machines

MDPI

Article

Securing Blockchain-Based Supply Chain Workflow against Internal and External Attacks

Sana Al-Farsi [1,*], Halima Bensmail [1] and Spiridon Bakiras [2]

[1] Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha P.O. Box 34110, Qatar; hbensmail@hbku.edu.qa
[2] Infocomm Technology Cluster, Singapore Institute of Technology, Singapore 138683, Singapore; spiridon.bakiras@singaporetech.edu.sg
* Correspondence: saalfarsi@hbku.edu.qa

Abstract: Blockchain is a revolutionary technology that is being used in many applications, including supply chain management. The primary goal of using a blockchain for supply chain management is to reduce the overall production cost while providing comprehensive security to the system. However, current blockchain-based supply-chain workflow(s) (BSW) are still susceptible to various cyber threats due to evolving business processes of different stakeholders involved in the process. In fact, current BSW protects the supply chain process based on the rules that have been implemented in the corresponding smart contracts. However, in practice, the requirements for the process keep evolving due to several organizational policies and directives of the involved stakeholders; therefore, current blockchain-based solutions fail to protect the supply chain process against attacks that exploit the process-related information that is not protected by smart contracts. Therefore, the goal of this work was to develop a methodology that enhances the protection of BSW against various internal (e.g., Stuxnet) and external (e.g., local data breach of a stakeholder) cyber threats through monitoring the stakeholder business process. Our methodology complements the blockchain-based solution because it protects the stakeholder's local process against the attacks that exploit the process information that is not protected in the smart contracts. We implemented a prototype and demonstrated its application to a typical supply chain workflow example application by successfully detecting internal and external attacks to the application.

Keywords: supply chain workflow; internal attacks; external attacks; blockchain; smart contract

check for updates

Citation: Al-Farsi, S.; Bensmail, H.; Bakiras, S. Securing Blockchain-Based Supply Chain Workflow against Internal and External Attacks. *Machines* **2022**, *10*, 431. https://doi.org/10.3390/machines10060431

Academic Editor: Richard Hill

Received: 31 March 2022
Accepted: 28 April 2022
Published: 31 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Following the development of Industry 4.0 [1], where systems were developed by combining underlying cyber and physical resources, more recent technical developments (e.g., Industry 5.0 [2,3], 5G [4], which aim to extend the role of industry to achieve some societal goals) have evolved the flow of industrial processes. As a result, current processes of supply chain workflow (i.e., involving demand and supply planning, procurement, manufacturing, warehousing, order fulfillment, and delivery business processes) have managed to allow several parties and stakeholders to communicate in a distributed but automated way. The process is highly distributed (i.e., external and third parties can communicate from different locations) and highly dynamic (i.e., any party can join during the process). Despite the significant progress in the development of automated techniques to manage the supply chain workflow, they still cannot be adapted for practical and industrial use due to lack of security, trust, and transparency among the participating parties. In particular, supply chain management's existing industrial workflow processes are only partially automated, implying that some third parties involved in the process may need to be integrated in a different way. Hence, such parties are not trustworthy and their activities cannot be traced for compliance or violation of the business rules and policies. Specifically, current supply chain process management involves several

challenges, including data tampering, single point of failure with a centralized storage, dubious traceability information, distrust of authorities on data authenticity, and lack of provenance [5]. Although blockchain has helped to address these challenges partially through directly supporting traceability, data security, and authenticity, there are internal and external attacks that can compromise the supply chain process and its various entities and stakeholders [6].

The contemporary blockchain technology has been widely used in various application domains (e.g., finance, audit, energy [7], and health [8]) to ensure trust among components of distributed systems [6], on the one hand, and to guarantee traceability and fraud detection [9,10], on the other hand. Blockchain is used as a distributed ledger database for recording transactions among parties that are verifiable [6,11,12]. Furthermore, the blockchain is capable of ensuring data immutability and public accessibility of data stores, which eventually increases the reliability, security, and transparency of each participating entity of the workflow process. More recently, some authors have developed blockchain-based management of the supply chain processes in different application domains, including fresh food delivery [13], pharmaceutical [14], logistics [13,15] and agriculture [16]. However, these solutions cannot be directly applied to ensure a secure supply chain workflow as they do not support transparent and trusted partially automated workflow processes that involve a third party, on the one hand, and fail to ensure consistency of products and business operations (i.e., the supply chain process), on the other hand.

Although blockchain ensures data integrity and provides protection against data tampering attacks by chaining data in a secure-hash way, many supply chain systems have been developed (using blockchain and related technologies) by academia and industries. Most of the academia research has focused on the use of blockchain technology to protect the information exchanged among different business stakeholders. These research efforts have lead to successful detection of attacks that are related to information leakage and tampering [6]. The current blockchain-based supply chain solutions aim to protect communications among distributed components of a business but are limited in detecting any threat that arises from any other layer of the architecture, e.g., the application. Most of the industrial research has focused on developing tools that use blockchain for transparent exchange of information among different supply chain parties by automatically enforcing the communication/transactions. However, these tools are limited in detecting threats that target business level agreements (e.g., agreed terms) and other local threats (e.g., involving stakeholders). Therefore, in this study, we aimed to develop a security monitor that could provide end-to-end protection to a supply chain process that includes stakeholders local protection. We propose a methodology that enhances protection of a blockchain-based supply chain workflow (BSW) against various internal (e.g., Stuxnet) and external (e.g., local but external data breach of a stakeholder) cyber threats through monitoring the stakeholder's business process. Our methodology complements the blockchain-based solution because it protects the stakeholder's local process against the attacks that exploit the process information that is not protected in the smart contracts.

Specifically, the contribution of this work is to develop a novel methodology that:

1. Allows dynamic configuration of the BSW to protect them against internal and external attacks; and
2. Enforces the configurations to ensure that execution of the workflow is indeed protected at run-time.

To the best of our knowledge, this is the first attempt that protects a blockchain-based supply chain process against internal and external attacks.

The rest of the paper is organized as follows. Section 2 introduces our proposed methodology and sketches the workflow of our approach; Section 5 presents an example case study based on the actual requirements of a real local stakeholder. Section 6 demonstrates the effectiveness of the approach in detecting internal and external attacks. Section 7 sketches prior art for workflow management of supply chain systems; Section 8 concludes our work.

2. Proposed Methodology

This section introduces our methodology that helps to protect a supply chain workflow process against internal and external attacks as identified in Section 1. The approach allows us to configure the process on the fly by automatically establishing its trust through making all interactions transparent and traceable, on the one hand, and protecting the workflow against identified attacks, on the other hand.

In detail, the methodology aims to develop a semi-automated mechanism to protect the supply chain workflow, which includes:

1. *Participating entities* that could be internal (i.e., part of the same organization) or external (i.e., third party);
2. *The role of entities* in the workflow process (e.g., supplier, procurement, finance); and
3. *Workflow process* based on the entities and operates as follows:
   (a) Defining dynamic constraints and configurations to secure interactions among participating entities against internal and external attacks; and
   (b) Enforcing the constraints and configurations to protect the execution of the workflow process against internal and external attacks at run-time.

2.1. Constraints

In principle, the constraints can be encoded using expressive language, as depicted in Listing 1. The constraints are valid Java Boolean expressions. One can encode any constraint using the language; however, we aim to encode the following types of constraints:

1. *Basic constraints* that are atomic constraints on the values of various parameters, e.g., length of a parameter, range of the parameter's value, and relationship between two or more parameters and
2. *Advanced constraints* that include complex relationships among different constraints and may require additional computations, e.g., filtering safe participants based on a given list of safe participants.

The former constraints can be encoded using simple Boolean and arithmetic expressions, while the latter constraints may require an invocation call to a Java method that implements such filtering. From the constraints, we generate Java assertions that are checked before processing the actual data (e.g., insertion in the database, retrieval of the data from the database). To ensure that the written constraints are correct, we first run them through the dynamic library of Java that allows us to test whether the given constraints are indeed valid Java expressions and can be executed with the desired input data at run-time. If the constraints do not run successfully, we report the errors so that they can be corrected. Finally, when all the constraints become valid, they are stored in the database.

The constraints ensure the consistency of the execution of the supply chain workflow, even if an adversary compromises the actual execution following a Stuxnet-like attack. Furthermore, the configurations allow protecting the workflow process against external attacks by restricting participation of some of the stakeholders, when their local data have been breached due to an external security incident.

Listing 1. Expressive language for constraints.

$\langle Constraints \rangle \rightarrow$ Expression

$\langle Expression \rangle \rightarrow$ Term | Bool | (Expression) | Expression $\wedge$ Expression
| Expression $\vee$ Expression | $\neg$ Expression

$\langle Bool \rangle \rightarrow$ A Java Boolean expression

$\langle Term \rangle \rightarrow$ A Java expression term

Before delving into the implementation details of our system, we briefly discuss the assumed threat model in the following section.

3. Threat Model

Our threat model involves a Stuxnet type of attack, but also incorporates an external threat intelligence on the involved stakeholders. For example, how to handle a threat that either occurs when the software (ICT) application of a supply chain stakeholder starts malfunctioning (intentionally or accidentally) or when a stakeholder has been declared financially bankrupt or is under attack (intentionally or accidentally). In fact, a blockchain can only record the interactions but it has no control over the business ICT system run by the stakeholders; therefore, it is difficult for it to detect any suspicious activities. For instance, if one of the stakeholders was attacked and we want to exclude them from bidding, we can continue the business activity without changing the actual implementation of the blockchain and smart contracts. Using such constraints, we can simply add the name of the compromised stakeholder in the file, and our system will automatically stop the stakeholder from participating.

A typical supply chain application involves the following two types of interactions:

1. The interactions within the internal processes or parties, e.g., finance, inventory, or warehouse; and
2. The interactions that involve the external processes or parties, such as supplier, vendors, or banks.

Being part of the same organization, the former interaction and associated processes are trusted ones, as they are usually fully digitized and automated. However, the latter interactions and associated processes are untrusted, as they usually require a combination of automated and manual interactions. Our threat model is generic enough to integrate any architecture among the supply chain network, and it is capable of integrating any distributed sub-system, e.g., various warehouses that may be located in different physical locations.

Analogously, a BSW is a typical supply chain application that requires an infrastructure to handle blockchain technology. The infrastructure includes smart contracts that usually ensure secure interactions among parties, and a distributed ledger that stores all the corresponding interactions. The focus of this work is to protect existing BSW systems against internal computational and external communication attacks as depicted in Figure 1, which is adapted from [17].

3.1. Internal Attack

In a supply chain process, a supply chain workflow performs *business computations* that typically implement rules of business processes in an application and *interaction computations* that typically implement interactions in a smart contract. Therefore, the process is protected when its execution is protected.

In an internal computational attack, the adversary aims to compromise the functionality of the supply chain setting by various means. For instance, the adversary may compromise the functionality either by modifying the execution of the workflow process application or smart contracts, or by exploiting any vulnerability or bug in the application or contract, its execution engine or workflow implementation. This attack can be viewed as analogous to a Stuxnet attack [18]. The attack can compromise the internal infrastructure of either the stakeholder or blockchain. The internal infrastructure involves software applications that implement the supply chain process and corresponding smart contracts. Examples of the internal threats are sketched in Figure 1.

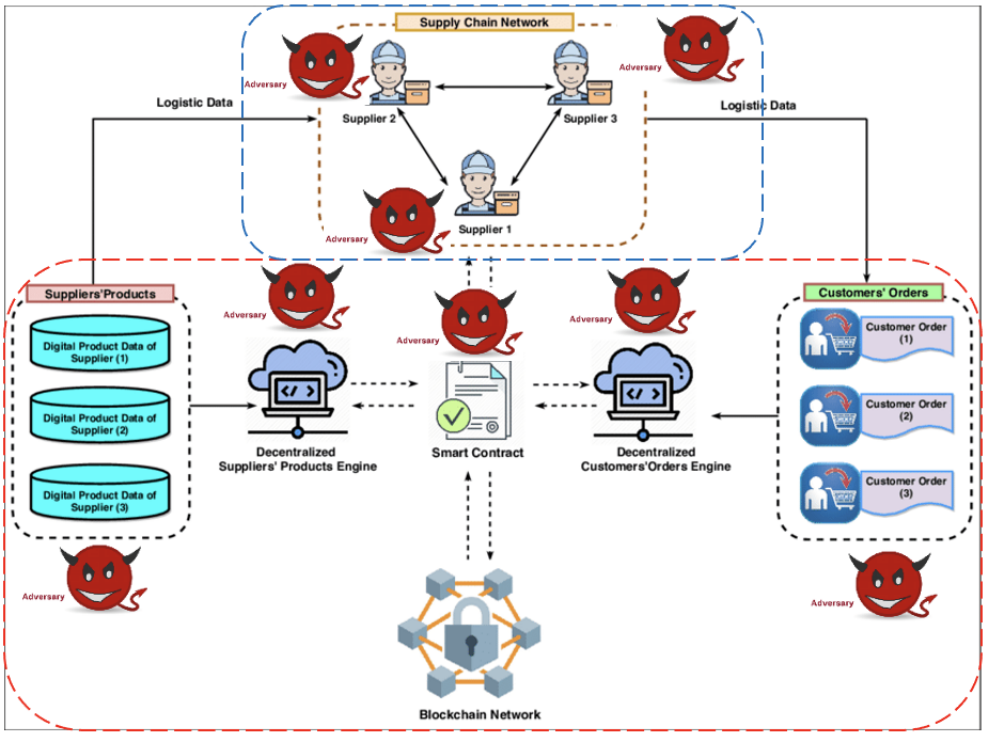

Figure 1. Threat model. Internal threats (red-dotted area): blockchain infrastructure; External threats (blue-dotted area): stakeholder's infrastructure.

3.2. External Attack

The *external communication process* of the workflow is responsible for the exchange of data among different parties (processes or people) involved in the supply chain. In an external communication attack, the adversary aims to compromise the information that is exchanged among various connected services. For instance, the adversary may compromise the information either by leaking the data of the involved parties, tampering input values to smart contracts or other components, or by breaching the integrity of the communication by selective forward and drop tactics, or by injecting false information based on mining of public contracts and the ledger. The attack can compromise the external infrastructure of either the stakeholder or blockchain. In fact, some attacks first compromise the other applications to steal data of the involved stakeholders or blockchain (e.g., customer data breach of British Airways [19]), which are then used by the adversary to compromise the actual stakeholder or blockchain infrastructure, e.g., breach of customer or business data by another attack as evidenced by recent attacks. Examples of external threats are sketched in Figure 1.

To demonstrate our methodology, the above-mentioned attacks may also involve *internal* and *external* attacks and sources. The former one includes Stuxnet-like attacks that can modify the application, while the latter one occurs when some involved stakeholder is under an external attack that causes a cascaded effect on the supply chain workflow process.

4. Implementation

To realize the objectives mentioned in Section 2, we developed a distributed web application that uses an Ethereum [20,21]-based blockchain to record the interactions among supply chain process entities. The key elements of the approach and its workflow are shown in Figure 2. The four main components of the prototype are the following:

1. **Workflow manager application (WMA)** is a Java-based server application that
   (a) Implements all the business rules for the workflow process; and

(b) Allows adding constraints and configurations to protect the execution at runtime; and
(c) Stores interaction-related information and constraints in the MySQL database.

2. **Client** is a web application client that provides an interface for interactions among all corresponding entities involved in the workflow process.
3. **Distributed ledger application (DLA)** is a smart contract-based distributed ledger (Ethereum [20,22]) implemented using Solidity—a blockchain implementation language [21]—that:
   (a) Records all the requests (from the client) and responses (from the server) to prevent process workflow inconsistencies among the corresponding entities and to allow for future audits of the process; and
   (b) Executes smart contracts that automatically enforce the security and compliance of the business process workflow.
4. **Database** is a MySQL-based storage of business information, which is exchanged between the client and the server. It does not need to be recorded on the public distributed ledger; it relates to entities and their data and other sensitive information.

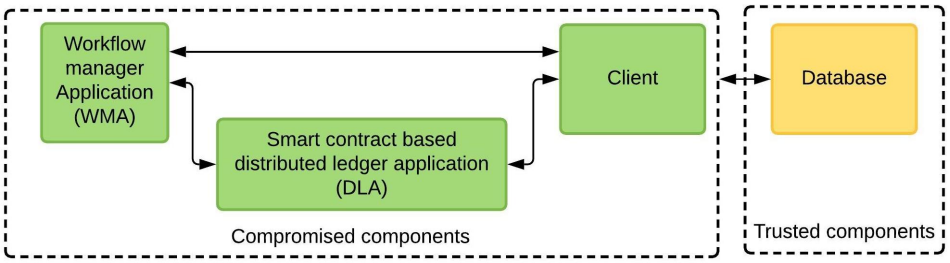

Figure 2. Workflow architecture of the proposed approach.

Note that, in Figure 2, we assume that green components are trusted, while yellow ones may be compromised. The implementation workflow is essentially a two-stage process that is described in the following subsections.

4.1. Stage 1

A client initiates the process (e.g., by requesting the server to register the participating entities and their roles for a given process). The request will be simultaneously sent to the DLA and the WMA, where the WMA will first check if the private and local information of the request (e.g., requesting and requested user, role, organization) is valid, which means that the information respects the required private business rules:

1. *IF* validated, then the DLA
   (a) Checks the compliance of the request with the workflow smart contract;
   (b) Records the desired information; and
   (c) Sends a positive acknowledgement to the WMA to process the request.
2. *IF NOT* validated, then the DLA
   (a) Sends a negative acknowledgement to the WMA to deny the request;
   (b) Records the essential information; and
   (c) Records the negative response information for future reference of the request's trust.

4.2. Stage 2

Once validated by the DLA, the WMA will process the request (e.g., store the necessary information) and will produce the desired response, which will be first sent to the DLA for a compliance check with the contract:

1. *IF* the response is found valid and compliant, then the DLA
   (a) Sends a positive acknowledgement to the WMA; and
   (b) The WMA also sends the response to the client. Importantly, the response includes essential information from the request to ensure that the right client is receiving the response.
2. *Otherwise*, the DLA
   (a) Records the essential information about the negative response; and
   (b) Suspends the process by sending an appropriate signal to the client.

Based on the above request–response style, the entire process workflow (e.g., interactions among entities) will be automatically executed by the smart contracts and recorded on the blockchain-based distributed ledger.

In detail, the WMA application is a Java-based web portal that works as a supply chain management ICT application, and includes the following modules:

1. **Organize module**, which performs the following operations:
   - User registration—by providing the user's details.
   - Organization registration—by providing the organization's information and its role in the supply chain process, e.g., supplier or consumer.
2. **Procurement module**, which performs the following operations:
   - Issuance of purchase requisition—by adding product details that may include up to five items.
   - Review of purchase requisition—by approving the best requisition.
3. **Supplier module**, which performs the following operations:
   - Issuance of quotation—by listing the amount for the product.
   - Preparing supply of the product—by issuing the exact product details of the shipment.
4. **Warehouse module**, which performs the following operations:
   - Inspection of products—by inspecting the received product for compliance with the agreed purchase specification.
   - Paying the cost—by paying the desired amount.

The sketch of the application is shown in Figure 3, which illustrates the process of initiating a purchase requisition. The simplified workflow of the application is as follows:

1. Issue a purchase requisition.
2. Obtain a quotation from different suppliers.
3. Approve the best bid for the product.
4. Obtain the supply from the chosen supplier.
5. Receive the products and inspect them.
6. Pay the amount.

The detailed workflow is explained in the following section through an example case study.

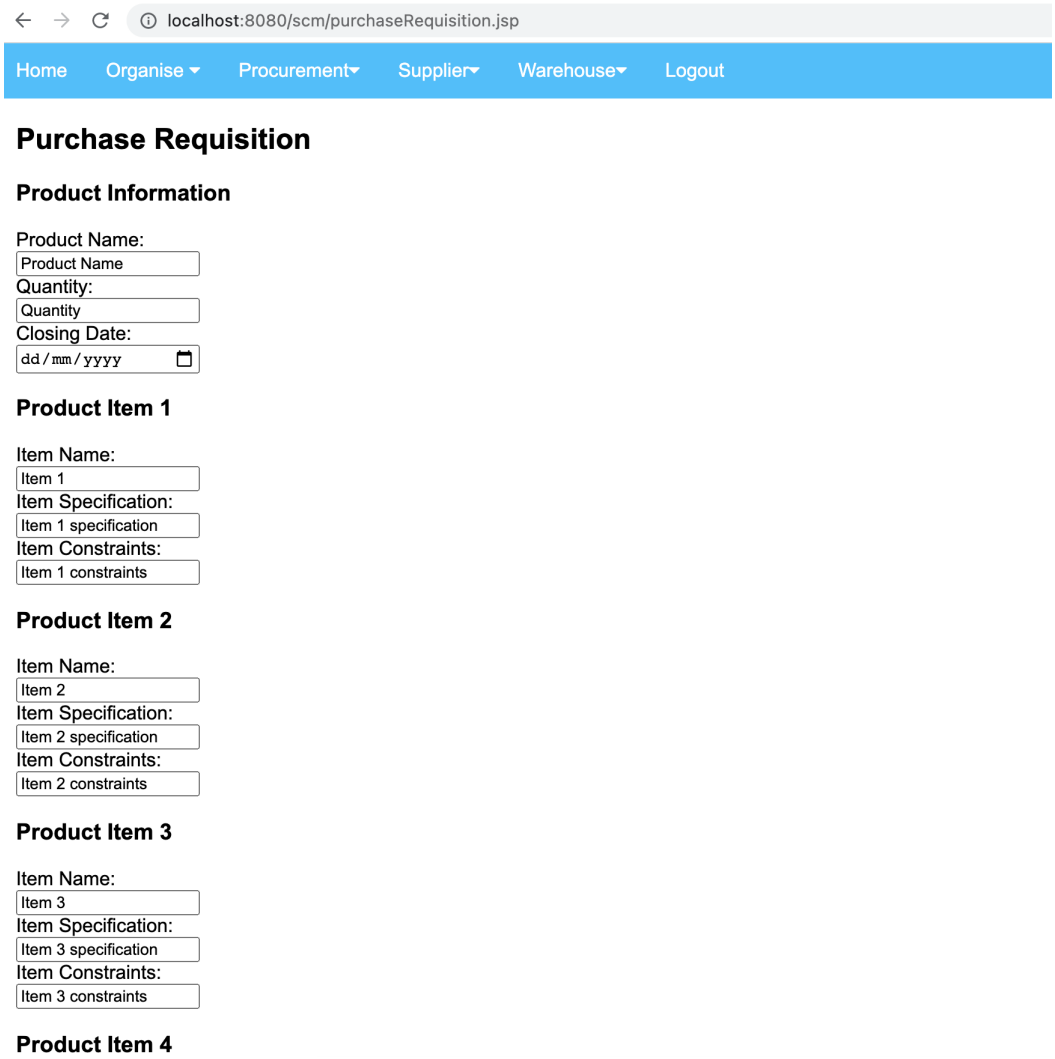

Figure 3. Supply chain ICT application.

5. Example Case Study

In this section, we show the applicability of our application with a case study involving a local stakeholder's real-time supply chain process. In our case study, we developed a prototype to simulate the actual requirements from the real stakeholder. To demonstrate the effectiveness of our methodology, we first describe the current supply chain process as experienced by the stakeholder, followed by a description of how our approach was used to automate the example process.

Supply Chain Process

Based on our discussions with local stakeholders, we developed a specific supply chain process that demonstrates the handling of the supply for various items, including hardware, by a stakeholder. As shown in Table 1, the process involves *internal* and *external* entities, where the former entities are part of the process organization, while the latter entities do not belong to the process organization.

Table 1. Types of supply chain process entities.

Type	Name	Role
Internal	Warehouse	Inventory Management
	Procurement	Supply and Quotation Management
	Finance	Financial Management
External	Suppliers	Management of Supplies
	Manufacturer	Product Evaluation and Quotation
	Others	Product Evaluation

As depicted in Figure 4, the supply chain process involves three stages that are discussed below:

1. The procurement department initiates the demand (supply requirements) of items by issuing a purchase requisition (PR), which is
   (a) Later reviewed by another division of procurement that ensures the specification of the requested items; and
   (b) After review, the PR is sent to the financial department to check the availability of finances to handle/supply the requested items. If the financial department does not approve, then re-evaluation of the PR is requested to the PR initiator.
2. If the financial feasibility of the PR is approved, then various suppliers are contacted after asking for tenders. Later,
   (a) Each supplier issues a quotation for the items as asked in the PR, which is later approved/disapproved by the procurement; and
   (b) For those suppliers, whose quotation is approved by the procurement, they start preparing items for supply along with their documentation, and finally, all prepared supply is delivered to the warehouse of the process organization.
3. When the requesting warehouse receives delivery of the items, then
   (a) The warehouse inspects the delivered items to ensure they received the product as per specification (i.e., PR). The process of inspection of items may involve external entities, for instance, manufacturers who manufactured those products and/or other entities that could provide some certified testing and attestation of the product as per the given PR specification.
   (b) If the products received are consistent with the PR-based specification of those products, then the financial department of procurement pays the amount to the supplier and stores the products in the warehouse after their registration.
   (c) If the products received are not consistent with the PR specification, then the procurement contacts the supplier and investigates the reason for inconsistency, until the matter is settled.

   The settlement is done in various ways, for instance, by replacing the delivered products, opting for another supplier, or a deduction in payment for the breach of contract.

The example supply chain process is typical in various industrial stakeholders. However, the only difference is in the implementation of this process, which is the main hindrance in realizing the process in true spirit. To this end, we have introduced our methodology, which is applicable to any arbitrary stakeholder for attestation of the supply chain process as demonstrated in the following subsection.

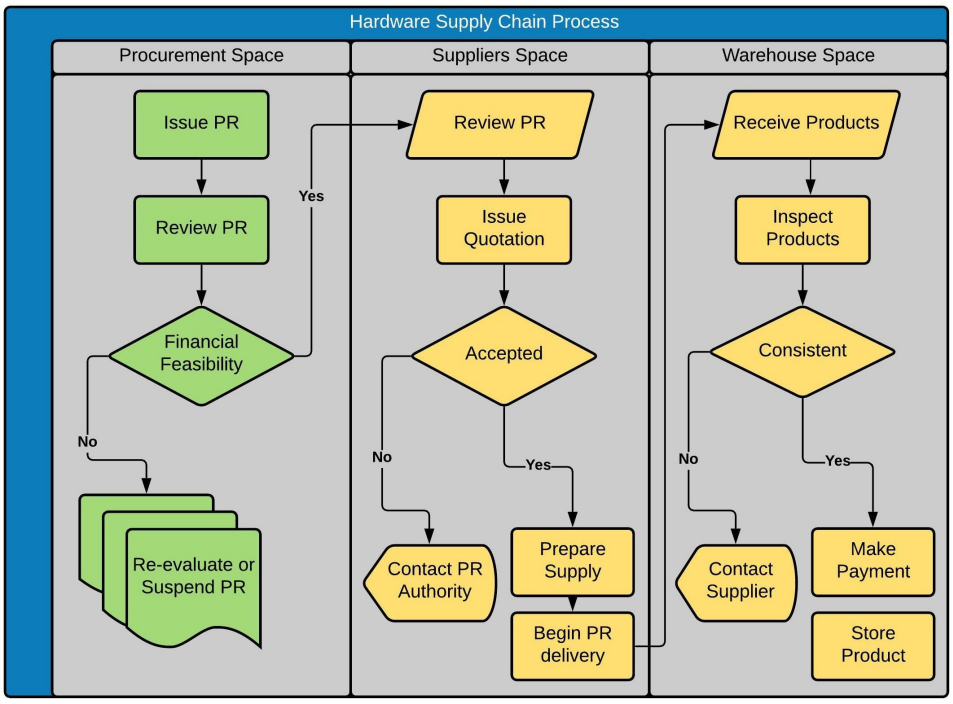

Figure 4. Proposed secure supply chain business process

6. Threat Detection

In this section, we discuss a simulation of example scenarios for internal and external attacks based on the previously-mentioned threat model and our detection mechanism to detect such attacks. Later, we explain how our threat detection mechanism can strengthen blockchain-based supply chain solutions.

6.1. Example—Internal Attack

This section elaborates on an example attack simulation scenario for an internal attack using the following steps, in order.

1. A stakeholder W initiates a supply process for buying some product P by using the ICT application W_a.
2. Stakeholders X, Y, and Z bid for the product P using their apps X_a, Y_a, and Z_a. For simplicity we assume W_a, X_a, Y_a, and Z_a are the same portal and are connected to each other to support digital operations.
   (a) The execution of the applications is protected by a run-time security monitor, which reports any violations when detected. The security monitor's focus is to protect the local ICT applications.
   (b) All interactions among the applications are recorded in a distributed ledger (i.e., blockchain-based) with smart contracts that only protect the collective role (i.e., agreement) of the stakeholders in the supply chain process.
3. Analogous to Stuxnet, an attacker attempts to modify the execution of W_a (attack scenario). For instance, providing incorrect bid values to W_a that the blockchain cannot detect because it does not know the local state of the stakeholder (the attack happens locally in the stakeholder's system) and is not part of the collective agreement among stakeholders.

Our prototype monitor is able to detect this attack, being a violation to the observed current state of the system that is maintained by the constraints through the monitor. Consider the following—we have a blockchain-based supply chain system and now someone tries to modify the stakeholder's application (e.g., malware), as discussed in the scenario above. The blockchain cannot detect such an attack because it does not have access to the

stakeholder's application. Our prototype is able to detect such attacks through internal information, as the monitor uses the constraints (for different values and processes) that need to be respected through monitoring (on top of the blockchain solution). Whenever any stakeholder's application behaviour differs from its constraints, the deviation will be characterized as suspicious activity or an attack. Consequently, to detect such threats, our prototype monitor compares the run-time data with the rules of normal behaviour of the business process to detect such violations that may be suspicious incidents. As sketched in Table 2, our monitor was able to detect internal threats in real-time with an average negligible overhead of approximately 1%. The table shows the time (in seconds) required to complete the execution of a normal operation (as shown in the column "Normal Execution") and also the time required to detect the threats (as shown in the column "Threat Detection"). Threat detection involves fetching constraints from the database and then comparing the constraints based on the application data to avoid their violation. In fact, detection of internal attacks requires monitoring the run-time execution of the actual software applications, which may compromise the real-time performance of the applications. To address this, we translated the constraints into efficient conditions that can be checked in real-time as demonstrated in the results.

Table 2. Evaluation of internal attacks.

Process	Operation	Normal Execution Time (s)	Threat Detection Time (s)	Overhead
Procurement	Issue	0.00201	0.00203	0.9852%
	Review	0.00300	0.00303	0.9900%
Supplier	Issue	0.00309	0.00312	0.9615%
	Review	0.00401	0.004045	0.8652%
Warehouse	Issue	0.00403	0.004071	1.0071%
	Review	0.00406	0.00410	0.9756%

6.2. Example—External Attacks

This section presents an example attack simulation scenario for an external attack, using the following steps, in order. In fact, the first two steps are similar to an internal attack scenario, as discussed above.

1. A stakeholder W initiates a supply process for buying some product P, using the ICT application W_a.
2. Stakeholders X, Y, and Z bid for the product P, using their apps X_a, Y_a, and Z_a. For simplicity we assume W_a, X_a, Y_a, and Z_a are the same portal and are connected to each other to support digital operations.
    (a) The execution of the applications is protected by a run-time security monitor, which reports any violations when detected. The security monitor's focus is to protect the local ICT applications.
    (b) All interactions among the applications are recorded in a distributed ledger (i.e., blockchain-based) with smart contracts that only protects the collective role (i.e., agreement) of the stakeholders in the supply chain process.
3. There is an external attack on Stakeholder W's infrastructure; it has an impact on the supply chain process, e.g., data of their customers have been breached or their system is compromised due to some other attack. The blockchain cannot detect this attack because it cannot directly use this information to protect the system, as it is not encoded on the smart contracts.

Our prototype monitor is able to handle this attack through dynamically reconfiguring the set of active stakeholders in the process, based on the list of safe stakeholders as described in the constraints. In fact, the prototype stops the compromised stakeholders

altogether from participating in the process. As discussed in the above scenario, consider that, in a supply chain process, one of the involved stakeholders is under a data breach attack and becomes vulnerable. The blockchain again fails to detect this local attack. However, our prototype is able to handle such incidents, through external information; that is, whenever a stakeholder becomes vulnerable, we add their identity as a constraint and the system does not allow the stakeholder to participate in the process. As sketched in Table 3, our monitor was able to detect external threats in real-time with an average negligible overhead. Specifically, the overhead of application execution increased approximately by 1%, and the smart contract by approximately 1.15%. The table shows the time (in seconds) required to complete the execution of a normal operation and also the time required to detect the threats in the corresponding columns. Threat detection involves fetching constraints from the network database and then comparing the constraints based on the application data to avoid their violation. Considering that the supply chain application is a web-based application and it requires communication with many other remote components, the overhead is indeed negligible, which proves that the prototype is efficient and can detect the threats in real-time. Furthermore, detecting external attacks is challenging mainly because it may involve some components that are beyond the control of monitor. Therefore, we demonstrated an attack scenario through a clever way, in which we only notified the safe stakeholders to the system.

Table 3. Evaluation of external attacks.

Process	Operation	Type	Normal Execution Time (s)	Threat Detection Time (s)	Overhead
Procurement	Issue	Application	0.00213	0.00215	0.9389%
		Contract	0.00502	0.00509	1.3752%
	Review	Application	0.00310	0.00313	0.9584%
		Contract	0.00510	0.00517	1.3539%
Supplier	Issue	Application	0.00312	0.003152	1.0152%
		Contract	0.00518	0.00525	1.3333%
	Review	Application	0.00407	0.00410	0.7317%
		Contract	0.00520	0.00525	0.9523%
Warehouse	Issue	Application	0.00413	0.00417	0.9592%
		Contract	0.00525	0.00532	1.3157%
	Review	Application	0.00412	0.00416	0.9615%
		Contract	0.00537	0.00543	1.1049%

6.3. Complementary Protection to Blockchain-Based System

In this subsection, we discuss the relationship of our protection mechanism with blockchain-based solutions. In fact, our monitor supports protection to the supply chain process and its stakeholders, complementary to what is protected by the blockchain. For instance, we see the blockchain as a central security monitor that is responsible for protecting the collective behavior of the stakeholders limited to their interactions for a specific process. Our monitor is a local security monitor that is responsible for protecting the local behavior of each stakeholder, thus ensuring the end-to-end protection of the supply chain process. Furthermore, in the case of external attacks, the local/distributed monitor can share the information with the blockchain, which can either reconfigure the system to protect against a specific attack or can initiate other appropriate mitigation strategies. The blockchain is mainly used for transparency, traceability, and auditing of the involved stakeholders to improve their confidence in the system. For instance, in the face of an external attack, all stakeholders can verify that their latter interactions did not involve a compromised stakeholder. The blockchain can also be used as a verification method that

ensures that our protection is indeed enforced. For instance, by providing evidence that only safe stakeholders were involved in a bidding and compromised stakeholders were indeed excluded from the process.

7. Related Work

The key focus of the contemporary approaches has been to employ blockchain to ensure traceability, anti-counterfeiting, trust, and auditing of the entities involved in the supply chain process. To the best of our knowledge, no effort has been made to protect blockchain-based systems against internal or external attacks. Therefore, in this section, we discuss various attempts that have developed secure supply chain processes based on blockchain in various application domains. For instance, Everledger [23] has been developed to track valuable items, e.g., diamonds, gemstones, wines, and artwork. The tool maintains a recorded history of various valuable items that need provenance with transparency. To this end, the tool records ownership of products and their history of access and profile. With blockchain-based records, the items become tamper-proof and their authenticity can be established automatically. However, the focus of the tool is to ensure traceability, provenance, and anti-counterfeiting.

BlockVerify [24] uses private blockchain-based implementation to monitor the distribution of products (e.g., pharmaceuticals, diamonds, and electronics products) including trading partners and product sourcing. The solution is transparent, scalable, and tamper-proof. Before storing product information into a distributed ledger, the system assigns a unique identification code to each product, which is later used to authenticate the product.

In another effort, Verisart [25] has been developed for protection of creation and ownership records of artwork and collectable items. The solution ensures transparency, anonymity, and security of the items. It has support for mobile devices as well, which allows to establish the quality of museum records in two steps. First, the item information is encrypted and then stored in a distributed ledger. The recorded items can be share and transferred anytime by the legitimate users. The focus here is to ensure privacy and security of information from the public.

To ensure data integrity and authenticity of pharmaceutical and logistics items, the blockchain-based supply chain process called Modum [26] has been developed. This product operation requires setting of specific temperatures of the products to activate the Modum sensor and its connection, which is used to scan the shipping IDs of the products. During transit of the items, Modum measures temperature change and stores data in a tamper-proof sensor. When a shipment arrives, the scanning of the items takes place directly from the sensor, which is connected to the secure back-end for the storage and integrity of data. The system aimed to ensure the environmental audit of the shipments that support a high-volume of pharmaceutical products. The focus of the system is to ensure secure item shipments. Recently, Provenance [27] was developed to achieve greater transparency in the supply chain through empowering brands to establish trust-based relationships using blockchain. The solution is targeted to establish trust among farmers, the fashion industry, and the coconut supply chain. To this end, the system requires participant membership. Then, profiles for users and the product information are uploaded. Later, the product information is shared via retailers to showcase various features of the product to make it superior. The focus here is to ensure traceability, provenance, trust, and integrity. Furthermore, it provides a data system to secure storage of information that is unchangeable, auditable, and open.

A summarized perspective of the current research topics on blockchain technology-based supply chain processes are tabulated in Table A1. The benefits delivered by the use of blockchain technology facilitate the delivery of benefits to the beneficiaries. The application areas that are identified in the 'application areas' column are not only limited to the indicated list but support a potential research direction in the field.

Recently, some efforts have been made to develop blockchain-based supply solutions for different items, including hardware. For instance, in [28], the authors developed a

blockchain-based solution for the supply chain of metal (hardware) products. The solution aims to efficiently ensure traceability, transparency, and interoperability among supply platforms. In fact, a distributed digital ledger maintains a record of every transaction, securely and reliably, without the need for third parties that reduce the exposure of the data to hackers. In another work, [29], the authors employed blockchain to ensure a transparent and trusted exchange of items by manufacturers. Usability of the solution was demonstrated through its application in the manufacturing process of card boxes.

Furthermore, some authors have developed blockchain-based supply chain management solutions for pharmaceutical business. For instance, references [30] developed smart pharmaceutical system by employing a blockchain that recommends various drugs in a traceable and auditable way. In other efforts, [31,32] employed blockchain to improve governance of the drugs to improve the trust between patients and health systems. In [33,34], the authors focused on using blockchain to trace the drugs by targeting inaccurate information, lack of transparency, and limited data provenance. Furthermore, their solution ensures the absence of the counterfeit drugs, which not only has a serious adverse impact on human health but also causes severe economic loss to the healthcare industry.

For further case studies, projects, and applications of supply chains in various application domains, the authors kindly direct interested readers to a very comprehensive survey [6] that exhaustively investigated academic and industrial blockchain-based solutions for supply chains.

Despite enormous efforts in developing blockchain-based secure supply chain processes, clearly, the aforementioned approaches are mainly focused on using blockchain to ensure transparency, trust, and authenticity of products in various application domains and not on protecting the blockchain-based supply chain process against internal or external attacks. Therefore, in contrast to the above-mentioned approaches, we developed a solution that

1. Allows to define a customized supply chain process based on participating entities and interactions among them and underlying business policies/rules for the process; and
2. Ensures that the blockchain-based supply chain process is protected against internal and external attacks.

The customized but dynamic constraints allow protecting the system against internal and external attacks that otherwise cannot be detected, with blockchain being unaware of the current real-time state of the system and/or process.

Our approach is comparable to a very recent effort [35], where the authors developed a runtime monitor to protect the blockchain-based supply chain process. However, the developed solution is focused only on detecting internal attacks that arise from various vulnerabilities of the underlying systems (e.g., smart contracts). Furthermore, the approach does not detect internal attacks that can happen to various stakeholders in the process.

8. Conclusions and Future Work

We introduced a novel blockchain-based supply chain workflow management for a supply chain management system that is protected against internal and external attacks. Furthermore, the interactions among participating entities are stored in a distributed ledger to ensure transparency, traceability, and trust among parties. The novelty of the methodology is associated with the fact that it allows for the addition of dynamic constraints that provide enhanced protection to the system against internal and external attacks. We also argued about the feasibility of the approach through its application to an example supply chain process. The developed prototype successfully detected internal and external attacks, in real-time, without compromising the performance of the actual system.

Currently, our monitor may allow some unsafe constraints to be executed, on the one hand, and may not monitor any arbitrary constraint due to limited expressiveness of the underlying language, on the other hand; therefore, as a future work, we plan to allow only safe and valid constraints that do not introduce any extra or insecure behaviours. Furthermore, the real-time performance of the monitor may also be compromised, if a constraint requires communication with external components to verify them.

Author Contributions: Conceptualization, S.A.-F., H.B. and S.B.; methodology, S.A.-F.; investigation, S.A.F.; writing—original draft preparation, S.A.-F.; writing—review and editing, S.A.-F. and H.B.; supervision, H.B. and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the anonymous reviewers and collaborators for their comments and feedback on the earlier versions of this article. We also thank our partners and stakeholders for their discussion and feedback to improve this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Overview of Selected Applications of the Supply Chain Process

Table A1. Overview of the various supply chain process systems.

	Definition	Application Areas	Work	Focus Areas	Benefits	Beneficiaries	Collaborations or Partnerships	Use of technology or Approach	Privacy and Security Concerns
Everledger	A global ledger used to secure and track valuable items [36]	Diamonds, Gemstones, Minerals, Wines, Luxury items, Artwork	It maintains a permanent record of the history of valuable goods and luxury items needing provenance with transparency. To facilitate that, the chronology of the product's ownership is recorded and tailored to involve the appropriate integrity markers. The products become tamper-proof as they are uploaded on the supply chain and it becomes possible to track their history and validate their authenticity [36]	Traceability, Provenance and anti-counterfeit	Verified identity, Secure sharing, Hybrid technology solution, Deep domain knowledge	Real-world solutions for all industries	True Image Solution, Ltd. (TIS)	Latest forensic approaches, NFC, RFID, IoT	Everledger offers secure sharing i.e., making use of advanced encryption and joint compute function to ensure 100% request and response anonymity
BlockVerify	Using private blockchains, BlockVerify monitors the entire distribution network, trading partners, and product sourcing [37]	Pharmaceuticals, Diamonds, Luxury items, Electronics	A private blockchain is used which is highly transparent, scalable, and tamper-proof. Each product, before being stored on the tamper-proof blockchain, is assigned a unique identification code. This number can be accessed by anyone to check the authenticity of the product [37].	Traceability, Provenance and anti-counterfeit	Identify counterfeits, Non duplicable, Companies verify, Global solution	Pharmaceuticals, Manufacturers	Trading partners	Blockchain technology, Customer level authentication, Track-and-Trace (TnT) technology	BlockVerify ensures anti-corruptible product security, i.e., each product is validated and recorded, so that even companies can not forge their own products
Verisart	For protection of creation and ownership records, Verisat Artwork, collectibles	Using mobile device or computer, the museum quality record of the product is created in 2 easy steps. The most trusted ledger of the world then encrypts and timestamps these records. The sharing and transference of certificates can be done at any time. (Varisart, 2018)	Certification and verification of art	Documentation, Access, Privacy, Tagging	Artist studios, Galleries, Auction houses, Cultural organizations	Paddle 8, DACS, Artsystems	Blockchain technology, Encryption of records	Default privacy settings in Verisart keep significant data and information hidden and secured from public view	

Table A1. *Cont.*

	Definition	Application Areas	Work	Focus Areas	Benefits	Beneficiaries	Collaborations or Partnerships	Use of technology or Approach	Privacy and Security Concerns
Modum	For global supply chain operations, Modum provides data integrity and authenticity [26]	Pharmaceuticals, Last mile logistics	Product-specific temperature are set at first place to activate modum sensor and its connection with the parcel after scanning shipment ID. During transit, modum measures temperature changes and stores data in tamper-proof sensor. As the shipment arrives, the scanning of shipment ID takes place to carryout immediate read-out of the sensor. The sensor is connected to the secured back-end for data verification and storing [26].	Environmental audit of shipments	Sensing + monitoring, Events + actions, Prediction + Analytics	Pharma distributors	Swiss Port, SAP, Variosystems, University of Zurich, University of St. Gallen	Emerging technologies, Blockchain, IoT, AI	Modum logger is especially designed to ship high-volume pharmaceutical products in last mile logistics. It is cost-effective and has the highest level security, i.e., to ensure the authenticity of each data point, it is cryptographically secured
Provenance	For greater transparency in supply chain, provenance believes in empowering brands to establish trust-based relationship using revolutionary technologies	Farmers' cooperative, Fashion industry, Coconut supply chain	Membership is requested at the first step to start for free. Once the membership is confirmed, profile is developed and the product information is uploaded. That product information is then shared via retailers to showcase the attributes of the product which make it superior [38].	Traceability, Provenance and anti-counterfeit	Trust, Collaboration, Integrity, Authenticity, Security, Loyalty	Business, Shoppers, Non-profit	Unilever, Sainsbury, Barclays, Standard Chartered [38]	Transparency tools, Traceability system	Provenance provides a data system for secure storing of information that is unchangeable, auditable, and open

References

1. El Hamdi, S.; Abouabdellah, A.; Oudani, M. Industry 4.0: Fundamentals and Main Challenges. In Proceedings of the 2019 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), Paris, France, 12–14 June 2019; pp. 1–5. [CrossRef]
2. Gilchrist, A. *Industry 4.0: The Industrial Internet of Things*, 1st ed.; Apress: Berkely, CA, USA, 2016.
3. Fraga-Lamas, P.; Varela-Barbeito, J.; Fernández-Caramés, T.M. Next Generation Auto-Identification and Traceability Technologies for Industry 5.0: A Methodology and Practical Use Case for the Shipbuilding Industry. *IEEE Access* **2021**, *9*, 140700–140730. [CrossRef]
4. Ghosh, A.; Maeder, A.; Baker, M.; Chandramouli, D. 5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15. *IEEE Access* **2019**, *7*, 127639–127651. [CrossRef]
5. Jabbar, S.; Lloyd, H.; Hammoudeh, M.; Adebisi, B.; Raza, U. Blockchain-enabled supply chain: Analysis, challenges, and future directions. *Multimed. Syst.* **2020**, *27*, 787–806. [CrossRef]
6. Al-Farsi, S.; Rathore, M.M.; Bakiras, S. Security of Blockchain-Based Supply Chain Management Systems: Challenges and Opportunities. *Appl. Sci.* **2021**, *11*, 5585. [CrossRef]
7. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
8. Gordon, W.J.; Catalini, C. Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 224–230. [CrossRef]
9. Bumblauskas, D.; Mann, A.; Dugan, B.; Rittmer, J. A blockchain use case in food distribution: Do you know where your food has been? *Int. J. Inf. Manag.* **2020**, *52*, 102008. [CrossRef]
10. Mirabelli, G.; Solina, V. Blockchain-based solutions for agri-food supply chains: A survey. *Int. J. Simul. Process Model.* **2021**, *17*, 1–15. [CrossRef]
11. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.
12. Pilkington, M. Blockchain Technology: Principles and Applications. In *Research Handbook on Digital Transformations*; Research Handbooks in Business and Management; Olleros, F.X., Zhegu, M., Eds.; Edward Elgar Publishing: Cheltenham, UK, 2016; Chapter 11, pp. 225–253. [CrossRef]
13. Perboli, G.; Musso, S.; Rosano, M. Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases. *IEEE Access* **2018**, *6*, 62018–62028. [CrossRef]
14. Bocek, T.; Rodrigues, B.B.; Strasser, T.; Stiller, B. Blockchains everywhere—A use-case of blockchains in the pharma supply-chain. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 772–777. [CrossRef]
15. Dobrovnik, M.; Herold, D.M.; Fürst, E.; Kummer, S. Blockchain for and in Logistics: What to Adopt and Where to Start. *Logistics* **2018**, *2*, 18. [CrossRef]
16. Caro, M.P.; Ali, M.S.; Vecchio, M.; Giaffreda, R. Blockchain-based traceability in Agri-Food supply chain management: A practical implementation. In Proceedings of the 2018 IoT Vertical and Topical Summit on Agriculture—Tuscany (IOT Tuscany), Tuscany, Italy, 8–9 May 2018; pp. 1–4. [CrossRef]
17. Torky, M.; Hassanein, A.E. Integrating blockchain and the internet of things in precision agriculture: Analysis, opportunities, and challenges. *Comput. Electron. Agric.* **2020**, *178*, 105476. [CrossRef]
18. Langner, R. Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Secur. Priv.* **2011**, *9*, 49–51. [CrossRef]
19. Ukwandu, E.; Ben-Farah, M.A.; Hindy, H.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Andonovic, I.; Bellekens, X. Cyber-Security Challenges in Aviation Industry: A Review of Current and Future Trends. *Information* **2022**, *13*, 146. [CrossRef]
20. Dannen, C. *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*, 1st ed.; Apress: Berkely, CA, USA, 2017.
21. Cai, W.; Wang, Z.; Ernst, J.B.; Hong, Z.; Feng, C.; Leung, V.C.M. Decentralized Applications: The Blockchain-Empowered Software System. *IEEE Access* **2018**, *6*, 53019–53033. [CrossRef]
22. Luu, L.; Chu, D.H.; Olickel, H.; Saxena, P.; Hobor, A. Making Smart Contracts Smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; CCS '16; ACM: New York, NY, USA, 2016; pp. 254–269. [CrossRef]
23. Everledger Uses Blockchain to Help Everyone Trust in What They Buy. Available online: https://www.everledger.io/ (accessed on 8 May 2022).
24. Blockverify. Available online: http://blockverify.io (accessed on 15 May 2012).
25. Verisart. Available online: https://verisart.com/ (accessed on 27 April 2022).
26. Modum. Available online: https://modum.io/ (accessed on 27 April 2022).
27. Ramachandran, A.; Kantarcioglu, M. SmartProvenance: A Distributed, Blockchain Based DataProvenance System. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy, Tempe, AZ, USA, 19–21 March 2018; CODASPY '18; ACM: New York, NY, USA, 2018; pp. 35–42.

28. Mann, S.; Potdar, V.; Gajavilli, R.S.; Chandan, A. Blockchain Technology for Supply Chain Traceability, Transparency and Data Provenance. In Proceedings of the 2018 International Conference on Blockchain Technology and Application, Xi'an, China, 10–12 December 2018; ICBTA 2018; ACM: New York, NY, USA, 2018; pp. 22–26. [CrossRef]
29. Monfared, S.A.R. Blockchain ready manufacturing supply chain using distributed ledger. *Int. J. Res. Eng. Technol.* **2016**, *5*, 1–10.
30. Abbas, K.; Afaq, M.; Ahmed Khan, T.; Song, W.C. A Blockchain and Machine Learning-Based Drug Supply Chain Management and Recommendation System for Smart Pharmaceutical Industry. *Electronics* **2020**, *9*, 852. [CrossRef]
31. Tseng, J.H.; Liao, Y.C.; Chong, B.; Liao, S.w. Governance on the Drug Supply Chain via Gcoin Blockchain. *Int. J. Environ. Res. Public Health* **2018**, *15*, 1055. [CrossRef] [PubMed]
32. Ahmadi, V.; Benjelloun, S.; El Kik, M.; Sharma, T.; Chi, H.; Zhou, W. Drug Governance: IoT-based Blockchain Implementation in the Pharmaceutical Supply Chain. In Proceedings of the 2020 Sixth International Conference on Mobile And Secure Services (MobiSecServ), Miami, FL, USA, 22–23 February 2020; pp. 1–8. [CrossRef]
33. Musamih, A.; Salah, K.; Jayaraman, R.; Arshad, J.; Debe, M.; Al-Hammadi, Y.; Ellahham, S. A Blockchain-Based Approach for Drug Traceability in Healthcare Supply Chain. *IEEE Access* **2021**, *9*, 9728–9743. [CrossRef]
34. Liu, X.; Barenji, A.V.; Li, Z.; Montreuil, B.; Huang, G.Q. Blockchain-based smart tracking and tracing platform for drug supply chain. *Comput. Ind. Eng.* **2021**, *161*, 107669. [CrossRef]
35. Azzopardi, S.; Ellul, J.; Pace, G.J. Runtime Monitoring Processes Across Blockchains. In *Fundamentals of Software Engineering*; Hojjat, H., Massink, M., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 142–156.
36. Diamonds, Blockchain and Banks: The Story of Everledger | BBVA. Available online: https://www.bbva.com/en/diamonds-blockchain-and-banks-the-story-of-everledger/ (accessed on 27 January 2022).
37. BlockVerify Review: Real Anti-Counterfeit Supply Chain Blockchain? Available online: https://bitcoinexchangeguide.com/blockverify/ (accessed on 27 January 2022).
38. Powering Impact-Led Commerce. Available online: https://www.provenance.org/technology (accessed on 27 January 2022).