for most software engineers, as they are probably already familiar with it.

Expert opinions:

- **Restricting access and privacy concepts.** UML diagrams lack privacy concepts in their modeling techniques. Use case and sequence diagrams fail to specify restrictions of access, anonymity of data or any privacy matters;
- **Scalability.** The combination of the UML use case Diagram with sequence diagrams is scalable: with an increased set of actors, the model will not become overly complex as the models have an organized structure and a specific flow, making it easier to follow even with an increasing number of SC participants;
- **Ability to express workflow patterns.** In the sequence diagram a certain sequence of events or order of actions is established. This fits very well with the flows of SCs as it also contains a strict sequential order. Showing this flow or structure makes it easier to understand the supply chain process, from start to finish. This is also particularly relevant for modeling the consensus mechanism behind a blockchain;
- **Norms.** UML use case and sequence diagrams do not allow the modeling of laws or norms that must be obeyed when adopting blockchain in SCM.

Other criteria:

- **Social focus.** By their nature, UML use cases and sequence diagrams are focused on late analysis and software design, rather than on social aspects. These diagrams are functionally or behaviorally defined for the purpose of building a to-be system. Social aspects, such as intentions behind actions, reasoning of actors or goals cannot be represented;

- **Dual granularity.** Like the i* approach, a dual granularity is possible within UML. The use case diagram depicts a high-level overview of the system while the sequence diagram provides more details about the internal working of the system and the different interactions with the actors. Both diagrams complement each other nicely to provide a better understanding of the requirements;
- **Flexibility in modeling.** The UML use case and sequence diagrams lack some flexibility as they tend to document a unique solution. They do not provide the flexibility of modeling different means to get to one end;
- **Technical concepts.** The two UML diagrams represent interactions of actors with a system. Both the use case and the sequence diagrams are targeted to represent the workings of the blockchain system itself and are able to represent the blockchain as a system.

7.3 Summary

As can be seen in Table 4 (that provides a final evaluation), the two frameworks distinguish themselves by their different purpose: while i* is social-focused, the UML use case and sequence diagrams are system-oriented. Both techniques are complementary. Therefore, we recommend to use the i* framework during the early phases of RE. This enables system developers to understand the 'why' of the SC process, giving a clear overview of the interdependencies and the goals of all nodes in the blockchain network, as this is the core of the blockchain's decentralized system. During later phases, UML diagrams can be applied to design the system's interactions with the different actors of the SC in more detail.

Table 4. Comparison of i* and UML for modeling BOSE in SCM

Criteria	Description	i*	UML
Generic			
Coverage of elements	Whether certain things are difficult or impossible to express. This is about the completeness of the available elements.	-	+
Reusability	Whether models can be reused in a different context.	+	+
Guidelines and tool-support	Whether clear guidelines and tools for the model are available.	-	+
Widespread in different areas	Whether the modeling technique is standardized and generally adopted.	-	+
Expert opinions			
Restricting access and privacy concepts	Whether the model can include privacy concerns.	-	-
Scalability	Whether the model is scalable.	-	+
Ability to express workflow patterns	Whether a flow or structure can be defined in the model.	-	+
Norms	Whether the model can include the compliance of norms and regulations.	-	-
Other Criteria			
Social focus	Whether the model can represent the actor's intentions and internal reasoning.	+	-
Dual granularity	Whether the model allows for both a high-level and a more detailed view of the system.	+	+
Flexibility in modeling	Whether the model is not 'deterministic' but allows to model different scenarios.	+	-
Technical concepts	Whether the model has notations to introduce technical concepts.	-	+

8 Conclusion

Blockchain is still a relatively young technology. Nevertheless, the many benefits of its adoption in SCM are apparent. The immutable characteristic of the ledger provides enhanced transparency, improved product traceability, higher transactional speed, increased security, and an overall cost-effective approach.

Simultaneous with blockchain's rise in popularity, the existing literature about blockchain's adoption in SCM is expanding as well. However, no standard modeling technique exists for the modeling of BOSE, especially with respect to its adoption within SC processes. This article therefore researches two different modeling languages, the i* framework and the UML use cases and sequence diagrams, which appear to be very well suited to model blockchain's adoption in SCM. The article contributes to the existing literature by applying both modeling languages to a blockchain case study and by making a comparative analysis between both. Also a number of model extensions are proposed to enhance the capabilities of i*.

After applying both modeling languages to the case, a comparative evaluation between both approaches was performed. A set of assessment criteria was established and we conclude on the complementarity of the approaches. The in-depth comparison between both approaches has revealed that they also lack some elements to model BOSE in SCM to its full extent. Hence, new graphical concepts are proposed to enhance the models. First, in line with Ben Hamadi *et al.* [13], this paper recommends the inclusion of privacy concepts. Next, because of the importance of laws and upcoming regulations that will determine the future direction of blockchain technology, the enhancement of the i* framework for laws and norms is also recommended.

From the underlying analysis for this article, it is clear that the i* framework can be used during the early phases of requirements engineering, to ameliorate and deepen the understanding of the social aspects, the intentions, and the underlying goals of all actors in the blockchain network. The i* focus on the interdependencies between actors in the network reflects the core of trust in the blockchain's decentralized system. The two UML models lead to late analysis and design diagrams and give an overview of how the system should work, including the interactions between the different actors. The UML diagrams can thus be used, complementary to the i* framework, during later phases of the requirements gathering process. Combined, the i* framework will help to understand the 'why' of the business processes, while the UML diagrams will focus more on the 'what'.

The present study combined with Ben Hamadi *et al.* [13] lead us conclude that i* and its refinements are relevant for each BOSE development in SCM. Future work includes (i) the application of the enhanced i* framework in other domains, (ii) the application of other frameworks, notable the business use case model together with BPMN (see [30], [31]) and (iii) the development of transformation (forward engineering) rules first from the i* SD to the use case model and from the i* SR to sequence diagrams to have an integrated framework. After that, the trace between elements from the i* SR and sequence diagrams and object and agent messages can also be studied to transform the models into code for runtime execution.

References

[1] H. Rocha and S. Ducasse, "Preliminary steps towards modeling blockchain oriented software," in *WETSEB2018*. IEEE, 2018, pp. 52–57. [Online]. Available: https://doi.org/10.1145/3194113.3194123

[2] E. Yu, P. Giorgini, N. Maiden, and J. Mylopoulos, *Social Modeling for Requirements Engineering*. MIT Press, 2011. [Online]. Available: https://doi.org/10.7551/mitpress/7549.001.0001

[3] Y. Wautelet and M. Kolp, "Business and model-driven development of BDI multi-agent systems," *Neurocomputing*, vol. 182, pp. 304–321, 2016. [Online]. Available: https://doi.org/10.1016/j.neucom.2015.12.022

[4] Y. Wang, T. Li, Q. Zhou, and J. Du, "Toward practical adoption of i* framework: an automatic two-level layout approach," *Requirements Engineering*, pp. 1–23, 2021. [Online]. Available: https://doi.org/10.1007/s00766-021-00346-4

[5] M. Kolp, Y. Wautelet, and S. Faulkner, "Sociocentric design of multi-agent architectures," in *Social Modeling for Requirements Engineering*. MIT Press, 2011.

[6] Y. Wautelet, "A model-driven IT governance process based on the strategic impact evaluation of services," *J. Syst. Softw.*, vol. 149, pp. 462–475, 2019. [Online]. Available: https://doi.org/10.1016/j.jss.2018.12.024

[7] Y. Wautelet, M. Kolp, S. Heng, and S. Poelmans, "Developing a multi-agent platform supporting patient hospital stays following a socio-technical approach: Mgmt. and governance benefits," *Telematics and Informatics*, vol. 35, no. 4, pp. 854–882, 2018. [Online]. Available: https://doi.org/10.1016/j.tele.2017.12.013

[8] Y. Wautelet, "Representing, modeling and engineering a collaborative supply chain management platform," *Intl. J. of Info. Systems and Supply Chain Mgmt.*, vol. 5, no. 3, pp. 1–23, 2012. [Online]. Available: https://doi.org/10.4018/jisscm.2012070101

[9] Y. Wautelet, M. Kolp, and L. Penserini, "Service-driven iterative software project management with i-tropos." *J. UCS*, vol. 24, no. 7, pp. 975–1011, 2018.

[10] M. Kolp and Y. Wautelet, "Human organizational patterns applied to collaborative learning software systems," *Computers in Human Behavior*, vol. 51, pp. 742–751, 2015. [Online]. Available: https://doi.org/10.1016/j.chb.2014.11.094

[11] Y. Wautelet, S. Heng, M. Kolp, L. Penserini, and S. Poelmans, "Designing an mooc as an agent-platform aggregating heterogeneous virtual learning environments," *Behaviour & Information Technology*, vol. 35, no. 11, pp. 980–997, 2016. [Online]. Available: https://doi.org/10.1080/0144929X.2016.1212095

[12] OMG, "Omg unified modeling language (omg uml). version 2.5.1," Tech. Rep., 2017.

[13] Y. B. Hamadi, S. Heng, and Y. Wautelet, "Using i*-based organizational modeling to support blockchain-oriented software engineering: Case study in supply chain mgmt." in *The Intl. Research & Innovation Forum*. Springer, 2020, pp. 495–515. [Online]. Available: https://doi.org/10.1007/978-3-030-62066-0_38

[14] S. Apte and N. Petrovsky, "Will blockchain technology revolutionize excipient supply chain management?" *Journal of Excipients and Food Chemicals*, vol. 7, no. 3, p. 910, 2016.

[15] I. Bashir, *Mastering blockchain*. Packt Publishing Ltd, 2017.

[16] R. Bettín-Díaz, A. E. Rojas, and C. Mejía-Moncayo, "Methodological approach to the definition of a blockchain system for the food industry supply chain traceability," in *Intl. Conf. on Computational Science and Its Applications*. Springer, 2018, pp. 19–33. [Online]. Available: https://doi.org/10.1007/978-3-319-95165-2_2

[17] M. Niranjanamurthy, B. Nithya, and S. Jagannatha, "Analysis of blockchain technology: pros, cons and swot," *Cluster Computing*, vol. 22, no. 6, pp. 14 743–14 757, 2019. [Online]. Available: https://doi.org/10.1007/s10586-018-2387-5

[18] S. Saberi, M. Kouhizadeh, J. Sarkis, and L. Shen, "Blockchain technology and its relationships to sustainable supply chain mgmt." *Intl. J. of Production Research*, vol. 57, no. 7, pp. 2117–2135, 2019. [Online]. Available: https://doi.org/10.1080/00207543.2018.1533261

[19] N. Rao, "The time is now," *Quality Progress*, vol. 51, no. 10, pp. 18–23, 2018.

[20] R. Casado-Vara, J. Prieto, F. De la Prieta, and J. M. Corchado, "How blockchain improves the supply chain: Case study alimentary supply chain," vol. 134, pp. 393–398, 2018. [Online]. Available: https://doi.org/10.1016/j.procs.2018.07.193

[21] M. Marchesi, L. Marchesi, and R. Tonelli, "An agile software engineering method to design blockchain applications," pp. 1–8, 2018. [Online]. Available: https://doi.org/10.1145/3290621.3290627

[22] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004. [Online]. Available: https://doi.org/10.2307/25148625

[23] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012. [Online]. Available: https://doi.org/10.1002/9781118181034

[24] Z. D. Kelemen, R. Kusters, J. Trienekens, and K. Balla, "Selecting a process modeling language for process based unification of multiple standards and models," Tech. Rep., 2013.

[25] F. Ruiz, F. van Harmelen, M. Aben, and J. van de Plassche, "Evaluating a formal modelling language," in *A Future for Knowledge Acquisition, 8th European Knowledge Acquisition Workshop, EKAW'94, Hoegaarden, Belgium, September 26-29, 1994, Proceedings*, ser. Lecture Notes in Computer Science, L. Steels, G. Schreiber, and W. V. de Velde, Eds., vol. 867. Springer, 1994, pp. 26–45. [Online]. Available: https://doi.org/10.1007/3-540-58487-0_2

[26] D. Huang, X. Ma, and S. Zhang, "Performance analysis of the raft consensus algorithm for private blockchains," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 172–181, 2019. [Online]. Available: https://doi.org/10.1109/TSMC.2019.2895471

[27] P. Wognum and T. van Erp, *TIVO-Traceerbaarheid van Individuele Varkens in de Organische keten. Een brug naar kennisdeling - Eindrapport*. TIVO, 2013.

[28] A. Siena, N. Maiden, J. Lockerbie, K. Karlsen, A. Perini, and A. Susi, "Exploring the effectiveness of normative i* modelling: Results from a case study on food chain traceability," in *CAiSE2018*. Springer, 2008, pp. 182–196. [Online]. Available: https://doi.org/10.1007/978-3-540-69534-9_15

[29] A. Siena, J. Mylopoulos, A. Perini, and A. Susi, "Designing law-compliant software requirements," in *International Conference on Conceptual Modeling*. Springer, 2009, pp. 472–486. [Online]. Available: https://doi.org/10.1007/978-3-642-04840-1_35

[30] Y. Wautelet and S. Poelmans, "An integrated enterprise modeling framework using the RUP/UML business use-case model and BPMN," in *The Practice of Enterprise Modeling PoEM 2017, Leuven, Belgium, Proceedings*, ser. LNBIP, vol. 305. Springer, 2017, pp. 299–315. [Online]. Available: https://doi.org/10.1007/978-3-319-70241-4_20

[31] Y. Wautelet and S. Poelmans, "Aligning the elements of the RUP/UML business use-case model and the BPMN business process diagram," in *Requirements Engineering: Foundation for Software Quality - 23rd International Working Conference, REFSQ 2017, Essen, Germany, February 27 - March 2, 2017, Proceedings*, ser. Lecture Notes in Computer Science, P. Grünbacher and A. Perini, Eds., vol. 10153. Springer, 2017, pp. 22–30. [Online]. Available: https://doi.org/10.1007/978-3-319-54045-0_2