HYPERLEDGER FABRIC

Spark! Living Lab

Spark! Living Lab Conditioned Goods

Version 0.8

Authors:

Name	Student number	Email
Tim Imming	S1132262	S1132262@student.windesheim.nl
Thijs Mansveld	S1127391	S1127391@student.windesheim.nl
Sjoerd van de Kerkhof	S1129492	S1129492@student.windesheim.nl
Gerard Wesseling	S1131366	S1131366@student.windesheim.nl
Berat Guzel	S1127994	S1127994@student.windesheim.nl

Version control

Version	*Date*	*Who*	*What*
V0.1	19-2-2021	Thijs, Tim, Gerard, Berat, Sjoerd	Initial setup
V0.2	23-2-2021	Thijs, Tim, Gerard, Berat, Sjoerd	Introduction, methods, sub-question 1 and 3
V0.4	26-2-2021	Thijs, Tim, Gerard, Berat, Sjoerd	Sub-question 1 and 3
V0.6	10-3-2021	Thijs, Tim, Gerard, Berat, Sjoerd	Sub-question 2 and 4, word list, abstract, conclusion
V0.8	11-3-2021	Thijs, Tim, Gerard, Berat, Sjoerd	Spell and grammar check, small changes

Distribution

Version	When	Addressee	Reason

Approval

Signature	**Date**	**Name**
	4/06/2021	Maxime Bouillon

Table of contents

Wordlist

Blockchain: A blockchain is a growing list of records, called blocks. Each block contains a hash of the previous block, a timestamp, and transaction data. By design, a blockchain is resistant to modification of its data. This is because once recorded, the data in any given block cannot be altered retroactively without alteration of all subsequent blocks [1].

Ledger: A database that is consensually shared and synchronized across multiple sites, institutions, or geographies, accessible by multiple people. [2]

The Internet of Things (IoT): describes the network of physical objects that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. [3]

Nodes: A node is a device or data point in a larger network. [4]

JavaScript Object Notation (JSON): is a standard text-based format for representing structured data based on JavaScript object syntax. [5]

Application Programming Interface (API): is a software intermediary that allows two applications to talk to each other. [6]

Radio-frequency identification (RFID): uses electromagnetic fields to automatically identify and track tags attached to objects. [7]

Abstract

Supply chains consist of multiple steps; some of these steps include transport by air, road or water. Some of the goods in the supply chain have specific conditions for transport such as temperature or humidity. These goods are so-called conditioned goods because of the specific conditions needed. To monitor these goods, sensors can be used during the transport. To make the sensor data unalterable, blockchain can be implemented to store the measurement data. This research is follow-up research on a study where Hyperledger Fabric was researched as the best-qualified blockchain framework for this purpose. This research paper will research if Hyperledger Fabric is suitable for use in a supply chain as results from the previous study claim. To do research, multiple sources have been used of which primarily websites. The research done confirms that Hyperledger fabric can indeed be used for this use case and fits the criteria.

Introduction

Supply chains are used to transport goods. Companies that offer certain parts of these supply chains must comply with a customer service level agreement (SLA) to ensure the customer that the goods are transported according to certain predetermined guidelines. Data that proves that the SLA's are adhered to are manually recorded in a traditional supply chain, which means that the data is changeable and thus not integer. Blockchain in combination with IoT devices could offer the solution to this. Data that proves the SLA's are met (or not) could automatically recorded by sensors and recorded in the blockchain, which results in unchangeable information.
This research paper will look at Hyperledger Fabric to determine if it could be used to implement in a supply chain with IoT.

This research report is a follow up to a previously written research report by another team of the University of Applied Science Windesheim. They concluded that Hyperledger Fabric was the best option to implement in a supply chain, based on the functionalities and underlaying features it provides. This research report will validate if the conclusion they made is valid and the best approach. Suppose the team finds any underlaying claims that compromise the conclusion of the research report. In that case, the team is potentially forced to redo the previous group's research to find a more fitting solution.

Methods

This Research report will be systematically done; By using mixed methods, research results reliability can be ensured. All the research methodology used in this report can be found in the list below.

- Online Desk research

Online Desk Research:

1. For Online Desk Research, relevant search terms/keywords are formulated.
2. The relevant search terms/keywords that were used (search engines) to find relevant sources on the subject.
3. The data found will be filtered for useful information.
4. This useful information will be processed in the Research report.
   [8]

In this research report, the following search terms/keywords were used:

[G] Google [D] DuckDuckGo

- ‘What is a supply chain’. [G]
- “Hyperledger”, “Hyperledger Fabric”, “Hyperledger Fabric smart contracts” [G]
- “data from a supply chain with IoT” [D]
- “data from a supply chain with IoT” [D]
- “Bacteria sensor” [G]
- “sourcing raw materials for supply chain” [G]
- “supply chain delivery process “[G]
- “supply chain customer support” [G]
- “Hyperledger fabric programming languages” [G]
- “Hyperledger fabric IoT research” [G]
- “Hyperledger composer” [G]
- “Hyperledger Fabric and composer” [G]
- “Hyperledger Fabric vs composer” [G]
- “Hyperledger Fabric Implementation” [G]
- “Hyperledger Fabric resources” [G]
- “Hyperledger Fabric iot research” [G]
- “Hyperledger Fabric supply chain” [G]
- “Hyperledger Fabric SDK” [G]
- “Hyperledger Fabric documentation” [G]
- “Hyperledger Fabric and docker” [G]
- “Hyperledger Fabric network” [G]
- “Hyperledger Fabric kunbernetes” [G]
- “Hyperledger Fabric deployment” [G]

In this research report, the following search engines were used:

- Google (https://www.google.nl)
- DuckDuckGO (https://www.duckduckgo.com)

Research question’s

The research is composed of the questions that are listed below. The main question will be answered with the use of the sub-questions. When all these questions have been answered, then the research is complete.

Main question

Is Hyperledger Fabric suited to be implemented in a supply chain using IoT?

Sub-question 1

What does a supply chain entail?

- What is a supply chain?
- What steps make up a supply chain?

Sub-question 2

What data can a supply chain with Internet of Things (IoT) generate?

Sub-question 3

What is Hyperledger Fabric?

- What is needed to implement Hyperledger Fabric?

Sub-question 4

Which functionalities should Hyperledger Fabric possess to be implemented in a supply chain with IoT?

Results

What does a supply chain entail?

What is a supply chain?

According to Ben Luckovich from TechTarget, "*A supply chain is the network of all the individuals, organizations, resources, activities and technology involved in the creation and sale of a product. A supply chain encompasses everything from the delivery of source materials from the supplier to the manufacturer through to its eventual delivery to the end-user. The supply chain segment involved with getting the finished product from the manufacturer to the consumer is known as the distribution channel*" [9].

A general supply chain is comprised of multiple links. Each link in the supply chain is a company that is driven by supply and demand. These companies want to add value to the product before exchanging it with another company [10]. The different companies in a supply chain are usually located in other geographical locations to keep production costs as low as possible [11]. This means that transport and storage is necessary between each link in the chain before the final product arrives to the customer.

What steps make up a supply chain?

A generic supply chain consists of six fundamental steps [9]:

1. Sourcing of raw materials.
2. Refining the raw materials into basic parts.
3. Combining the basic parts to create a product.
4. Sale of the product.
5. Delivery of the product.
6. Customer support and return services.

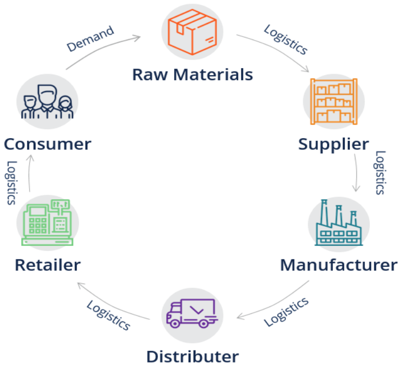

Figure 1: Supply chain [15]

Sourcing of raw materials

Raw materials are unprocessed or minimally processed materials such as wood or iron [12]. These raw materials are gathered by suppliers, who will then sell the materials to other parties [13].

Refining the raw materials into basic parts

Processors will buy the raw materials to change them into a more refined product. For example, iron can be used to create iron sheets. These basic parts can be bought by another processor to create a new basic part. Iron sheets can for example be used to create cut iron sheets.

Combining the basic parts to create a product

Manufacturers can then buy basic parts to combine them into a final product. How many basic parts must be purchased and what basic parts must be bought depends on the final product the manufacturer want to produce [13].

Sale of the product

The sale of fabricated products takes places in shops or web shops of retailers. These products are sold with a margin to profit from the sales and thus bring income for the selling retailer.

Delivery of the product

According to Damon Schecter and Gordon Sander from Inbound Logistics, the supply chain’s delivery step can be viewed with a tri-level view [14]. The three levels of the stage are physical assets, processes, and measurements.

The physical assets consist of the distribution centers, transportation hardware such as trucks or cars, the handling equipment, and the inventory of products. These assets will aid in the process of delivering goods to customers.

The supply chain’s delivery processes use the earlier mentioned assets and operations like assembling products before delivery.

To check if the delivery process is efficient means of measuring the performance is needed. The performance measurement could be based on the cost of delivery and speed or other factors such as customer satisfaction.

Customer support and return services

Most companies in a supply chain have customer support. Customer support is necessary for the contact between customers or buying company and the delivering company and for receiving feedback on the products delivered.

What data can a supply chain with Internet of Things (IoT) generate?

Supply chains that have Internet of Things (IoT) integrated could generate a lot of data. The data generated by the chain can be combined or used independent to gain knowledge about the cargo. Data is categorized in multiple categories and are elaborated on below. These categories have been composed through brainstorm sessions in combination with online sources.

Figure 2 shows the different processes in a supply chain. The actors use these processes to generate data.

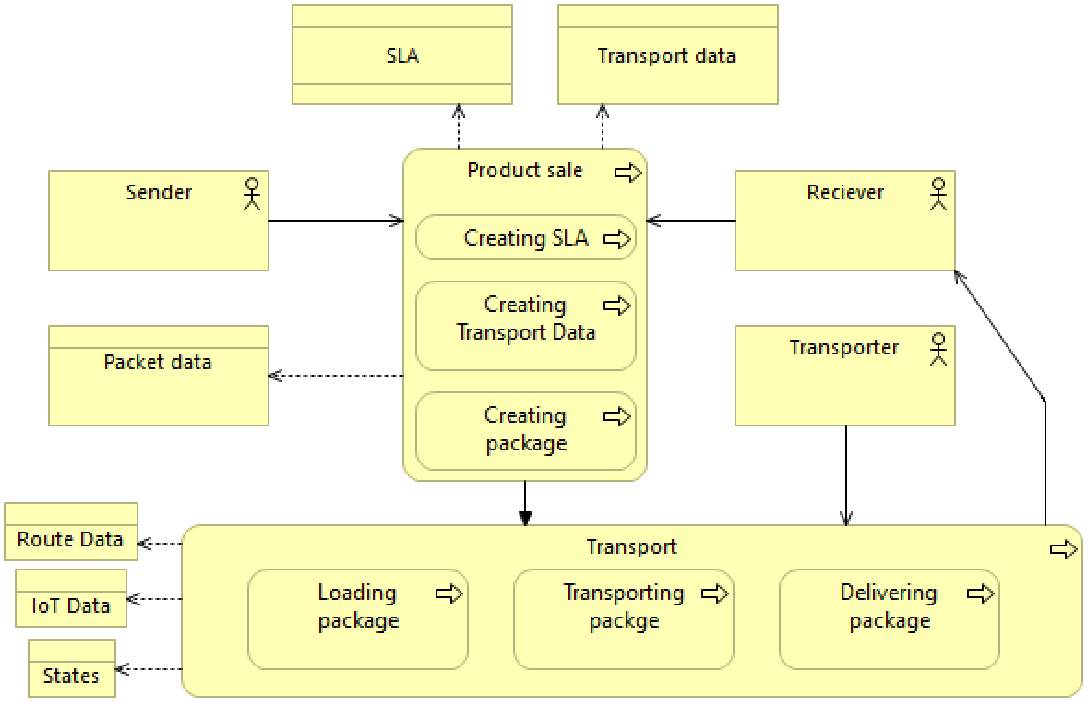

Figure 2: Supply chain data

States

The States category contains all the data needed to monitor the package state. This data will be collected with the use of sensors. These sensors can be either in the package or the container in which the parcels are transported in. Events such as opening and closing packages and the containers are registered or whether these are damaged. [15]

Transport Data

Transport Data contains all the data necessary for the transporter to deliver the shipment successfully. This category includes data such as receiver (name, address etc.), sender and the return address. Without this data the company that transports the package cannot deliver it.

IOT Data

IOT Data is a collection of all the possible data that the different sensors could compose. These sensors can be in the truck, packet or anywhere else in the environment. The sensors could produce data such as acceleration [15], location [16], or radiation levels [15]. The data will be used to monitor the circumstances of the package and the environment around it.

Packet data

Packet Data includes all the information about a packet. This information can be used to identify the packet. Data included could be the weight, packet I.D. of the package, and dimensions. These packets are easily identified and tracked with RFID [16].

Route Data

Route Data contains all the data necessary to route the packet during transport, to transport it successfully to the correct location. The data includes the next delivery point and the final destination.

Service Level Agreement (SLA)

A Service Level Agreement is a commitment between a service provider and a client. It contains all the agreements the service provider should adhere to during the transport of the goods [17]. Data such as expected delivery time, maximum deviation delivery time and specific delivery location could be included in a service level agreement.

The table below lists all the different data of a supply chain divided into categories. Each category shows a collection of possible data the category could contain. This does not mean all the data will be available in a specific supply chain; some of the data could be optional and is different per supply chain.

Packet Data	**States [18]**	**IoT Data [18]**	**Route Data**	**Service Level Agreement**	**Transport Data**
Weight	Package opened	Atmosphere composition	Next delivery point	Expected delivery time	Receiver
Packet ID	Content of package opened	Electromagnetism [15]	Destination [18]	Maximum deviation delivery time	Sender
Dimensions	Damaged	Radiation levels [15]		Contents of package	Transporter
Packaging material		Humidity [19]		Packaging material	Transport ID
Packaging		Acceleration [15]		Minimum/maximum deviation from IoT Data	Shipping date
Shape		Ambient light [19]		Minimum/maximum states	Expected arrival date
Monetary value		Orientation [15]		Cost	Shipping location
Description		Air pressure [15]		Warranty	Return address
Barcode/RFID **[16]**		Temperature [19]		Additional insurance	
Part of order		Location (GPS) [20]		Specific delivery location	
		Bacteria [21]			
		Pressure on packing material [15]			
		(External measurements) [15]			

What is Hyperledger Fabric?

Hyperledger is an umbrella project for blockchain founded in 2015. The project contains open-source frameworks and tools. The Linux Foundation is in charge of maintaining the different subprojects. Hyperledger is part of a cross-industry collaboration to develop blockchain-based ledger solutions. Notable contributors to the frameworks and tools are IBM, Intel, and Cisco.

One of the frameworks developed by the foundation is Hyperledger Fabric. Hyperledger Fabric is a foundation for a modular permission blockchain infrastructure. That means it allows components such as consensus and membership services to be built in. Because of the modularity it provides, the framework can be implemented in a wide range of use cases. Hyperledger Fabric also offers the option to establish smart contracts, or how Fabric calls them “chain code”. These contracts regulate the access to the ledger (who can access what) [22] [23] [24].

The following functions are available in Hyperledger Fabric [25]:

Identity Management

To divide sensitive data between many parties, Hyperledger Fabric provides users with a membership identity service. With this service, the user can manage all the participants in a network through user I.D.s and authentication. Hyperledger Fabric also provides access control lists for additional layers of permissions. This means it is possible to grant specific user IDs different permissions, such as retrieving a chain code or blocking them from deploying new chain codes.

Privacy and confidentiality

Hyperledger Fabric makes it possible to have multiple competing companies in the same permissioned network. To coexist in the same network, these company must protect their sensitive data from competitors. To solve this issue, Hyperledger Fabric allows private channels. Private channels are restricted messaging paths that can be used to achieve transaction privacy and confidentiality for subsets of network members. All data (transactions, members, channel information) are only visible or accessible to members granted with the corresponding private channel.

Efficient processing

In Hyperledger Fabric transaction execution, ordering and commitment are separated. The transaction execution has a higher priority than ordering them; this enables each peer node to process multiple transactions simultaneously. This increases processing efficiency and accelerates the delivery of transactions which is called parallel processing.

Hyperledger Fabric’s architecture defined multiple types of nodes with their own tasks. Some nodes are tasked to order transactions, so peer nodes are freed from ordering workloads. This behaviour results in limiting the processing required for authorization and authentication. All peer nodes do not have to trust all ordering nodes, and vice versa, so processes on one can run independently of verification by the other.

Research papers have demonstrated that Hyperledger Fabric is scalable up to at least 20.000 transactions per seconds by studying and testing performance capabilities [26].

Chain code functionality

Chain codes define many rules in Hyperledger Fabric. Chain code can be applied on several levels, such as: parameters for changing asset ownership or operating parameters for entire channels (system chain code). Lifecycle and configuration system chain code defines the channel's rules; endorsement and validation system chain code defines the requirements for endorsing and validating transactions.

Modular design

Hyperledger Fabric is built in a modular way. This modularity provides network designers to implement it in their own way using specific algorithms for identity, ordering (consensus), and encryption of their choice to plug into Hyperledger's network. This helps Hyperledger Fabric to be adoptable in many markets.

Features

Beside the function listed above, Hyperledger Fabric also provides several additional features. These features differentiate Hyperledger Fabric from the other distributed ledger technologies. The features include [27]:

Permissioned architecture

Hyperledger Fabric is built from the ground up based upon the principle privacy. This means Hyperledger Fabric's architecture is designed to handle different levels of privacy by using permissions to separate and protect sensitive data. This can be accomplished by using channels or private data.

Highly Modular

Hyperledger Fabric is designed to be highly modular. This allows developer to implement Hyperledger Fabric in several enterprises and use cases by adjusting its architecture.

Pluggable consensus

Hyperledger Fabric offer user a choice on which consensus mechanisms they want to use. Consensus is the way nodes communicate in a network about the ordering of transaction. Because Hyperledger Fabric is opensource developer can also implement their own consensus solution to optimize performance [26].

Open smart contract model

Hyperledger Fabric's smart contracts are designed to provide developes with unrestricted flexibility. Therefore developers can implement any desired solution modal such as: account-model, UTXO-model, structured data, unstructured data, etc. [28]

Low latency of finality/confirmation

"The lottery-based algorithms are advantageous in that they can scale to a large number of nodes since the winner of the lottery proposes a block *and transmits it to the rest of the network for validation. On the other hand, these algorithms may lead to forking when two "winners" propose a block. Each fork must be resolved, which results in a longer time to finality.*

The voting-based algorithms are advantageous in that they provide low-latency finality. When a majority of nodes validates a transaction or block, consensus exists and finality occurs. Because voting-based algorithms typically require nodes to transfer messages to each of the other nodes on the network, the more nodes that exist on the network, the more time it takes to reach consensus. This results in a trade-off between scalability and speed." [29]

Flexible approach to data privacy

Because Hyperledger Fabric's architecture is designed to handle several levels or privacy by being permission based, this allows developers to use a few ways to isolate data. One way to isolate data is channels. Channels are groups of organizations with their own ledger. This means only organizations in the channel can access its data. Another way is private data collections. Private data collections allow developers to share data on a need-to-know basis. Only the organizations that is obligated to receive the data is send the data [26].

Multi-language smart contract support

Hyperledger Fabric offers support for multiple programming languages to tailor to a wide range of developers. Languages for smart contracts are: Go, Java, JavaScript.

Support for EVM and Solidity

For developers that are experienced with blockchain solutions such as Ethereum, Hyperledger Fabric allows them to use Hyperledger Fabric permissioned blockchain platform to interact with Ethereum smart contracts written in an EVM compatible language such as Solidity or Vyper [30].

Designed for continuous operations.

Hyperledger Fabric is designed to support rolling upgrades and asymmetric versioning. This means developers can update their applications while keeping their application online and keep supporting outdated versions without forcing the user to update to a later version [31].

Governance and versioning of smart contracts

In Hyperledger Fabric updating or accepting a smart contract within the network Is no longer the task of one node, multiple organization must agree on the parameters of a chaincode. A contract will only be updated if a certain number of organization accept is, instead of a single organization [32].

Flexible endorsement model

Endorsement policy describs the number of peers on a channel that must execute the chain code and endorse the results to conder a transaction valid. Hyperledger offers multiple ways to require endorsement. Endorsement policy can be defined on chain code, collection of even key level [33].

Queryable data

Hyperledger Fabric allows developer to query data based on keys by using the default database (levelDB). It also offers supports for JSON based queries by allowing developers to switch to CouchDB (a JSON database) to support more rich and complex queries. [34]

Programming languages

Hyperledger Fabric was initially built using the Go language (a programming language developed by Google); they aim to support as many programming languages as possible so that developers familiar with the language can start working on chain code (business logic agreed to by members of the network) rather than learning a new language. To achieve this, Hyperledger Fabric offers several APIs & APKs to support developers in various programming languages. [35] [36]

APIs

Hyperledger Fabric offers several APIs to support developing smart contracts (chain code) in various programming languages. Currently, Hyperledger Fabric has smart contract APIs available for Go, Node.js, and Java. Furthermore, a wide array of documentation is available for developing applications using these APIs. [36]

SDKs

Hyperledger Fabric has software development kits (SDKs) available; an SDK is used in applications to access APIs that permit queries and updates to the ledger. You can install and use the Hyperledger Fabrik SDKs to develop an application that interacts with a Fabric Blockchain network. It provides simple APIs to submit transactions to a ledger or query the contents of a ledger with minimal code [37]. Currently, Hyperledger Fabric has SDKs for Node.js and Java, but it also has plans to release SDKs for Python, REST, and Go in future releases. Furthermore, a wide array of documentation is available for developing applications using the SDKs. [38]

Hyperledger Composer

Hyperledger Composer is a set of collaboration tools for building blockchain business networks. It provides a GUI (Graphical User Interface) called “Playground” for the creation of applications and therefore represents an excellent starting point for Proof of Concept work. Composer is essentially a tool that is a simplified way of working with Fabric. Under the hood, Composer calls the Fabric APIs and is working on top of Fabric. Unfortunately, as of the 29th of August 2019, the Hyperledger Composer project's status is deprecated. Meaning that none of the maintainers is actively developing new features for the project. [39] [40]

What is needed to implement Hyperledger Fabric?

Hyperledger fabric is very flexible, which leads to very complex network setups. This unfortunately leads to difficulties in deploying on production environments. Not completely understanding the network can lead to network configuration choices which can destroy the purpose of using blockchain in its entirety. The following is a list of possible solutions for ensuring Hyperledger fabric deployment goes as smoothly as possible.

IBM BP

The IBM Blockchain platform (IBM BP) provides an environment within Visual Studio Code (VSC) to develop, package, and test smart contracts. the provided tools can be used to create a smart contract and get started developing the business logic. After developing the business logic it is possible to use the tools provided by IBM BP to test the smart contract. The test environment can be initiated on a local machine by using a preconfigured instance of Hyperledger fabric, or by connecting to an IBM BP network, before deploying the smart contract to the IBM BL platform. [41]

Docker

Docker is an open-source platform for developing, shipping, and running applications. Docker makes it possible to separate applications from the infrastructure, so software can deliver Quicker. With Docker, it is possible to manage the infrastructure in the same ways one would manage their applications. By taking advantage of Docker’s methodologies for shipping, testing, and deploying code quickly, significantly reducing the delay between writing/testing code and running it in production. [42]

Kubernetes

Kubernetes is an open-source container platform, which simplifies orchestration in cloud nativer environments, for automating deployment, scaling, and management of containerized applications [43]. Kubernetes is compatible with Docker, which is used by many official Hyperledger Fabric tutorials building Hyperledger Fabric networks. [44]

Which functionalities should Hyperledger Fabric possess to be implemented in a supply chain with IoT?

How do we know that Hyperledger fabric is suitable for a supply chain? Well we know what data a supply chain generates and we know what kind of demands a supply chain has in order to work. Combining these two will result in functionalities that Hyperledger Fabric must possess. These functionalities are elaborated down below.

Support different datatypes

To store all the data and documents that a generic supply chain can generate, multiple data types need to be supported. This way, every possible object can be stored in the blockchain.

Retrieve store data

All the data that is once stored in the chain needs to be used at a later point in time or direct. To use the data stored in the chain, a way of retrieving the data out of the chain needs to be possible.

Integrity

To ensure that the data can be trusted, no party should be able to change the data. The different values of the data need to be recorded and stored to guarantee the SLA. Other parties should not be able to change or fake the data in their favour, so they can blame the other party for not complying with SLA's agreed upon.

Validation of data

The data inserted in the blockchain should be validated. Validation is necessary to keep the data accurate and prevent inaccurate data from entering the system. The data added to the system need to correspond to a specific schema, so that the data can be validate based on requirements such as type, range of limit. A type of an attribute should not be able to change from an integer to a string. For example, if the value of a sensor is a measurement.

Permissions

Not all data in the blockchain should be shared with other parties. Because of this Hyperledger Fabric should possess the functionality to share data with other companies on a selective basis. This ensures that private company data stays safe.

Hyperledger Fabric should also possess the functionality to allow select organizations to the blockchain network. Other parties joining the blockchain network without permission could lead to a breach in confidentiality and safety. Only allowing selective organizations to the network would also ensure that the blockchain only contains relevant information.

Big data processing

A supply chain with IoT could generate loads of data. This means that Hyperledger Fabric should possess the functionality to process all this data. Dependent on the use case of the blockchain, this data also must be processed quickly and efficiently.

Smart contract

With all this data generated from the Supply chain with IoT, some kind of automation on the conditions needs to be in place. For this kind of automation, smart contracts are the way to go. Smart contracts hold the transaction dependencies and it validates the transaction before an acceptance or rejection is given. [45]

Conclusion

Hyperledger Fabric is suitable for use in a supply chain with IoT because it possesses all the needed functionalities as drawn up in sub-question 4.

Table 1-1 shows all the needed Hyperledger Fabric functionalities, and if Hyperledger Fabric possesses these features.

Tabel 1-1: Hyperledger Fabric needed/possessed functionalities

Needed functionalities Hyperledger Fabric	**Why is the functionality needed?**	**Does Hyperledger Fabric possess the functionality?**	**Explanation**
Support different datatypes	A supply chain with IoT could generate a lot of data with different data types, all these datatypes must be supported.	Yes	Hyperledger Fabric has support for the JSON data format, which supports all data types.
Retrieve stored data	To be able to use and view the blockchain data.	Yes	Hyperledger Fabric allows developers to query data based on keys by using the default database (LevelDB). It also offers supports for JSON based queries by allowing developers to switch to CouchDB (a JSON database) to support more complex queries.
Integrity	To ensure that the blockchain's data can be trusted and is not changeable by any unauthorized party.	Yes	Blockchain uses peer-to-peer consensus and hashes to ensure data in the blockchain is unchangeable.
Validation of data	To ensure that the data entering the blockchain is valid.	No	Hyperledger Fabric does not possess this feature.
Permissions	Private company data should not be shared with unauthorized parties. Only select organizations should be allowed to join the blockchain	Yes	Hyperledger Fabric provides users with a membership identity service. With this service, the user can manage all participants in a network. Hyperledger Fabric also offers access control lists to

	network to ensure confidentiality and safety. Also ensures that the blockchain only contains relevant information.		grant specific user IDs permissions. Private channels can be used in Hyperledger Fabric to share data with other companies on a selective basis.
Big data processing	A supply chain with IoT could generate a lot of data. All this data should be processable by the blockchain.	Yes	Hyperledger Fabric allows the network to be scaled and handle 20,000 transactions per second.
Smart contracts	To automatically check contracts between different companies, and ensure the SLA terms are met.	Yes	Chain codes define rules in Hyperledger Fabric. These chain codes can be applied on several levels. Endorsement and validation system chain code define the requirements for endorsing and validating transactions.

As shown in table 1, Hyperledger Fabric possesses almost all functionalities needed to be implemented in a supply chain with IoT. Only the feature ‘validation of data’ is missing, which means that this feature needs to be implemented using another method.

Discussion

This reach report is a validation of the research report written by the previous team from Windesheim. The previous team was researching on a lower level and went deeper into how frameworks such as Hyperledger work. In comparison, this research report was performed on a higher level and is more of a global overview. This research report tries to confirm and validate the previous conclusion without redoing the research.

In this paper, the team did not research how Hyperledger Fabric works in detail of how Hyperledger Fabric should specifically be implemented for this use case. That is because this research was already performed by the previous research team from Windesheim. To not redo this research, the team decided to validate if Hyperledger Fabric is a valid option to be implemented in a supply chain.

This paper aims to provide the team with valuable insights on what a (generic) supply chain is, what Hyperledger offers, the different ways it can be implemented and if it is even valid to use in a supply chain. This is also why the report does not exceed into an in-depth technical explanation of the given results.

An obstacle the team experienced while researching was that there is not much information available about the different cycles within a supply chain, specifically on the subject such as customer support. The data the team found is limited to a specific supply chain and is tough to compose as generic steps within a supply chain. A research team could perform a more detailed research paper on composing a generic supply chain, researching the subjects that are though and consume loads of time.

Bibliography

[1] Wikipedia, “Blockchain,” 7 3 2021. [Online]. Available: https://en.wikipedia.org/wiki/Blockchain. [Accessed 10 3 2021].

[2] C. Majaski, “Distributed Ledgers,” Investopedia, 12 04 2020. [Online]. Available: https://www.investopedia.com/terms/d/distributed-ledgers.asp#:~:text=A%20distributed%20ledger%20is%20a,geographies%2C%20accessible%20by%20multiple%20people.&text=Blockchain%20is%20a%20type%20of%20distributed%20ledger%20used%20by%20bitcoin.. [Accessed 10 03 2021].

[3] Oracle, “What is IoT,” Oracle, [Online]. Available: https://www.oracle.com/nl/internet-of-things/what-is-iot/. [Accessed 09 03 2021].

[4] TechMonitor, “What is a node?,” TechMonitor, [Online]. Available: https://techmonitor.ai/what-is/what-is-a-node-4927877. [Accessed 09 03 2021].

[5] M. W. Docks, “Working with JSON,” MDN Web Docks, [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON. [Accessed 09 03 2021].

[6] MuleSoft, “What is an API?,” MuleSoft, [Online]. Available: https://www.mulesoft.com/resources/api/what-is-an-api. [Accessed 09 03 2021].

[7] Wikipedia, “Radio-frequency identification,” Wikipedia, 09 03 2021. [Online]. Available: https://en.wikipedia.org/wiki/Radio-frequency_identification. [Accessed 09 03 2021].

[8] P. Juneja, “Desk Research - Methodology and Techniques,” managementstudyguide, [Online]. Available: https://www.managementstudyguide.com/desk-research.htm. [Accessed 23 2 2021].

[9] B. Lutkevich, “Supply chain,” TechTarget, October 2020. [Online]. Available: https://whatis.techtarget.com/definition/supply-chain. [Accessed 23 Februari 2021].

[10] Wikipedia, “Supply chain,” Wikipedia, 14 February 2021. [Online]. Available: https://en.wikipedia.org/wiki/Supply_chain#cite_note-Wieland-Wallenburg-3. [Accessed 26 February 2021].

[11] Chartered Institute of Procurement & Supply, “Oursourcing / Offshoring and Insourcing,” Chartered Institute of Procurement & Supply, [Online]. Available: https://www.cips.org/knowledge/procurement-topics-and-skills/understand-need---market-and-options-assessment/outsourcing--offshoring-and-insourcing/. [Accessed 26 February 2021].

[12] Wikipedia, “Raw materials,” Wikipedia, 27 Januari 2021. [Online]. Available: https://en.wikipedia.org/wiki/Raw_material. [Accessed 24 Februari 2021].

[13] Manufacturing & Technology Enterprise Center, “Supply chain - Explained,” Manufacturing & Technology Enterprise Center, 15 September 2017. [Online]. Available: https://mfgtec.org/supply-chain-explained/. [Accessed 25 Februari 2021].

[14] D. Schecter and G. Sander, “Delivering the Goods: The Art of Managing Your Supply Chain,” Inbound Logistics, 1 January 2003. [Online]. Available: https://www.inboundlogistics.com/cms/article/delivering-the-goods-the-art-of-managing-your-supply-chain/. [Accessed 28 February 2021].

[15] digiteum, “iot-development,” digiteum, [Online]. Available: https://www.digiteum.com/iot-development/. [Accessed 3 3 2021].

[16] S. Johnson, “IoT for Supply Chain Management: How to Use IoT for Enhanced Business Operations,” datasciencecentra, 12 november 2019. [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/iot-for-supply-chain-management-how-to-use-iot-for-enhanced. [Accessed 3 3 2021].

[17] Wikipedia, “Service-level agreement,” Wikipedia, 1 March 2021. [Online]. Available: https://en.wikipedia.org/wiki/Service-level_agreement. [Accessed 3 March 2021].

[1 K. R. S. F. a. M. S. Balbir Wadhwa, “Supply Chain Tracking and Traceability with IoT-Enabled Blockchain on AWS,” Amazon, 16 september 2020. [Online]. Available:

8] https://aws.amazon.com/blogs/apn/supply-chain-tracking-and-traceability-with-iot-enabled-blockchain-on-aws/. [Accessed 3 3 2021].

[19] D. Team, “Internet of Things for Smart Warehouses,” digiteum, 17 september 2019. [Online]. Available: https://www.digiteum.com/internet-of-things-for-smart-warehouses/. [Accessed 3 3 2021].

[20] Leverege, “supply-chain-management-iot,” Leverege, 4 june 2020. [Online]. Available: https://www.iotforall.com/supply-chain-management-iot. [Accessed 3 03 2021].

[21] P. University, “Electronic bacteria sensor is potential future tool for medicine and food safety,” sciencedaily, 14 juni 2016. [Online]. Available: https://www.sciencedaily.com/releases/2016/06/160614100347.htm. [Accessed 3 3 2021].

[22] Hyperledger, “Introduction,” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/blockchain.html. [Accessed 24 2 2021].

[23] Wikipedia, “Hyperledger,” 8 11 2020. [Online]. Available: https://en.wikipedia.org/wiki/Hyperledger#Hyperledger_Fabric. [Accessed 24 2 2021].

[24] Hyperledger, “Hyperledger Fabric - Hyperledger,” [Online]. Available: https://www.hyperledger.org/use/fabric. [Accessed 24 2 2021].

[25] Hyperledger, “Hyperledger Fabric Functionalities,” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/functionalities.html. [Accessed 24 2 2021].

[26] Hyperledger, “Introduction,” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html#modularity. [Accessed 10 03 2021].

[27] Hyperledger, “Hyperledger Fabric whitepaper,” 03 2020. [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2020/03/hyperledger_fabric_whitepaper.pdf. [Accessed 10 02 2021].

[28] zhangyaning, “Comparison between the UTXO and Account Model,” 20 oct 2018. [Online]. Available: https://medium.com/nervosnetwork/my-comparison-between-the-utxo-and-account-model-821eb46691b2. [Accessed 26 3 2021].

[29] Hyperledger, “Hyperledger Architecture, volume 1,” [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf. [Accessed 2021 3 3].

[30] Hyperledger, “fabric-chaincode-evm,” [Online]. Available: https://github.com/hyperledger/fabric-chaincode-evm. [Accessed 10 03 2021].

[31] Wikipedia, “Rolling release,” [Online]. Available: https://en.wikipedia.org/wiki/Rolling_release. [Accessed 10 03 2021].

[32] Hyperledger, “What’s new in Hyperledger Fabric v2.x,” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatsnew.html#decentralized-governance-for-smart-contracts. [Accessed 10 03 2021].

[33] Hyperledger, “Endorsement policies,” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/endorsement-policies.html. [Accessed 10 03 2021].

[34] hyperledger-fabric, “CouchDB as the State Database,” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/couchdb_as_state_database.html. [Accessed 26 3 2021].

[3 H. Fabric, “Getting Started — hyperledger-fabricdocs master documentation,” Hyperledger, [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/getting_started.html. [Accessed 01 03 2021].

5]

[36] H. Fabric, “Writing Your First Chaincode — hyperledger-fabricdocs master documentation,” Hyperledger, [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/chaincode4ade.html. [Accessed 01 03 2021].

[37] Github, “Hyperledger Fabric SDK for Node.js,” Github, [Online]. Available: https://hyperledger.github.io/fabric-sdk-node/. [Accessed 10 03 2021].

[38] H. Fabric, “Hyperledger Fabric SDKs — hyperledger-fabricdocs master documentation,” Hyperledger , [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/fabric-sdks.html. [Accessed 01 03 2021].

[39] Hyperledger, “Composer,” Hyperledger, [Online]. Available: https://www.hyperledger.org/use/composer#:~:text=Hyperledger%20Composer%20%E2%80%93%20Hyperledger&text=As%20of%20the%2029th%20August,project%20is%20in%20deprecated%20status.&text=It%20is%20highly%20recommended%20that,including%20a%20new%20programming%20mo. [Accessed 24 Februari 2021].

[40] Hyperledger, Director, *Build and Deploy a Blockchain Web App With Hyperledger Fabric & Hyperledger Composer.* [Film]. Hyperledger, 2018.

[41] IBM, “IBM Cloud Docs,” IBM, [Online]. Available: https://cloud.ibm.com/docs/blockchain?topic=blockchain-develop-vscode#develop-vscode. [Accessed 01 03 2021].

[42] Docker, “Docker overview,” Docker, 19 februari 2021. [Online]. Available: https://docs.docker.com/get-started/overview/. [Accessed 24 februari 2021].

[43] Kubernetes, “Production-Grade Container Orchestration,” Kubernetes, [Online]. Available: https://kubernetes.io/. [Accessed 02 03 2021].

[44] Docker, “Deploy on Kubernetes,” Docker, [Online]. Available: https://docs.docker.com/docker-for-windows/kubernetes/#:~:text=The%20Kubernetes%20server%20runs%20within,not%20affect%20your%20other%20workloads.. [Accessed 02 03 2021].

[45] HyperLedger, “Smart Contracts and Chaincode,” 2020. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.2/smartcontract/smartcontract.html#:~:text=Think%20of%20smart%20contracts%20as,defined%20within%20the%20same%20chaincode.. [Accessed 09 03 2021].

Appendix

This document contains no annexes.