



UNIVERSIDADE D
COIMBRA

Hugo Miguel Barros Abreu

**BIOGRAPHIES OF THINGS USING
BLOCKCHAIN**

A USE CASE FOR A SUSTAINABLE AND CIRCULAR
TEXTILE INDUSTRY

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Professor Vasco Nuno Sousa Simões Pereira and Professor João Nuno Lopes Barata and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July of 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Hugo Miguel Barros Abreu

Biographies of things using blockchain

A use case for a sustainable and circular textile
industry

Dissertation in the context of the Master in Informatics Engineering,
specialization in Software Engineering, advised by Prof. Vasco Nuno Sousa
Simões Pereira and Prof. João Nuno Lopes Barata and presented to the
Department of Informatics Engineering of the Faculty of Sciences and
Technology of the University of Coimbra.

July of 2023

Acknowledgements

I would like to express my deepest gratitude to Professor Vasco Nuno Sousa Simões Pereira and Professor João Nuno Lopes Barata for their support, guidance and concern throughout the development of this thesis.

I am also grateful to my parents for giving me the opportunity to pursue my studies, for always believing in me, and for giving me constant support.

I would also like to thank the paper pulp and fiber producer company for providing the necessary details to help develop this thesis.

Abstract

Nowadays, the textile industry faces several problems regarding environment pollution, scarcity of resources, and lack of transparency and traceability on their supply-chains. The creation of a digital product passport that supports the digitalization of each product biography, is a key enabler to the transition to a circular economy. Digital product passports carry product life cycle information making it available through data carriers. These passports shall contain information on the products entire life cycle to ensure that all actors can access data relevant to them, that verification of compliance with legal obligations is facilitated, and that traceability is improved throughout the supply chain. Blockchain, due to its many benefits such as immutability, traceability, transparency, and decentralization, is a suitable technology for the storage of these digital product passports.

The development of this thesis aims to create a blockchain-based prototype applied to digital product passports in a textile use case, and also provide a proof of concept of a digital product passport in the textile industry. The results of this thesis showed that blockchain has the necessary capabilities to support the storage of digital product passports. A fiber and paper pulp producer company supported the development of this thesis and gave positive feedback on the prototype. This use case is focused on a business area that throws away 85% of textiles each year, and that will be forced to fully apply digital product passports by 2030.

Keywords

Digital Product Passport, Blockchain, Circular Economy, Textile, Supply Chain

Resumo

Atualmente, a indústria têxtil enfrenta vários problemas relacionados com a poluição do meio ambiente, escassez de recursos e falta de transparência e rastreabilidade em suas cadeias de suprimentos. A criação de um passaporte digital de produtos que suporte a digitalização da biografia de cada produto, é um facilitador fundamental para a transição para uma Economia Circular. Os passaportes digitais de produtos carregam informações do ciclo de vida do produto, tornando-as disponíveis por meio de portadores de dados. Estes passaportes devem conter informações sobre todo o ciclo de vida dos produtos para garantir que todos os atores possam acessar os dados que lhes sejam relevantes, que a verificação do cumprimento das obrigações legais seja facilitada e que a rastreabilidade seja melhorada em toda a cadeia de suprimentos. Blockchain, devido aos seus muitos benefícios, como imutabilidade, rastreabilidade, transparência e descentralização, é uma tecnologia adequada para o armazenamento destes passaportes digitais de produtos.

O desenvolvimento desta tese visa criar um protótipo baseado em blockchain aplicado a passaportes digitais de produtos em um caso de uso têxtil, e também fornecer uma prova de conceito de um passaporte digital de produtos na indústria têxtil. Os resultados desta tese mostraram que a blockchain possui as capacidades necessárias para suportar o armazenamento de passaportes digitais de produtos. Uma empresa produtora de fibras e pasta de papel apoiou o desenvolvimento desta tese e deu feedback positivo sobre o protótipo. Este caso de uso é focado em uma área de negócios que descarta 85% dos têxteis a cada ano, e que será forçada a aplicar os passaportes digitais de produtos até 2030.

Palavras-Chave

Passaporte Digital de Produtos, Blockchain, Economia Circular, Têxtil, Cadeia de Suprimentos.

Contents

1	Introduction	1
2	Methodology and work plan	3
2.1	Methodology	3
2.2	Work plan	4
2.3	Risk management	6
3	Background	9
3.1	Circular economy	9
3.2	Digital product passports	10
3.3	Internet of things	11
3.4	Blockchain	13
3.5	Summary	16
4	Literature review	17
4.1	Digital product passport initiatives in the EU	17
4.2	Blockchain-based traceability	19
4.3	Blockchain-based digital product passports	21
4.4	Hyperledger	21
4.4.1	Sawtooth	22
4.4.2	Fabric	24
4.5	Summary	29
5	Use case and requirements definition	31
5.1	Use case	31
5.2	Requirements	33
5.3	Blockchain adoption in this use case	36
5.4	Summary	37
6	Architecture	39
6.1	Context layer	39
6.2	Container layer	40
6.3	Component layer	43
6.4	Summary	44
7	Development	47
7.1	Setting up the platform	47
7.1.1	Starting point	47
7.1.2	Prerequisites	48

7.1.3	Assembling the fabric network	49
7.2	Prototype development	50
7.2.1	Data structure	50
7.2.2	Permissions	51
7.2.3	Private data	53
7.2.4	Files	54
7.3	Prototype overview	56
7.3.1	Prototype user interface	56
7.3.2	Verification measures	64
7.3.3	Real use case demonstration	67
7.4	Summary	72
8	Evaluation and testing	75
8.1	Evaluation	75
8.2	Testing	78
8.3	Summary	83
9	Conclusions	85
	Appendix A Context diagram	95
	Appendix B Container diagram	97
	Appendix C Component diagram	99
	Appendix D Data structure	101

Acronyms

- CA** Certification Authority.
- CE** Circular Economy.
- CEAP** Circular Economy Action Plan.
- CLI** Command Line Interface.
- DLT** Distributed Ledger Technology.
- DPP** Digital Product Passport.
- DSR** Design Science Research.
- ECF** Elemental chlorine free.
- ESPR** Ecodesign for Sustainable Products Regulation.
- EU** European Union.
- FSC** Forest Stewardship Council.
- gRPC** Google Remote Procedure Call.
- IoT** Internet of Things.
- MSP** Membership Service Provider.
- NFC** Near Field Communication.
- P2P** Peer-to-peer.
- PBFT** Practical Byzantine Fault Tolerance.
- PoET** Proof of Elapsed Time.
- PoS** Proof of Stake.
- PoW** Proof of Work.
- RFID** Radio Frequency Identification.
- ToS** Threshold of Success.
- UI** User Interface.
- URL** Uniform Resource Locator.

List of Figures

2.1	Design Science Research (DSR) iteration process applied in this thesis, adapted from [Peffer et al., 2007]	4
2.2	Work plan for the first semester	5
2.3	Work plan for the second semester	6
3.1	The circular economy [Spring, Araujo, 2017]	10
3.2	Digital product passport use case [Ya, 2022]	11
3.3	Data structure of blocks [Zhu et al., 2021]	13
4.1	Sawtooth architecture [Sawtooth, 2022]	23
4.2	Example of a Hyperledger Fabric architecture [Phuwanai, 2019]	25
4.3	Sequence diagram of the example	28
5.1	Organizations and their interaction with the blockchain	32
6.1	Context diagram for a single organization	40
6.2	Container diagram - Organization only	41
6.3	Container diagram - Client Application only	42
6.5	Container diagram - Orderer Organization only	43
6.6	Component diagram - Peer only, adapted from [Wrisez, 2023]	44
6.4	Container diagram - Channel only	45
7.1	Collections configuration file	54
7.2	Dropbox storage example	54
7.3	Dropbox file example	55
7.4	Local storage example	55
7.5	PostgreSQL storage example	56
7.6	Login	57
7.7	Products page	57
7.8	Add new product	58
7.9	Add new product from others	59
7.10	Add new event	60
7.11	Add new transformation	60
7.12	Example of storage of private products	61
7.13	Add files	61
7.14	Example of the files	62
7.15	Product Fabric1 history	63
7.16	Example of the life cycle of products	63
7.17	Product Wood1 history	64
7.18	Monitor docker output	65

7.19	Smart contract permissions function	65
7.20	Reader policy permission denied	66
7.21	Writer policy permission denied	66
7.22	Front end code for permissions distinction	67
7.23	Front end code for documents permissions	67
7.24	Textile life cycle, adapted from [Eckhardt, 2011]	67
7.25	Created products	68
7.26	Eucalyptus with bark	69
7.27	Eucalyptus ECF	70
7.28	Lyocell fiber	71
7.29	Lyocell fabric	71
7.30	Lyocell sweater	72
8.1	Hierarchical tree example	75
8.2	Directory with the certificates and private keys	76
8.3	Docker compose CA configuration file	77
A.1	Context diagram of the architecture	96
B.1	Container diagram of the architecture	98
C.1	Component diagram of the architecture	100

List of Tables

2.1	Risk management	7
5.1	Blockchain constraint	35
5.2	Unauthorized data insertion	35
5.3	Unauthorized data access	36
5.4	Certificate and key management	36
7.1	Data insertion permissions	52
7.2	Read permissions	53
8.1	Test cases of data insertion related functional requirements	80
8.2	Test cases of data access related functional requirements	81
8.3	Test cases of non-functional requirements	83

Chapter 1

Introduction

Developed within the scope of the Masters in Informatics Engineering at the University of Coimbra, this document is the final report of the thesis that aims to create a blockchain-based prototype for storing Digital Product Passport (DPP)'s relating to the life cycle of textile products. The work is conducted in cooperation with a paper pulp and fiber producer company interested in adopting DPP's in their supply chain, and which will be referred to as "case company" in the remainder of the report.

Nowadays, most businesses apply the traditional linear economy [Stefan, 2022], which uses raw materials to create products and throws them away when they lose their value [Santander, 2021]. This strategy, despite being beneficial in earlier eras, is leading to scarcity of resources and increased pollution [eCycle, 2022]. From this need for change arose the Circular Economy (CE) that came to make the world more sustainable by reducing the negative impacts that the linear economy carries. Of the many benefits, the efficient use of resources and materials, and the reduction of pollution through recycling and reuse processes stands out [Stahel, 2016].

The DPP's appeared to improve the traceability of resources used in modern supply chains, providing more information to end consumers. This recent concept is part of the Ecodesign for Sustainable Products Regulation (ESPR) and Circular Economy Action Plan (CEAP), which aims to facilitate the European Union (EU) transition to a CE [Stefan, 2022], providing transparency along the entire supply-chain. From planting a tree to creating a piece of clothing, these passports are intended to store all the information of the product's life cycle to ease decision-making regarding the processes of reusing, repairing, refurbishing and recycling.

The textile sector prioritizes the CE and is currently studying how to implement DPP's in practice [Cura et al., 2022]. As identified in [ECOLabel, 2022], the textile sector encounters several problems, such as the use of fibers sourced from inefficient managed forests, leading to deforestation, and poor management of the paper cycle, affecting the environment. For each of these problems, proposed solutions are presented, but the processes and technologies that support these solutions are still in its infancy [Adisorn et al., 2021].

With an ever increasing demand in supply-chain businesses, blockchain arises as a technology that carries several advantageous characteristics for these businesses, such as: traceability, transparency, and security [Queiroz et al., 2020]. With the possibility of storing product information in a system shared by a group of identified entities, it becomes possible to create a trusted environment where all life cycle information is available at any time to all stakeholders. However, this does not mean that blockchain is the best solution for storing the life cycle of products, the intention is to take advantage of the characteristics it carries and test it in a use case within the textile sector.

This thesis is the follow-up of another one [Wrissez, 2023], where the main objective was to present a generic solution of a blockchain hosting product biographies. Therefore, the main objectives of this thesis are as follows:

- Provide a blockchain prototype, where it is possible to store information and navigate through the life cycle of textile products.
- Provide a proof of concept of a DPP in the textile sector.

The remainder of this document is structured as follows. In chapter 2 is presented the methodology adopted for this study, the work plan, and risk management. Chapter 3 introduces key concepts that are necessary to understand the rest of the report, where the following subjects are mentioned: CE, DPP's, Internet of Things (IoT), and blockchain. Next, in chapter 4, a literature review is carried out on the first initiatives on DPP's in the EU, product traceability solutions, blockchain-based DPP's, and two frameworks with the potential to host the presented solution. Subsequently, the use case and the requirements are defined in chapter 5, along with some notions on why blockchain is a suitable solution for this use. The design phase ends with the presentation of the prototype architecture in chapter 6. After everything is set, the steps taken during development are presented in chapter 7, followed by the respective prototype evaluation in chapter 8. Finally, the report ends with the main conclusions and future work opportunities.

Chapter 2

Methodology and work plan

After introducing the scope and the objectives of this thesis, and some concerns of the textile industry, this chapter starts by presenting the thesis methodology. Subsequently, a brief summary of the work carried out during the two semesters is presented, with the help of the respective Gantt diagrams. Finally, the risk analysis is included with the respective impacts and probabilities, and the mitigation plans.

2.1 Methodology

Considering the novelty of the topic addressed in this thesis and the opportunity to contribute to the body of knowledge in blockchain-based DPP's, the Design Science Research (DSR) [Peppers et al., 2007] methodology was selected. This methodology suggests a sequence of iterative steps [Peppers et al., 2007], which goes from identifying the problem and its motivation to presenting a conclusion. Despite being a sequence of steps, there is always the possibility of going back to rectify or improve previous steps.

Therefore, at the end of the first semester we found ourselves in the third step where an architectural proposal for the problem was presented. At the end of the second semester, we find ourselves in the final step where we reach the conclusions: the answers to our use case. It is important to mention that, during the development, it was necessary to change the proposed architecture, and during the prototype evaluation, it was necessary to go back to the development to correct the errors found.

Figure 2.1 shows the division of the DSR steps for each semester, and the identification of the steps where it is possible to go back.

Regarding the development, an agile methodology was adopted with continuous deliveries to the company that has followed the development of the thesis. These deliveries served to receive feedback, discuss problems, add new features to the prototype, and evaluate the use case results.

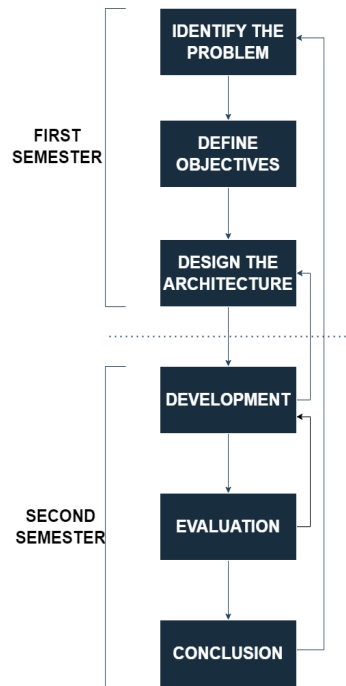


Figure 2.1: DSR iteration process applied in this thesis, adapted from [Peffer et al., 2007]

2.2 Work plan

As an initial part of the development of this thesis, a research was carried out on basic concepts necessary for understanding the main topic, such as: blockchain, IoT, DPP's and CE. Subsequently, for the literature review, the first initiatives of DPP's in the EU, solutions based on product traceability, and blockchain-based solutions for storing digital passports were studied. To finish the literature review, two frameworks provided by Hyperledger were analyzed to understand which would be the best choice to adopt for our solution.

Subsequently, the definition of the use case, and the functional and non-functional requirements was made with the help of the case company. Finally, the prototype architecture was made using the first three levels of abstraction of the C4 model [C4Model, 2022].

The preparation of the report was done simultaneously with the other tasks, excluding the first month where only research was carried out. All this information with the respective period of time in which they occurred is present in figure 2.2.

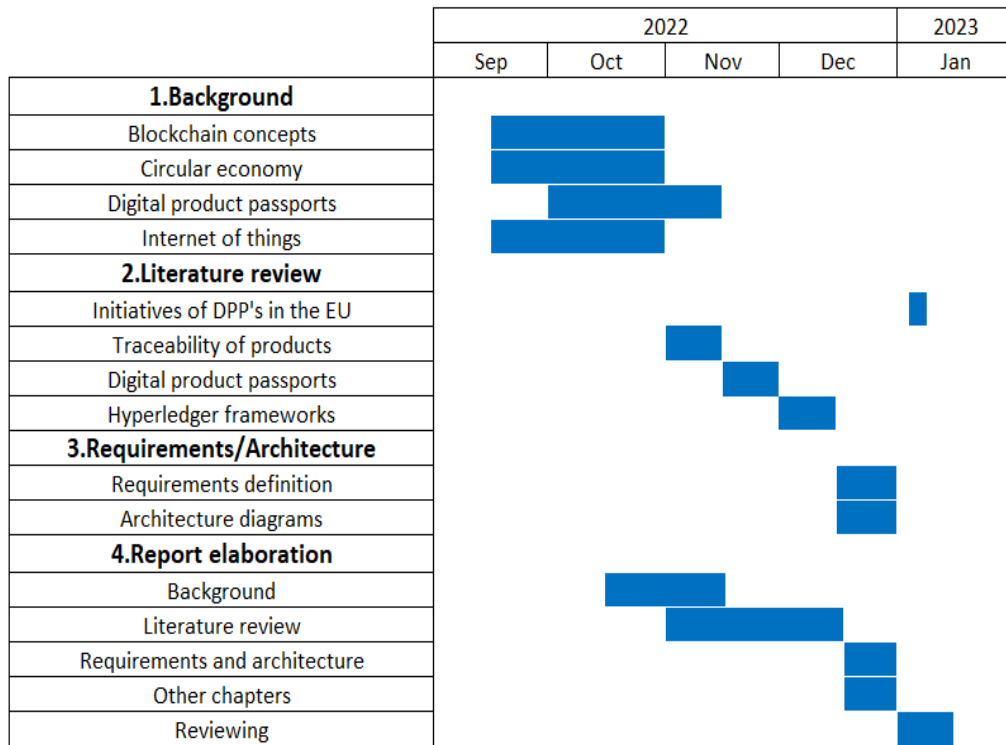


Figure 2.2: Work plan for the first semester

After the intermediate defense, the errors pointed out were rectified so as not to lose certain details that could be forgotten over time. Together with this step, the setup of the platform developed by [Wrisez, 2023] was carried out to get everything ready to start the development as soon as possible.

With the previous steps completed, the solution was developed, where it was necessary to adjust the architecture presented in the first semester, and the data structure defined by [Wrisez, 2023]. With the development completed, the tests to be carried out on the prototype were defined and subsequently performed.

Finally, similarly to the first semester, the preparation of the report was concurrent with all other tasks. Also, as in the first semester, around 10 to 15 days were left to review the entire report and make the necessary changes.

For a better perception of the time frames in which each task was carried out, this information is made available in figure 2.3.

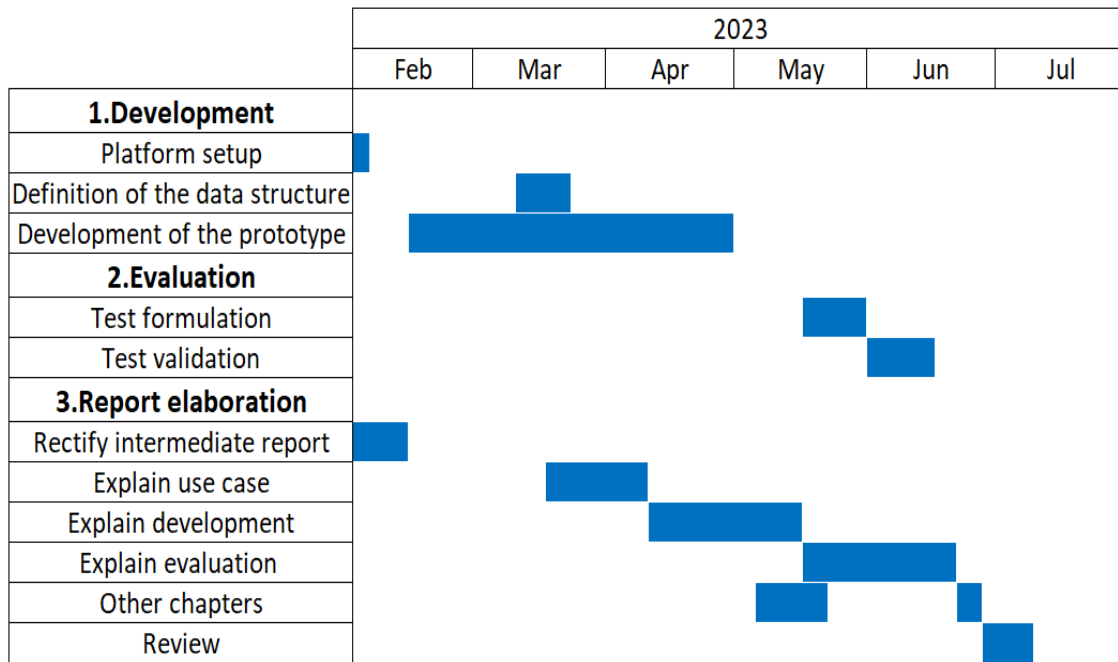


Figure 2.3: Work plan for the second semester

2.3 Risk management

As with any solution, there is always associated risks and it is necessary to take them into account to minimize their impacts or even avoid them. So, as to understand the impact and probability of each of the identified risks, it is necessary to define the Threshold of Success (ToS): the minimum requirements for the solution to be considered a success. In that regard, the key points for the success of the solution are identified below. Following, the levels for each factor will be identified as well.

Definition of the ToS:

- A prototype that allows the company to evaluate the use of blockchain for DPP's was presented.
- All defined requirements were developed and evaluated.
- A prototype of a DPP was presented.

Levels of impact:

- **Catastrophic** (Doesn't reach the ToS).
- **Critical** (Reaches the ToS with great effort/cost).
- **Marginal** (Reaches the ToS without great effort/cost).

Levels of probability:

- **High** (>70%).
- **Medium** (between 40% and 70%).
- **Low** (<40%).

With the levels identified, we then proceed to the definition of the risks, where the condition, respective consequence, impact and probability of occurrence, and mitigation plan are identified for each one.

Risk Management					
ID	Condition	Consequence	Impact	Probability	Mitigation Plan
R1	The case company takes a while to provide information to help defining the use case	Delays the definition of the requirements and development of the prototype	Critical	Low	Pressure the case company to provide the information, indicating that this lack could harm the final prototype or move forward with the definition of the use case based on the studies and research carried out
R2	The case company doesn't provide a lot of real data	Weakens the demonstration of a use case with real data	Marginal	High	Proceed with the demonstration based on the few data received

Table 2.1: Risk management

With the thesis methodology selected, the work plan presented, and the risks defined, the next chapter presents a background with key concepts for understanding the rest of the report.

Chapter 3

Background

Before moving on into a literature review where possible solutions for our use case are researched and analysed, several concepts will be introduced first to give higher quality knowledge to further link the concepts more easily.

Firstly, this chapter begins explaining CE and DPP's to figure out the aspects that make them so essential in the world nowadays, along with notions to correlate these concepts with a priority of this thesis: product traceability.

Secondly, an overview of IoT will be made to understand how this concept can help to enhance the process of gathering products information.

Thirdly, the concept of blockchain will be introduced to understand its structure and benefits. Aspects such as digital signatures, consensus protocols, hashing, data storage and merkle trees are part of this structure and will be addressed to understand how this concept fits into our use case.

3.1 Circular economy

Nowadays most supply-chains follow a take, make, dispose linear model [Spring, Araujo, 2017] that throws away the products when they lose their value. This model originates a lot of waste in a time where resources are becoming scarce. Aside from the waste that this model generates it is important to notice that the constant production of new products increases the levels of gas emissions, thus impacting the environment as well.

With the need to avail products to their full potential and to eliminate waste throughout the supply-chain a new model arose to revolutionize the world economy. The circular model, in opposition to the linear model, focuses on eliminating waste and pollution, circulate products and regenerate the nature [Ellen, 2022].

With this strategy, as seen in figure 3.1, at every step of the products life-cycle there is an opportunity to maintain the product or its materials in the economy by reusing, repairing, refurbishing and recycling instead of throwing them away.

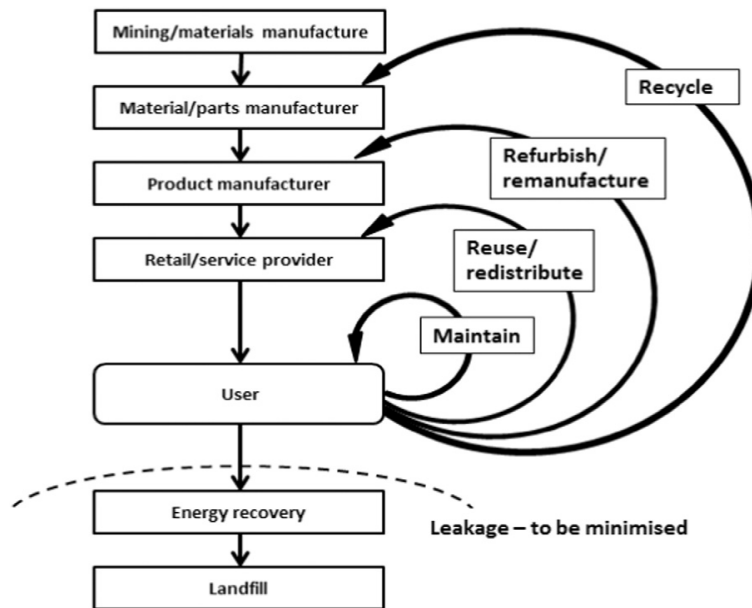


Figure 3.1: The circular economy [Spring, Araujo, 2017]

Although the idea of CE is fascinating, it comes with the need to add new processes of qualification to the businesses. This processes firstly figure out whether a product needs to be repaired or not and then determine which actions to take, from the ones seen in figure 3.1.

Transitioning from a linear economy to a CE will not be easy [McGinty, 2020], and as a way to ease businesses transition to this new economy model, the DPP concept emerged [Circularise, 2022], which is discussed in the next section.

3.2 Digital product passports

With the world now wanting to change into a more sustainable economy by adopting circular business models, the initiative of DPP has become more of a reality. Being part of the ESPR and CEAP, this initiative aims to collect information about the products and their supply-chains to make it available to all stakeholders [Circularise, 2022].

Therefore, the DPP's can help the sustainability of product production by improving material and energy efficiency, and extending the products life time. In addition, with access to the products data, more businesses can create value through circular business models and consumers can now make more informed purchasing decisions. Lastly, the data can be useful to verify the products compliance with legal obligations [GS1, 2022].

DPP's also offer the possibility of limiting access to the information they contain, so that certain actors can view all product information and others part of it [Nokelainen et al., 2023].

The information kept by the DPP's will be made available via data carriers such

as QR codes, Radio Frequency Identification (RFID) codes or Near Field Communication (NFC) accessed electronically. Since it is a new concept there's not much information related to the data requirements that this DPP's must comply with, but it is expected that by the end of 2024 the final ESPR working plan is published [Lewe, 2022], presenting the data requirements needed.

For now, the European Commission has released in 2022 a draft for a ESPR, where references to global and open standards are made a considerable amount of times. In this sense, the GS1 in Europe has been very proactive in this initiative to try and play a role to meet both industry and regulators needs [GS1, 2022]. The GS1 provides a system of standards for identification, capturing, and data exchange of products information to help on supply-chain businesses [GS1, 2023].

The EU has a priority to implement these passports in three sectors: electric & electronics, battery, and textile [DIGITALEUROPE, 2023]. In this regard, the EU funded a project with 30 partners that focuses on developing a roadmap for prototypes in these three sectors [CIRPASS, 2023], which is evolving in parallel with this thesis.

Image 3.2 illustrates an example use case of a DPP. As seen on the right side of the image, this passport could contain information such as the materials that make up the product, the place of origin of each one, the working conditions, etc. In short, these passports contain information on the entire life cycle of products, from the raw materials stage to the production and sale of clothing, in the case of the textile sector.

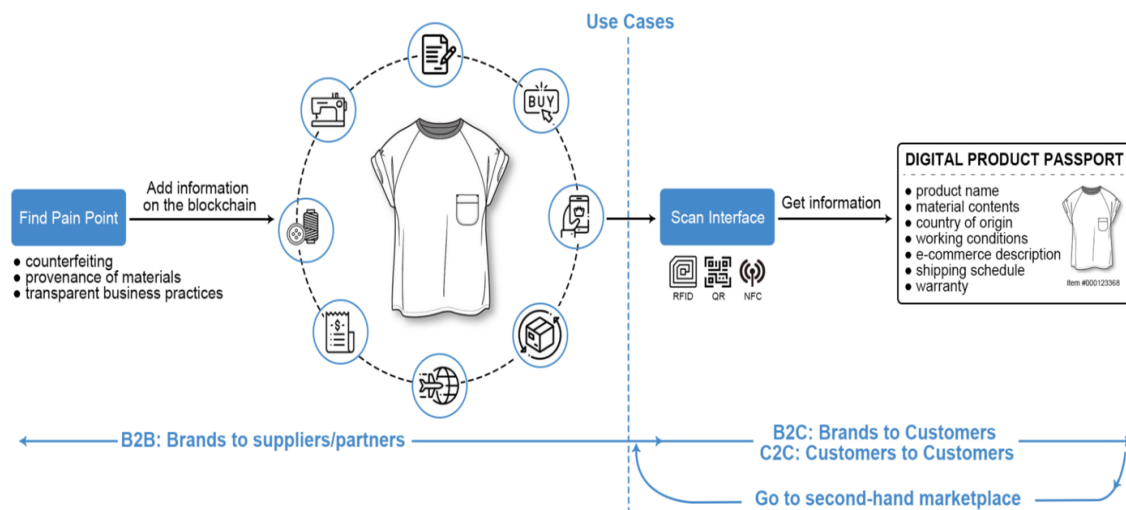


Figure 3.2: Digital product passport use case [Ya, 2022]

3.3 Internet of things

IoT is a concept that implies a set of objects, referred to as "things", connected in a network with little to none human-to-human or human-to-computer interactions. These devices being connected to a network are allowed, through sensors,

softwares or other technologies to collect and share data among themselves [PwC, 2022].

Nowadays it is very common to see this concept in our life, from smart homes to smart pacemakers. The possibilities of new businesses integrated with IoT are unlimited, and because of that, this concept just keeps on growing year after year [Bultin, 2022].

Since product traceability is one of the priorities of this thesis, information about the products life cycle must be collected and registered. However, sometimes it becomes impractical for this registration to be made with human intervention, specially when it is necessary for example, to register the temperature and humidity of the place where trees are planted. Therefore, it would be much easier to use a IoT device with temperature and humidity sensors that would register the data directly, in our use case, on the blockchain.

Obviously the security of this data has to be guaranteed, however, due to the characteristics of IoT, data security and privacy come as a big challenge [Wang et al., 2019]. Physical devices are connected to the internet, and therefore, data must be forwarded through communication channels for processing and storage. This results in a bigger threat landscape increasing the possibility of tampering. Normally, this devices are present in unattended areas such as the one given in the example above, where this devices are physically more vulnerable to a fair amount of attacks. This comes from the inability of shared communication channels to include more robust security systems.

From a variety of attacks possible to IoT networks, [Wang et al., 2019] identifies the following ones as the most typical:

- Attacks to end devices.
- Attacks to sensory data.
- Attacks to network protocols.
- Denial of service (DoS) attack.
- Software attacks.

Evidently, in a business where data must be kept safe and private, this type of attacks cannot happen. Therefore, various solutions come into hand to try and mitigate some of this issues. As expected, blockchains came as one of them due to its characteristics and architectural decisions [Šarac et al., 2021].

It is important to mention that IoT devices will not be used in this thesis, and are only explained to make it clear how later, data collection for the blockchain can be performed in a real context. Even so, as will be visible in the next chapters, the architecture adopted allows the addition of these devices to be performed in the future. Although this devices are not used, the problems they carry were taken account throughout the thesis, such as the attacks to sensory data, since "blockchain is believed to hold the key to settle security, data integrity and reliability concerns in IoT networks" [Wang et al., 2019].

3.4 Blockchain

First thought by Stuart Haber and Scott Stornetta in 1991, blockchain, a type of Distributed Ledger Technology (DLT), is a distributed and tamper-resistant database that provides decentralized data storage. It consists of blocks chained in serial in distributed Peer-to-peer (P2P) networks [Shrimali, Patel, 2022].

With this chained structure, any transaction that is added to the blockchain is joined as a new block at the end of the chain. After the addition of this new block, the data that it carries cannot longer be updated or deleted, thus ensuring the integrity and immutability of the data.

Every individual block contains data about transactions, the basic unit of records in blockchain [Wang et al., 2019], which is usually hashed and stored in a Merkle tree. Aside from the transactions records, each block has its own hash and the hash of the previous block, as well as other details seen in figure 3.3. Basically, when creating a new block, its hash is generated using the hash of the previous block. In this way, a chain of hashed blocks is created, making it practically impossible to change transactions.

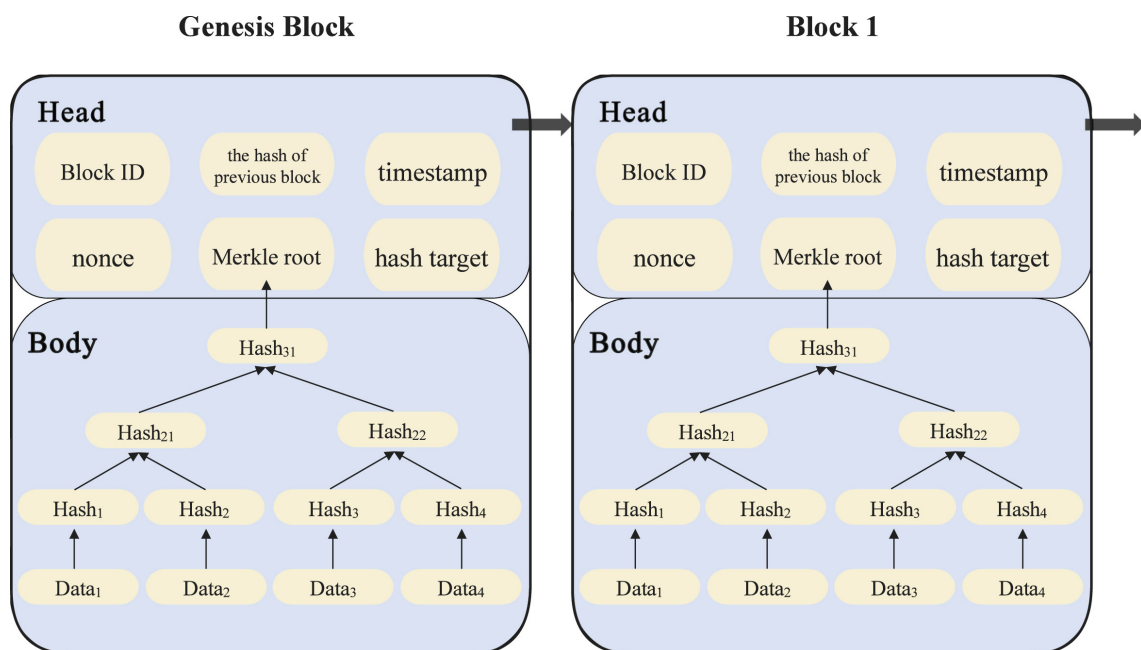


Figure 3.3: Data structure of blocks [Zhu et al., 2021]

Data storage

Blockchains store data in merkle trees to assure better performance. A merkle tree works from bottom to top and is nothing more than a hash tree that aims to provide efficiency and secure verification of the contents of the data structure. Every leaf node is the hash value of a transaction and every intermediate node is the hash value of its child nodes [Shrimali, Patel, 2022]. Reaching the top of the tree there is the merkle root which is stored in the block header.

The decision of using a merkle tree to storage data comes with the substantial

benefit of making the validation of the integrity of the data a lot easier. For example, if one of the leaf's hash is tampered, every node till the root will change as well. Therefore, it is only needed to check whether the two root nodes are the same or not. Another benefit is that the hashes make use of less storage space compared to raw data, consequently making the blockchain more efficient and scalable [Simplilearn, 2022].

Hashing

As mentioned before, blocks from the blockchain are hashed to ensure a tamper-resistant system. This is reached by using cryptographic hash functions, such as SHA-256. A hash is a "one-way" mathematical function that takes any input of data and converts it into a fixed-size length value.

Taking the same string and hashing it using SHA-256 will always result in the same output. Consequently, it is possible to compare the hashes to verify the integrity without requiring access of the raw data. In addition, from two different inputs it is highly improbable that the same hash is generated.

Digital signatures

Every single attempt to insert a new transaction into the blockchain, demands the authentication of the user that signed it. For this purpose, blockchain makes use of digital signatures, more specifically, asymmetric cryptography. This encryption method uses a pair of related keys, a public and a private key. The private key signs the transaction and the resultant of the encryption is attached to the transaction. Hereafter, any node in the network can use the public key related to the private key to verify that the transaction is genuine.

Consensus protocols

Since blockchain has a lot of peers and transactions happening all the time, there must be a way to decide which blocks are added to the chain to prevent any malicious attacks. In this way, blockchain requires that the peers that mine the blocks follow a consensus protocol. This protocol consists of having a specific method of confirmation of what data should be added to the blockchain.

Evidently, each consensus protocol has its own advantages and disadvantages, and therefore, it must be chosen carefully to fulfill the business needs. Within several options, Proof of Work (PoW) and Proof of Stake (PoS) come as the most known as a result of their usage in Bitcoin and Ethereum.

- **PoW** is a protocol that generates highly complex cryptographic puzzles that need to be solved by the nodes in the network to be able to broadcast and create new blocks. This puzzles require computational power to be solved, and therefore, the nodes compete to be the first ones to broadcast the new blocks to the chain [Wahab, Memood, 2022]. As expected, the nodes with more computational power have the highest odds of successfully creating the blocks first. As more blocks are added to the chain, this puzzles keep on increasing complexity, thus needing more CPU power to solve them.
- **PoS** is a protocol where nodes, referred to as validators, stake capital in the

form of cryptocurrencies, for example, ether's in the case of Ethereum. By doing this, if a node acts dishonest towards the network this capital can be destroyed [Ethereum, 2022], acting as a form of punishment. Additionally, nodes with more staked capital and for a longer period of time act as a more trustworthy validator.

Smart Contract

Being the innovation that Ethereum brought to the cryptocurrency world, smart contracts are programs, normally following a "if/when...then..." structure, that are written in code and stored in the blockchain [IBM, 2022]. These contracts are executed automatically and immediately when certain conditions are met, and therefore, efficiency is guaranteed. Important to mention that the smart contract concept already existed, previously to their adoption in Ethereum [Szabo, 1997].

Types of blockchain

Despite the same general architecture explained above, blockchains can differ from each other, specially when it comes to permissions. Depending on the needs of each business, four different types of blockchain can be applied.

Public blockchain

Being the most common, due to their application in cryptocurrencies, public blockchains don't have restrictions or permissions. Therefore, anyone with internet access can join these blockchains and become an authorized node to send transactions and mine blocks [Kathleen, Eugenia, 2021].

As a result, with the high number of nodes in the network a better decentralization is guaranteed. Additionally, with the existence of more copies of the ledger and with no authority controlling the blockchain, a more trustable and secure environment is delivered, being almost impossible to occur data breaches or data tampering.

On the other hand, since anyone can join the blockchain, the tendency is that the blockchain becomes slower and slower as more nodes access the network. This happens due to the lack of scalability that this types of blockchains hold.

Private blockchain

Unlike public blockchains, private blockchains, that can also be referred to as permissioned blockchains, have a central authority that controls not only the nodes that can access the network but also which type of access rights these nodes can have [Shrimali, Patel, 2022].

Thus, being a more closed network and smaller in size, these blockchains are faster, scalable and more efficient.

Evidently, since there's less nodes compared to a public blockchain, these blockchains are more vulnerable to security breaches such as data tampering [Investopedia, 2022]. Not to mention that these blockchains are centralized, and therefore, less trustworthy.

It is important to note that these disadvantages are compared to public blockchains, which does not mean that these blockchains are not reliable or easy to tamper with.

Hybrid blockchain

Coming as the third type, hybrid blockchains are a mix of both private and public blockchain solutions. These blockchains try to combine the best of both worlds by creating a private permissioned system alongside a permissionless system. Transactions are stored safely in a private network but can still be verified as necessary, by granting access through smart contracts [Zeeve, 2022].

Resultant of the chosen architecture, hybrid blockchains protect from 51% attacks because hackers can't access the network due to the access rights [Zeeve, 2022]. Also, the infrastructure of the blockchain is flexible and, as a result, the organization can change the rules based on their needs.

Consortium blockchain

Lastly, there's consortium blockchains, that are a combination of multiple private blockchains belonging to different organizations [Bybit, 2022]. Each organization has a node in each of the blockchains and in this way data can be accessed and shared within the consortium.

With this structure, data is kept private within the organizations and is tamper-resistant. As a downside, they inherit the vulnerabilities of the private blockchains, i.e., vulnerability to malicious participants due to the network centralization.

Therefore, to provide an example of each one, a private and consortium blockchain would be Hyperledger [Hyperledger, 2022a], a public blockchain, Bitcoin [Nakamoto, 2018], and a hybrid blockchain, IBM food trust [IBM, 2023].

3.5 Summary

This chapter covered several important concepts for understanding the use case and the developed prototype. Initially, the circular model was presented, sharing some problems found in today's supply chains such as waste and pollution, as well as the advantages of this new model. Then, the new concept of DPP was introduced, presenting the benefits that these would bring to modern supply chains.

Subsequently, the concept of IoT was presented, announcing the problems that IoT networks encounter, and how these devices can help in capturing information on the life cycle of products. Finally, blockchain technology was explained, detailing its structure, and making known the different types of blockchain that exist, since it is in this technology that our use case is supported.

Chapter 4

Literature review

After introducing essential concepts for perception and linking knowledge, in this chapter, a literature review will be carried out to study and analyze solutions to help in the subsequent creation of our own solution.

To start the literature review, the first initiatives in the EU of the DPP's will be studied and analyzed to understand which regulations, standards, and requirements they must follow for later adoption in our solution.

Following, solutions about product traceability are studied in order to understand in which steps the supply-chain is divided for later storage in the blockchain. The notion of IoT already mentioned in section 3.3 is also referenced in some of the solutions and will aid to link how this concept helps in gathering information about products along the supply-chain.

After analyzing blockchain-based traceability solutions, we will proceed to the study of solutions that use the concept of DPP's together with DLT's, to understand the status of the solutions developed, and if they were tested in real contexts.

Subsequently, the literature review studies two frameworks. Both provided by Hyperledger, and which are mainly focused on private and permissioned blockchains aimed at companies in various sectors, with the textile sector being one of them. Finally, the literature review ends with a brief summary.

4.1 Digital product passport initiatives in the EU

Despite being a new concept, DPP's are already a much talked about subject in all industries due to the benefits they provide in the transition to a CE. In this sense, several initiatives have already been carried out in different sectors, some still in the development and testing phase, and others already in practice [Jansen et al., 2022].

Therefore, in this section, we will address the initiatives related to the textile sector to collect information on data requirements, standards, and regulations, fol-

lowed by some initiatives of other sectors to understand the status of these DPP's in the EU.

[Haegglom, Palmer, 2019] released the circularity.ID Open Data Standard that focuses on the fashion industry, regarding labelling, identification and storage of DPP's. This standard defines the information that must be stored regarding the life cycle of products, and is divided into an XML component that aims to store immutable data, such as the materials that make up the product, and a set of mutable data, such as product descriptions or images.

Another initiative taken within the textile industry was presented by [TrueTwins, 2022]. Developed to comply with the EU Draft Sustainable Product Regulation [Commission, 2022b], this solution offers a plug and play interface accessed through NFC tags, bar codes or QR codes. Also, in compliance with data protection legislation, TrueTwins provides a platform for voluntary sharing of data related to the product.

[Nokelainen et al., 2023] presented a solution to identify how it would be to develop this concept of DPP from scratch, taking into account various aspects such as regulations, points of view of various stakeholders, and technical feasibility. Thus, the objective was to create a generally applicable concept of these passports, which they did in collaboration with actors from the textile industry and the battery value chain. In this solution they identify various regulations regarding the DPP's, and the textile products, for example, the EU Strategy for Sustainable and Circular Textiles [Commission, 2022a]. Therefore, the textile sector still has several ongoing projects, aimed at proposing a definition of the DPP [Trace4Value, 2023].

Regarding the remaining sectors, many of them already have initiatives put into practice [Jansen et al., 2022], such as: buildings, automobiles, valves manifolds, batteries, industrial production, bikes, and food. However, they do not reveal information that can be used for the textile sector, and therefore, only serve to show the status of passports in the EU. An example of a recent call for proposals to implement DPP's in two value chains is the [Commission, 2023], revealing the importance of this topic to the EU and the initial stages of implementation, requiring additional research.

Although the final regulation with the requirements for sustainable products has not yet been released, it is already possible to find several solutions on the market with some of them following the ESPR draft released in 2022. This is due to the fact that the final ESPR working plan will be released in 2024 [Lewe, 2022], and an early adoption of the ESPR draft will ease the businesses transition into the final one. This regulation intends to provide a framework that will demand ecodesign requirements on products intended for the EU market [Chopova-Leprêtre et al., 2023].

4.2 Blockchain-based traceability

Product traceability is a crucial aspect to achieve sustainable and circular supply chains in the textile industry. The possibility of having access to a complete history of the entire lifespan of a product, known as a product biography [Barata et al., 2020], is indeed a huge help to achieve sustainability in any sector. Since this history is accessible at any time, it makes it possible to understand when a product needs to be recycled, or reused, like any other possibility already mentioned in section 3.1 about the CE. With the possibility of tracking products along the supply-chain, a DPP, referred to in section 3.2 would be more easily implemented.

Now with an idea of what product traceability is, this section will address three solution proposals that used this concept of product biography in order to understand the decisions taken and technologies used. Understanding notions such as the division of supply-chain phases, as well as the way data is stored will be important for later application to our case study.

With the need to obtain and store information about the life cycle of products, some issues arise that cannot be ignored. How this information gathering is performed depends on many factors and is not always easy to accomplish due to many limitations. As in the use case analyzed by [Barata et al., 2020] where digital twins (replicas of physical products) are used, obtaining and tracking this data was extremely difficult due to the impossibility of having only one digital twin per product.

To solve this problem, two possible strategies were identified. The first one was based on having a digital twin for each phase of major transformation, where three major phases were identified: (1) Smart materials, (2) Smart factory, and (3) Smart product.

The second solution was to have a digital twin for each group of similar products, for example, trees sown in the same forest under the same atmospheric conditions would be represented by the same digital twin.

[Barata et al., 2020] also presented a prototype of an IoT device that would collect information regarding the smart products phase, where humidity and temperature sensors were visible. This was due to complaints that the products were having due to water retention. However, this prototype still had to be tested. An interesting solution given that for other sectors and businesses, more types of sensors could be applied to detect a variety of relevant information.

[Barata et al., 2020] gave us strategies on how to be able to store data related to the product along the supply-chain, but it was only mentioned that the system would use on-chain and off-chain data storage without specifying any type of technology. Therefore, further research on this topic was needed and [Yang et al., 2021] came to help by proposing a solution for a traceability system regarding fruit and vegetable agricultural products using the HyperLedger framework, explained in section 4.4. The objective of this solution was to solve problems such as heavy load, slow query speed, and private data protection, that are also relevant to our

use case.

The process of tracking the products was divided into 4 different phases, (1) production, (2) processing, (3) logistics and (4) sales, but there is not much information about how this tracking was done. Therefore, the focus is on how this data was stored, since [Yang et al., 2021] presented a solution where two types of storage were used. The first one was on-chain storage, that is, the blockchain and the second one was off-chain storage, the database. The important thing to take away from this structure is the way in which the data were organized and placed in each one to guarantee the fulfillment of the aforementioned objectives.

In this sense, the public information about the products was stored in the database, while the encryption of the private information, as well as, the hash value of the public information was stored in the blockchain. In this way, data query speed, as well as the usage of less storage space on the blockchain is guaranteed. As far as security against data tampering, this is guaranteed due to the possibility of comparing the hash value present in the blockchain with the information present in the database.

Moving on to the third solution proposal presented by [Agrawal et al., 2021], and the most important one due to its focus on the textile industry, the interest was on understanding the different phases chosen for the supply-chain, and on the type of blockchain and resultant performance. [Agrawal et al., 2021] divided the supply-chain into six phases using the same reasoning as [Barata et al., 2020], dividing it according to major transformations.

Additionally, with different partners, such as organic cotton suppliers and fabric manufacturers related to their cotton use case, [Agrawal et al., 2021] selected a private and permissioned blockchain (HyperLedger Fabric), where each partner has its own permissions. For example, the organic cotton supplier is the only partner who can add cotton mass in the supply-chain and corresponding transactions on the blockchain.

For testing purposes, [Agrawal et al., 2021] defined the consensus protocol to be PoW and deployed a smart contract to verify the public-private key signature and the account balance of the sender. After several tests varying the difficulty of the PoW protocol, and the number of transactions sent to one block, it came to conclusion that the number of transactions didn't affect significantly the time for hashing a block but, on the other hand, the difficulty of the PoW protocol affected this time exponentially.

Blockchain-based traceability is a popular research area, including contributions for the textile industry [Hader et al., 2022], but a lot of this literature is still very conceptual [Sunny et al., 2020], "and there is a clear need for developing and testing real-life traceability solutions, especially taking into account feasibility and cost-related SC aspects" [Dasaklis et al., 2022].

4.3 Blockchain-based digital product passports

As DPP's carry the entire life information of a product, it is essential to ensure that this data is never tampered with or lost. If there is any type of contamination derived from a product, it must be possible to trace its entire life cycle to identify at what stage such contamination may have occurred, and to identify other products that may have been contaminated as well. Thus, it is very important that the information present in these DPP's is correct. For this reason, blockchain emerges as a potential solution due to its characteristics, such as: immutability and resistance to tampering [Armin, 2021].

As mentioned in section 3.2, DPP is a fairly recent concept that is still being debated within the EU, i.e., it has not been implemented yet. Therefore, solutions of these DPP's integrated with DLT's are still hard to find, specially in the textile sector. However, and reminding that DPP's are expected to be a obligation for all sectors by 2030 [GS1, 2022], a few solutions have already been developed and tested, while others are still on the design phase.

LyondellBasell developed a prototype of a DPP integrated with blockchain technology with the help of Circularise's traceability software [LyondellBasell, 2023]. After the development of the prototype, which has a focus on industrial supply chains, Circularise partnered with Porsche to help them achieve better traceability and transparency. As a result, the prototype helped Porsche achieve its sustainability goals, and compliance with upcoming regulations [Circularise, 2023].

Another solution was the one developed by Digimarc and IOTA for the European Commission [Guinard, 2023]. These organizations cooperated to create a blueprint of a DPP integrated with blockchain technologies to support some of the EU initiative requirements for the DPP's. With the blueprint developed, Digimarc and IOTA tested the solution in a use case for electric vehicle batteries.

[Billon, 2023] also developed a blockchain-based DPP solution, but it lacks sufficient documentation and tests in practice to demonstrate the advantage of using blockchain in the storage of DPP's. Apart from the first two solutions, which were tested, the industry still needs more real use cases to understand the necessity to use blockchain technology to move towards a CE [Brown, 2022], specially the textile sector.

Still, many other organizations have already presented DPP architectures for integration with blockchain technologies [Marchesi et al., 2022], but they are merely conceptual and do not deliver results in a real context.

4.4 Hyperledger

After reviewing and analyzing both product traceability solutions and digital passports with DLT's integration, let's move on to the analysis of frameworks that can help build our blockchain network for storing DPP's.

Hyperledger, part of the Linux Foundation, was launched in 2016 by a consortium of different enterprises that found out that working together was more beneficial than working alone [Hyperledger, 2022c]. The goal is to have an open source community focused on developing different frameworks, tools and libraries for enterprise-grade blockchain deployments spread across different sectors [Hyperledger, 2022c].

For the subsequent creation of our architecture, the aim will be only on the Sawtooth and Fabric frameworks that Hyperledger offers since they focus on supply-chain solutions [Hyperledger, 2022c]. It is also important to mention that the focus is given to these two frameworks due to the fact that this thesis is the continuation of another one [Wrisez, 2023], already mentioned in section 1.

4.4.1 Sawtooth

Sawtooth is a framework for building distributed ledger applications and networks, provided by Hyperledger, focused on security, scalability and modularity [Olson et al., 2018]. It was developed by the Linux Foundation in collaboration with enterprises such as Intel, IBM and SAP.

Architecture

On the Sawtooth platform, as on any other blockchain, it is necessary to validate transactions and, for this purpose, nodes called validators are used. These nodes receive a batch, which is nothing less than a set of transactions, instead of just one at a time. When validating each of these transactions, it is inherent that the invalidation of one of them causes all transactions present in that batch to be invalidated.

With regard to transaction scheduling, Sawtooth supports serial scheduling, i.e., the usual, and parallel scheduling, which is used whenever possible. By doing this scheduling in parallel a better performance is achieved.

For a connection between the user and the network, Sawtooth provides a REST API, as shown in figure 4.1, that allows users to interact with the validator through HTTPS/JSON requests.

Another feature of Sawtooth is the possibility, within a permissioned network, to have different roles for each node. These roles control the type of messages these nodes can send and receive on a given connection. For nodes that intend to have roles within the network, an authorization "handshake" must be done with the subsequent request for the type of role they want to represent.

Smart contracts in this framework, are referred to as transaction processors and, when a transaction passes through the distribution log, it is forwarded to the appropriate transaction processor. In addition to these transaction processors that can be imposed by validators both in relationships between transactions and in inter-transaction validation, Sawtooth offers a group of predefined smart contracts called transaction families. Additionally, it is also possible to have on-chain validation rules, that is, rules that are already defined in the blockchain.

Consensus protocols

One of the functionalities of Hyperledger Sawtooth is the offer of a consensus interface where developers have the possibility of choosing the desired protocol. In this interface, four consensus protocols are available, one of which is still under development:

- **Practical Byzantine Fault Tolerance (PBFT)** is an algorithm that aims to guarantee byzantine fault tolerance. In this framework it is extended from the original algorithm as it offers features such as dynamic network membership, regular view changes and a block catch-up procedure. For the implementation of this algorithm, the network must have four or more nodes [Sawtooth, 2022].
- **Proof of Elapsed Time (PoET)** is a protocol very similar to PoW, but consumes much less energy since it is not necessary to solve the cryptographic puzzles. Instead, each validator is given a random waiting time that it must wait until it can mine the next block. The first validator to "wake up" from this waiting time then gets the chance to generate the block. Basically it is a fair lottery-style consensus protocol where each node has a certain probability of mining the next block [Wahab, Memood, 2022].
- **Dev_mode** is a protocol for testing purposes that chooses a random leader to validate the transactions.
- **Raft** is a leader-based protocol that guarantees tolerance against crash faults. However, it only guarantees this for small networks with restricted members. Being the one that is still under development it cannot be implemented for the time being [Sawtooth, 2022].

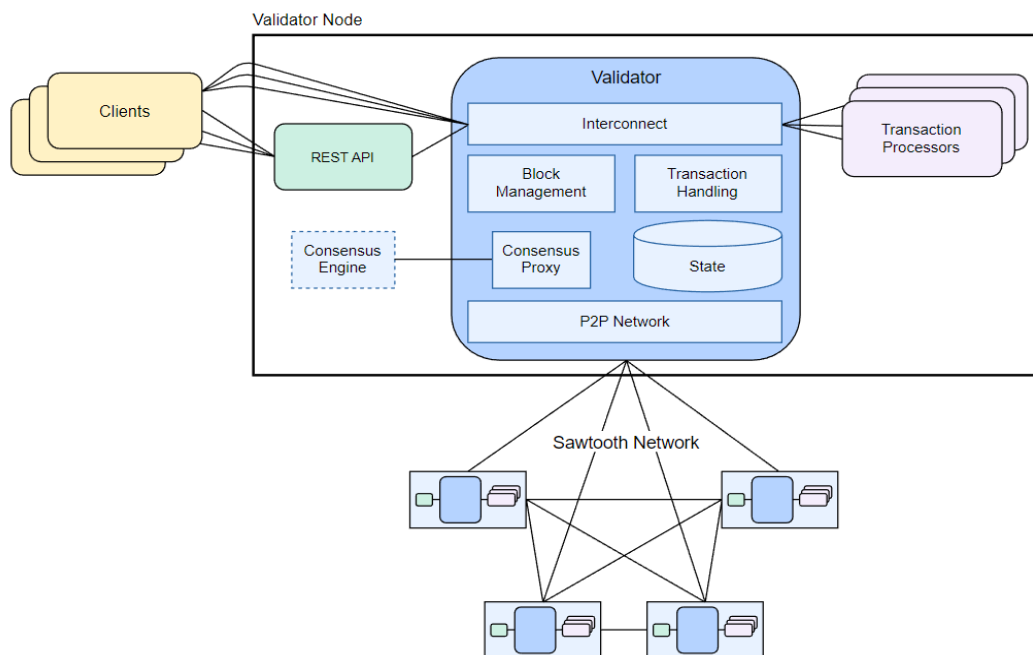


Figure 4.1: Sawtooth architecture [Sawtooth, 2022]

Due to its scalability, as well as being IoT-driven, Sawtooth has become the platform of choice for Scanztrust [Morris, 2018], a company that provides IoT services to its customers to identify their products on the internet, ensuring their traceability [Hyperledger, 2022c]. This choice was made to improve its offer in supply-chain traceability and help its customer, Cambio Coffee.

4.4.2 Fabric

Fabric is another framework provided by Hyperledger that focuses on confidentiality, flexibility, resiliency and scalability [Hyperledger, 2022c]. Like Sawtooth, it is a private, permissioned blockchain with a modular architecture where consensus protocols and smart contracts can be plugged in.

Architecture

In this framework, three types of nodes are found: (1) Client, which sends transaction requests to endorsers, (2) Peer, which commits transactions and maintains a copy of the ledger, and (3) Orderer, which orders incoming transactions [Androulaki et al., 2018].

To increase transaction privacy, Fabric has a very interesting feature called channels. Within the same network it is possible to create several channels, each with its own ledger where only authorized organizations can act. In this way, two participants can create a private channel where they keep their ledger, thus guaranteeing the privacy of their transactions [Hyperledger, 2022a]. Note that each channel has its own configuration, which is independent of other channels and the Fabric network.

Regarding the smart contracts, in Hyperledger Fabric, they are called chaincode and can be implemented in several languages, like GO and Node. These chaincodes can be installed directly on the peers as these are the ones that normally carry out the transactions. On the other hand, despite being present in all channels, orderers do not need to install the chaincodes as they do not commit transactions.

In order to be able to act on the network and respective channels, all nodes must carry an identity. In this sense, a Certification Authority (CA) has to issue an X.509 certificate which includes the digital identity of the node. Along with this certificate, each node has its own private and public key. The private key is used to sign transactions, and the public key to verify the integrity of the transactions.

When a node intends to carry out a transaction signed with its private key, it must be validated and this is where the Membership Service Provider (MSP) appears. These entities check the permissions of the node to conclude if the transaction can be done or not. Briefly, the MSP's indirectly give roles to the nodes and identify the organizations present in the channels [Hyperledger, 2022a]. Furthermore, they have the ability to check the list of identities that have already been revoked.

Thus, two types of MSP's are identified based on their scope [Hyperledger, 2022a]:

- **Local MSP**, exists locally on an actor's node. Every node must have its own local MSP.
- **Channel MSP**, exists in the channel configuration. Contains the information of all the local MSP's.

Image 4.2 provides an example of a possible Hyperledger Fabric architecture. Three organizations are present, organization 1 and 2 are associated with channel A and organization 2 and 3 are associated with channel B. It should be noted that organization 2, as it is linked to both channels, has a copy of the ledger of each of them. Finally, there is the orderer, which operates in all channels to order transactions.

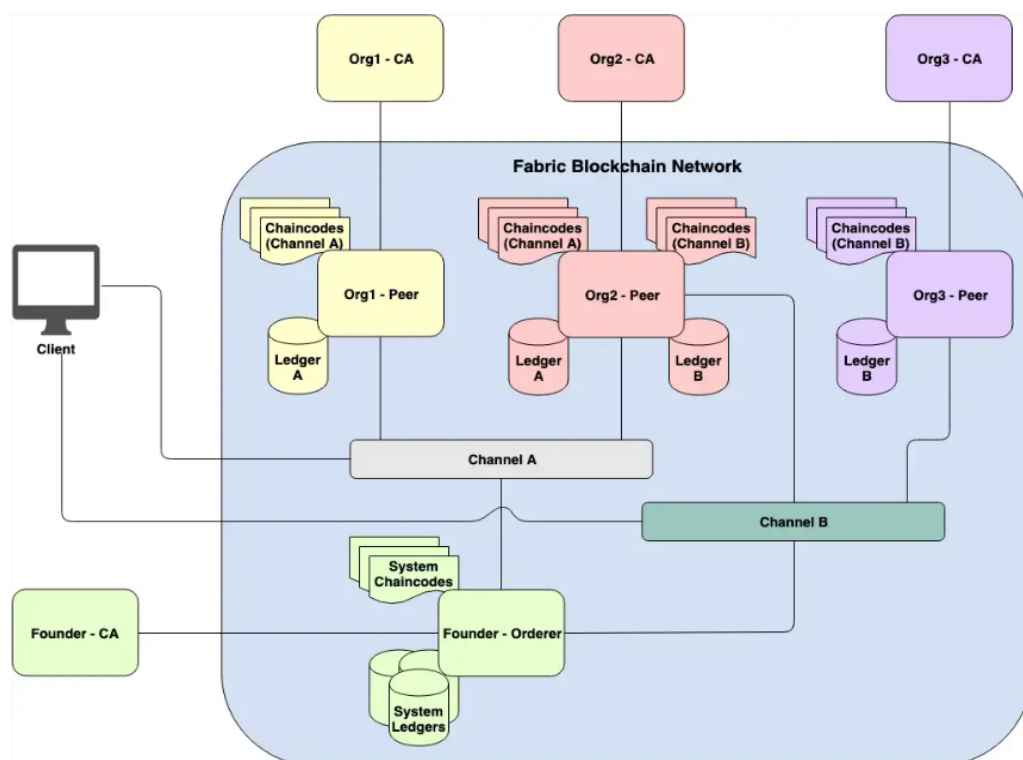


Figure 4.2: Example of a Hyperledger Fabric architecture [Phuwanai, 2019]

Policies

Fabric contains rules for any action that is performed on the network, be it the addition of new organizations, or the endorsement of a smart contract. These rules are stored in what are called policies, and these define, for example, how many peers must endorse a transaction for it to be validated or how many organization admins must accept the addition of a new organization. Basically, they define the structure of how certain decisions must be made and certain outcomes achieved.

These policies are defined when the channel is created and are approved by the organizations that will be present in it. This does not mean that these policies cannot be changed after the creation of the channel, it is only necessary that the changes are validated by everyone.

Policies can be defined within three groups, (1) Application group, everything related to organizations, e.g., adding new organizations to the network, (2) Orderer group, related to orderers, such as adding new orderers to the network, and (3) Channel group, relating to everything that crosses organizations and orderers scopes, such as the endorsement of a transaction.

In this regard, there are two types of policies that can be implemented:

- **Signature policies**, these are more flexible as they allow the definition of more specific rules, e.g., a peer from organization 1 and a peer from organization 2 must endorse the transaction, or an admin from organization 1 and two from organization 2 must validate the addition of a new organization.
- **ImplicitMeta policies**, these are only valid in the context of channel configuration and are based on a hierarchical tree of policies that will eventually be defined by signature policies. In this case, for these policies to be valid, all sub-policies must be valid too. For example, an ImplicitMeta policy could be "Majority of admins", meaning that the majority of admins must accept the addition of a new organization to the channel. In a lower level, in the Application domain there would be an admin signature policy such as, "OR(Org1MSP.admin)" that would need to be validated for the ImplicitMeta to be validated as well.

Identity

As already mentioned, each node present in the network needs something to identify itself, since Fabric is a private blockchain. Therefore, each node must have a digital identification, in this case, an X.509 certificate accompanied by a public and private key. This certificate is signed by the certificate of the organization CA, which has its own certificate signed by the Fabric CA, for example. In this way, a chain of trustworthy certificates is created.

In Hyperledger Fabric, although its use is not mandatory, the Fabric CA is available for managing and issuing certificates. If a certificate or private key is leaked, there is always the possibility to generate new certificates and keys for the organization in question. It should be noted that for the generation of this new data, there is no need to restart the blockchain, nor does it affect other organizations.

Security model

Given that the solution resulting from this thesis will use a client application to interact with the blockchain, it makes perfect sense to explain the process of inserting new transactions. In this respect, below is an ordered list of the steps that lead to the addition of new blocks to the ledger:

1. A user makes a request to add a new product to the ledger.
2. Through the gateway service, a transaction request is sent with the endpoint of the desired peer and the user's certificate and private key.

3. The gateway service will communicate with the desired peer, and it will first check the policy for transaction validation.
4. The peer will communicate with the other peers so that the policy is fulfilled, and they will take the necessary precautions. They will verify that the request is in the correct format, that it has not yet been made, that the user's signature is valid, and that the user has permissions to carry out the transaction.
5. Each peer will execute the chaincode with the inputs from the transaction proposal and will generate test results, that is, without changing the current state.
6. Each peer will sign the generated result and return it to the main peer. The main peer will put all these results together into a proposal and send it to the orderer.
7. The orderer will sort the transactions and place them in a block, which it will send back to all peers present on the channel.
8. Each peer will verify that the policy has been adhered to and that there have been no changes since the test results were generated.
9. If everything is valid, each peer will proceed to add the new block to its ledger.

To give a visual idea of this process, image 4.3 shows a sequence diagram with four organizations represented, an orderer, and the gateway service. Let's assume that the transaction validation policy requires that the majority of peers validate the transaction proposal, and therefore, at least three peers must validate the transaction and execute the chaincode.

Ledger

The ledger in Hyperledger Fabric is made up of two components that are key to understanding how storage works:

- **World state**, is a database that stores the current state of objects. Let's assume that a textile product is an object and contains a name, this information is stored in the state database with a key-value pair. If the name is changed, the state is updated.
- **Blockchain**, stores all the steps that led us to the current state. It is a historic record that contains information about the changes of objects up to their current state.

Let's assume we have an empty blockchain and we want to add a new object, for example, a textile product with name and description. Upon its addition, the world state is changed and information about this object is added in a key-value format, while in the blockchain the transaction that led us to this new state is

added. Now, let's assume we want to change the product name. The world state is updated again, changing the name of the product and keeping the description, while in the blockchain a new transaction is added where the object has a new name and the same description. The result of these two iterations is a world state with only the current object information, and a blockchain with the two transactions that led us to the current state. In this way, we manage to keep the history of the products.

As far as the storage of the state of the blockchain two possibilities are offered. The first is LevelDB which is usually the default, but has some disadvantages. As an alternative, CouchDB appears, which turns the disadvantages of LevelDB into its advantages. It is possible to query more complex JSON data, as well as create indexes, which is not possible in LevelDB, but being a database that exists externally to Hyperledger Fabric makes it a bit of a problem [LogRocket, 2022]. It all depends on the needs of each enterprise to adopt the one that best aligns with their requirements.

Case study

Walmart wanted better traceability of their supply-chain, similar to our use case, and for that purpose adopted the Hyperledger Fabric framework. The idea was to run two proof of concept projects to test the framework performance. The first one was to trace mangos sold in the Walmart's US stores and the second one was to trace pork sold in its China stores. Both of the projects turned out successful making Walmart wanting to adopt this framework for other products as well [Hyperledger, 2022b]. Having a real solution implemented and becoming successful makes Hyperledger Fabric an interesting option.

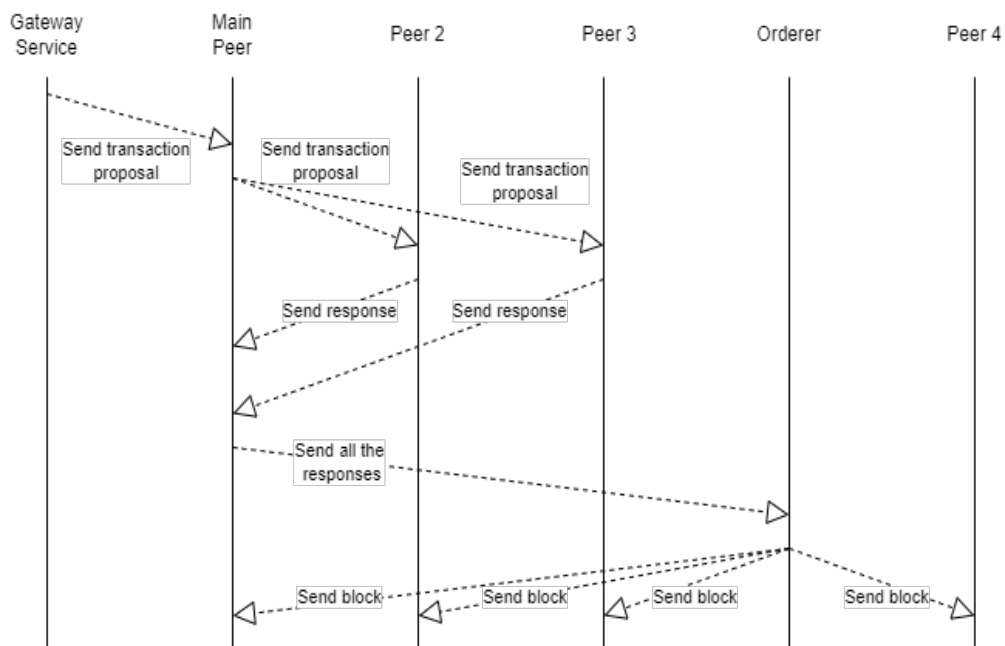


Figure 4.3: Sequence diagram of the example

4.5 Summary

Before moving on to defining the use case and requirements, it is important to make a brief summary of the conclusions drawn from these solutions. To recap, this thesis aims to create a blockchain-based prototype where DPP's of textile products will be stored. The textile sector causes major impacts on the environment due to the processes it uses. Therefore, the goal is to provide the textile industry with a prototype that can ease the transition to a CE, and also guide the EU with one of the first blockchain-based DPP initiatives in the textile sector.

From the solutions studied in section 4.2, we can deduce that for each business and specific product we may have a variety of steps related to the product's life cycle. Thus, for our use case, the definition of these steps must be done together with the case company to represent the life cycle of the textile products, and to detail the data that the fiber production phase must have. It has been suggested in the literature that "many [solutions for DPP] utilise decentralised technologies (Blockchain) [... but] it becomes evident that different capabilities and information types require different system features. Each discrete system in the ecosystem could adopt different types of technology architecture [... and [f]uture research should investigate system architectures in existing DPP systems (and those that provide similar capability) to model common Enterprise Architectures and Solution Architectures for the DPPE, particularly at key integration points" [King et al., 2023].

Regarding what was studied in section 4.3, we conclude that the industry still lacks real use cases to motivate the application of blockchain integrated with DPP's. Apart from the two solutions found that were tested in real contexts, the remainder were very conceptual, since they only presented a proposal for the architecture of the DPP's. Furthermore, the textile sector specifically does not yet have solutions tested in real contexts, raising the need for a solution like the one presented in this thesis.

Finally, we understand that in general Hyperledger Fabric and Sawtooth are similar in many ways, making it difficult to choose one of them for our solution. However, due to the offer of creating several channels within the same network as for the better transaction efficiency that it provides [Hasib, 2021], and this thesis being the follow-up of another one that adopted Hyperledger Fabric, the decision was to adopt Hyperledger Fabric from this point forward. In the next chapter, we proceed to present the use case and respective requirements.

Chapter 5

Use case and requirements definition

This chapter moves on to the definition of the use case, the functional and non-functional requirements, which had the help of the case company based on the in-depth literature review on DPP's, and some blockchain features to understand its adoption in this use case.

5.1 Use case

This use case consists of the development of a DPP for the textile sector, with the aim of facilitating the textiles transition to a more sustainable and circular economy. For the creation of the prototype, and concerning the storage of the DPP's, it was necessary to identify all the phases of the textile product life cycle. So, with the support and confirmation of the case company, the supply chain phases follow as listed below:

1. Raw material.
2. Fiber producer.
3. Fabric manufacturing.
4. Apparel manufacturing.
5. Retailer.
6. Final consumer.

With regard to stakeholders, these are represented by an organization in each of the supply-chain phases, and by an organization representing a certifying entity (e.g., third-party institutions for product testing).

Blockchain being an initial restriction of the thesis, and due to the need for organizations to interact with the prototype, there must be seven peers in the network: five peers with the possibility of inserting data, referring to the first five phases of

the supply-chain, and a sixth and seventh peer that can only read data, referring to the final consumer and certifying entity organizations. Peers are a type of node present in the Hyperledger network that contain the ledger and chaincode of the channels on which they operate.

Figure 5.1 shows the various stakeholders and how they can interact with the blockchain.

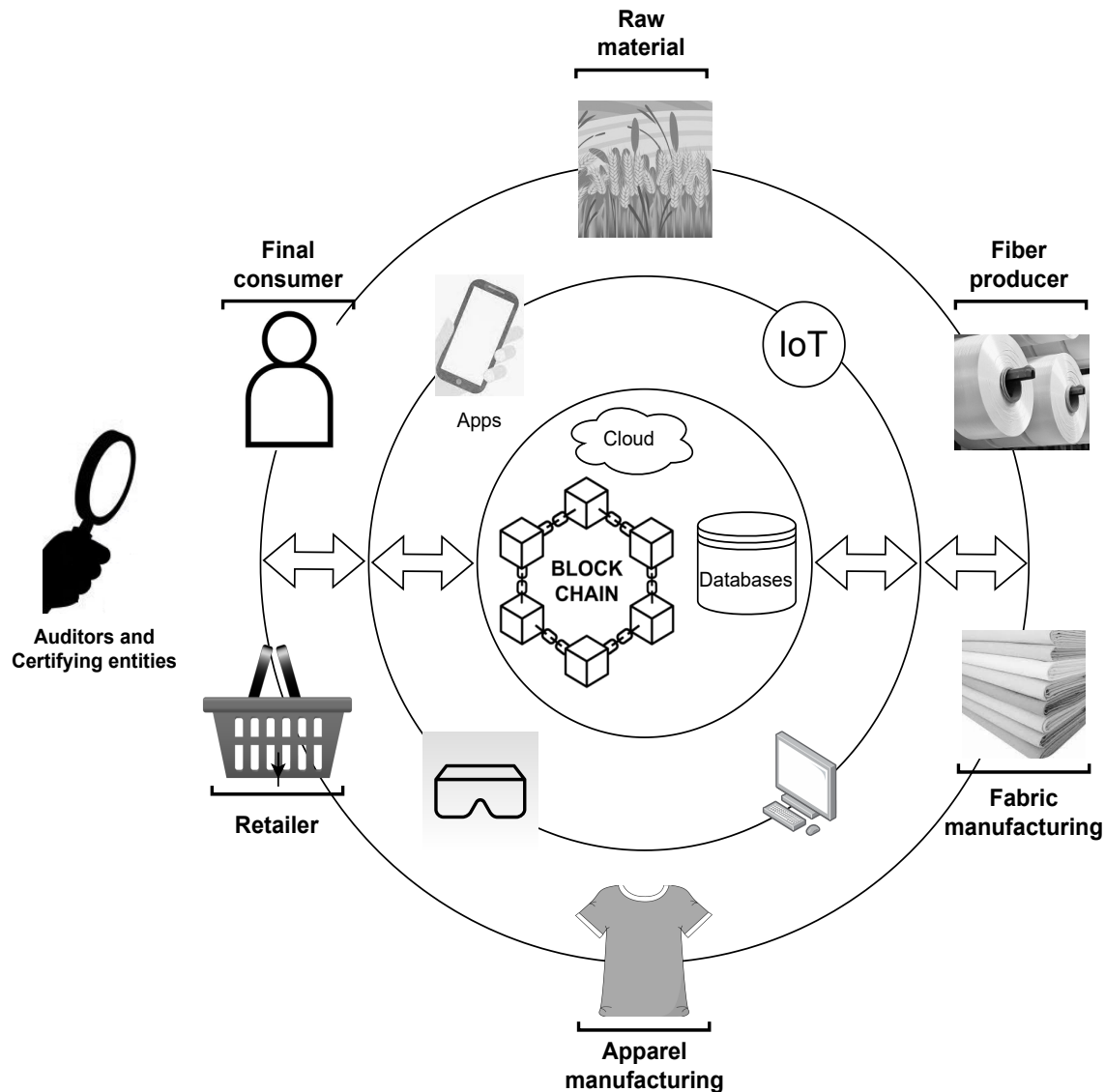


Figure 5.1: Organizations and their interaction with the blockchain

Due to the need to guarantee the credibility of the data, there must be write permissions so that not all organizations can insert data, and that each organization can only insert data specific to their supply chain phase. Also, there must be restrictions on the contents accessed, that is, each organization must have its own reading permissions.

As for the storage of DPP's, it is intended to use only a single channel, where all organizations can act in accordance with the previously discussed permissions, since there is no need to distribute information through different channels. As

explained in section 4.4, channels are a type of private subnet of communication between one or more specific organizations present in the network.

To keep private information isolated from public information, there is the need to use collections. These collections allow, within the same ledger, to store private information that is only seen by the desired organizations. Assuming there is a collection where only the organizations Fiber Producer and Raw Material are associated, only these can insert and retrieve information. By default, each organization has a private collection where only them can insert and read data.

Different text fields are needed to store the product name, location and coordinates that are stored on the blockchain. It is also necessary to store files such as: images, videos, or even any other type of documents that are associated with the products (e.g., certificates, tests, product photos). Therefore, it is necessary to explore the use of off-chain data so as not to overload the blockchain with heavy files. Off-chain data is all data stored outside of the blockchain.

There are two storage options for associated files:

- **Cloud.**
- **Local**, stored locally on the server side, and the file path, filename, and file description stored in the external database.

Regarding what is stored on-chain in these situations, the goal is to hash the file using the SHA-256 algorithm and store the result on the blockchain along with the file name and a description entered by the user via input.

Each organization must login into their account to prevent unknown parties to insert data or get unauthorized access to the prototype. The usernames and passwords must be stored in a JSON file, where each username and password is hashed using the SHA-256 algorithm.

5.2 Requirements

The textile industry needs to adopt circular models to reduce the waste and pollution it creates each year. To get an idea of the environmental impact that the textile industry causes, the production of a t-shirt requires an estimated 2.700 liters of water, an amount that is enough drinking water for one person for 2.5 years, and for the washing of synthetics, 0.5 million tonnes of microfibers are released into the ocean every year [Parliament, 2020].

With the use of DPP's in this use case, it is expected that end consumers will now be able to make more informed choices about the products they buy, that authorities can verify the products compliance with regulations and laws, and that all stakeholders can now navigate through the life cycle of the textile products. Therefore, with information on all products life cycle, the textile industry can reduce the waste and pollution it produces by monitoring their supply chains (e.g., reuse or recycle the fibers instead of throwing them into the ocean).

For the storage of these DPP's, and due to the need to maintain the integrity of this data, blockchain technology will be used, which carries characteristics such as immutability and resistance to tampering.

As in any project, it is important to have a prioritization of the requirements to help realize those that are most important and that should be implemented first. Therefore, the prioritization technique designed by Dai Clegg in 1994 [AgileBusiness, 2022] will be used. The technique is known as MoSCoW prioritization technique and the capital letters correspond to the following:

- **M** - Must have.
- **S** - Should have.
- **C** - Could have.
- **W** - Won't have.

To aid the definition of requirements, both functional and non-functional, the organizations Raw material, Fiber producer, Fabric manufacturing, Apparel manufacturing and Retailer will be represented hereafter, as Producers.

Functional requirements

With regard to fulfilling the objectives of this thesis, and presenting a functional prototype for the textile industry, below are the defined functional requirements:

- Producers **must be** able to insert new products, events, and transformations on the blockchain.
- Producers **must be** able to insert new products, originated by those already existing on the blockchain.
- Producers **must be** able to associate files with products, events and transformations.
- Producers **must have** access to all products, events, transformations and associated files.
- Producers **must be** able to download files associated with products, events and transformations.
- Final consumer organization **must have** access only to events, without access to associated files and still having restriction on event fields.
- Certifying entity organization **must have** access to all products, events and transformations, not having access to associated files.
- Producers **should be** able to insert private data in their personal collection.
- Raw material and Fiber producer organizations **should be** able to insert private data in a collection where only the two are present.

- All organizations **must** log in to interact with the prototype.
- All organizations **must be** able to navigate through the products life cycle.

For the storage of files, the Dropbox API will be used regarding cloud storage, and PostgreSQL will be used as an external database.

Non-functional requirements

With the aim of offering a prototype capable of guaranteeing trust between the participating entities, and the credibility of the data, the decision was to focus solely on security.

Despite the decision to proceed with only security, it does not mean that other quality attributes were not taken into account, for example, scalability and reliability are something important for a prototype of this type, but it is difficult to test without putting the prototype into practice. Still, scalability and reliability were already taken into account when the decision was made to proceed with Hyperledger Fabric. Another non-functional requirement taken into account despite not being tested is the performance, since others studies already support that performance exists in a Hyperledger Fabric environment [Hyperledger, 2023a], [Wrisez, 2023].

The definition of the non-functional requirements is presented as follows, along with a technical constraint.

Title	Blockchain adoption
ID	C1
Description	The prototype must use blockchain technology for the storage of the life cycle of textile products

Table 5.1: Blockchain constraint

Title	Unauthorized data insertion
Category	Security
ID	QA1
Description	The prototype shall enforce access control measures to ensure that unauthorized organizations can't insert new information on the blockchain
Rationale	By restricting data insertion solely to authorized organizations, the prototype can maintain the integrity and security of the blockchain, preventing unauthorized entities from tampering with the data or introducing potentially malicious information
Measurable Criteria	Only authorized organizations shall have the capability to insert data into the blockchain. Unauthorized attempts to insert data should be rejected and logged

Table 5.2: Unauthorized data insertion

Title	Unauthorized data access
Category	Security
ID	QA2
Description	The prototype shall ensure that only authorized organizations present within the network have access to the data
Rationale	By limiting data access to authorized organizations within the network, the prototype can protect the confidentiality and privacy of the data, preventing unauthorized entities from viewing or retrieving sensitive information
Measurable Criteria	Only organizations that are authenticated and recognized within the network shall be granted access to the data. Unauthorized attempts to access the data should be denied and logged

Table 5.3: Unauthorized data access

Title	Certificate and key management
Category	Security
ID	QA4
Description	The prototype shall ensure that if a certificate or key is tampered with or deleted, organizations involved shall be prevented from further participation in the network
Rationale	By implementing strict certificate and key management practices, the prototype can maintain the integrity and security of the network. If a certificate or key is compromised, it is crucial to immediately revoke the privileges of the associated organizations to prevent unauthorized access or malicious activities
Measurable Criteria	In the event of certificate or key tampering or deletion, the affected organizations shall have their network privileges revoked, rendering them unable to continue operating within the network

Table 5.4: Certificate and key management

5.3 Blockchain adoption in this use case

Blockchain is a technology with many interesting characteristics for storing DPPs, favoring the industry's transition to a CE. Especially in the textile sector, where it will be mandatory to adopt these passports by 2030 at the latest.

By storing the life cycle of products in a sequence of chronologically ordered blocks, the blockchain manages to guarantee traceability and transparency throughout the entire supply chain [Cem, 2023]. The actors can access the information needed about the products at any time, whether from the Raw material or the Final consumer phase, since everything is stored in the same place and shared by

all actors.

The possibility for actors to share the information present on the blockchain comes from the decentralization that this technology provides. All actors have a copy of the ledger, making it difficult for data tampering to occur, and therefore, create an environment of trust between the intervening actors. Also, consensus is required for new blocks to be added to the ledger: all actors must validate the block for it to be added to the ledger. It is then guaranteed that the present information is valid and reliable, also increasing the trust of end users when choosing which products to buy.

Due to the immutability and decentralization that this technology offers, and also, in the case of private blockchains, the necessity for actors to have a valid certificate to be able to interact with the blockchain, data security is assured.

Additionally, textile supply-chains face several difficulties in terms of performance due to the usage of manual processes, paperwork and in most cases several intermediaries. Blockchain, with the use of smart contracts, makes validation processes much faster as these contracts make the necessary updates when certain conditions are met [Cem, 2023].

This technology has been increasingly sought after in this business area due to the various advantages it carries, and everything indicates that demand will continue to increase as this technology continues to mature [Henry, 2023].

5.4 Summary

This chapter started with the definition of the use case, after discussion with the case company, to understand which technical aspects are required for the development of the prototype. Subsequently, derived from the use case, the high-level functional and non-functional requirements were identified.

Finally, it is concluded that blockchain being a technology that allows the storage of data in a sequence of chronologically ordered immutable blocks, providing participants with the life cycle of products at any time, can improve the lack of traceability and transparency in supply-chain businesses [Ahmed balaghi, 2021]. Therefore, a variety of problems can be tackled (e.g., counterfeiting, contamination of products, components harmful to health or environment, and compliance with regulations and laws). In the next chapter, follows the presentation of the prototype architecture.

Chapter 6

Architecture

After defining the use case and the most relevant requirements, in this chapter, we will move on to the definition of the architecture of the prototype. To help with the explanation, the C4 model will be used, which offers the construction of four levels of abstraction, respectively, (1) Context, (2) Container, (3) Component and (4) Code [C4Model, 2022].

We will make use of the first three levels of abstraction to give a full point of view of the architecture:

- **Context layer**, represents what the prototype does from a high level perspective, and which actors and external systems interact with the main system [Gliffy, 2021].
- **Container layer**, details the containers the prototype will use and how they communicate with each other [Lucidchart, 2023].
- **Component layer**, divides the containers into several components to understand the responsibilities of each one, and the technologies to be used.

6.1 Context layer

Starting with the first and highest level of abstraction, figure 6.1 depicts the context diagram of the prototype with only the Raw material organization. Still, the full diagram is included in appendix A.

Each organization is composed of an admin with access to all blockchain functionality (e.g., requesting new user registration or deploying smart contracts) and a user who has permissions defined by the admin. Additionally, there is also the organization of orderers, which is composed of an admin, with the ability to add new orderers to the channels. This organization is composed of the so called ordering nodes or orderers, which receive and order the transactions, placing them in a new block and consequently sending them to the endorsing peers.

All these organizations participate in what is the main point of storage and where the life cycle of textile products will be stored: the channel. As we will see in the next levels, the organization's administrator accesses the channel directly, while the users access it through the client application.

The client application allows the user to have a more interactive interface, avoiding the impracticality of the command line. Finally, we can verify the existence of an external system, more specifically, a Dropbox API that will allow us to store the files associated with each product in the Cloud.

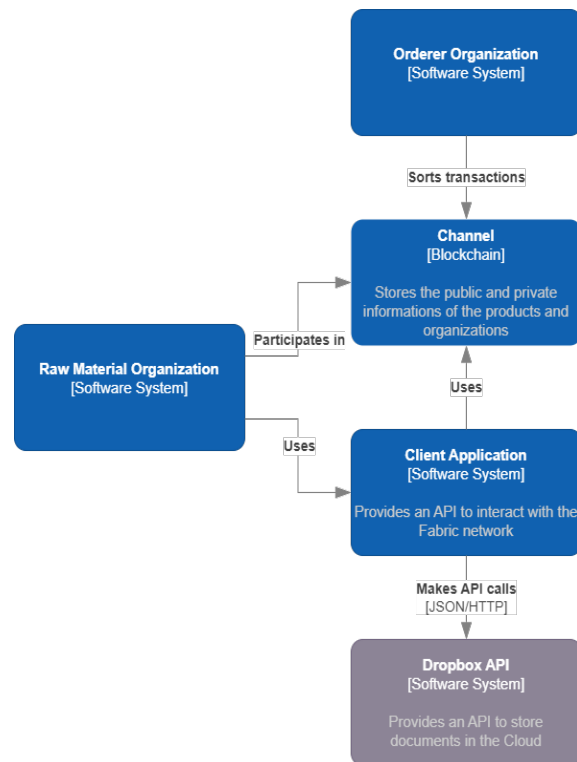


Figure 6.1: Context diagram for a single organization

6.2 Container layer

Now, moving on to the container diagram and advancing to the second level of abstraction of the C4 model, the intention is to describe the prototype, showing the API's that constitute it and how organizations can interact with channels and acquire their identity on the network.

Due to the high number of containers and connections in the container diagram, it has become too complex to be explained as a whole, and therefore, certain parts of the total diagram will be presented individually to ease understanding. Still, the full diagram is included in appendix B. The complete diagram represents a single organization, noting that for the remaining organizations the architecture is the same. Only one is shown for the sake of readability and simplicity.

Figure 6.2 presents the container diagram. As already mentioned in section 4.4,

all nodes in the network must carry an identity to be able to perform any type of action in the network. In this way, we can see the representation of the local MSP whose function is to identify the actors. Through the Command Line Interface (CLI), the organization admin can interact with the Fabric CA Client where the request for registration and enrollment of identities is made. As a follow-up, these requests are sent to the Fabric CA Server, which registers them and issues the certificates, storing them in the local MSP.

Consequently, with the identities and certificates created, the actors can interact with the network. The administrator, having all the permissions in the network, can use the CLI to change settings and send transactions, while the user only has some permissions. The user can use the CLI to interact with the channel as well, however, it is more practical to use the provided client application.

Although already visible in this part of the diagram, the client application will be explained later in its corresponding part. Notwithstanding, it can already be seen that there is a front-end and a back-end for interacting with the network.

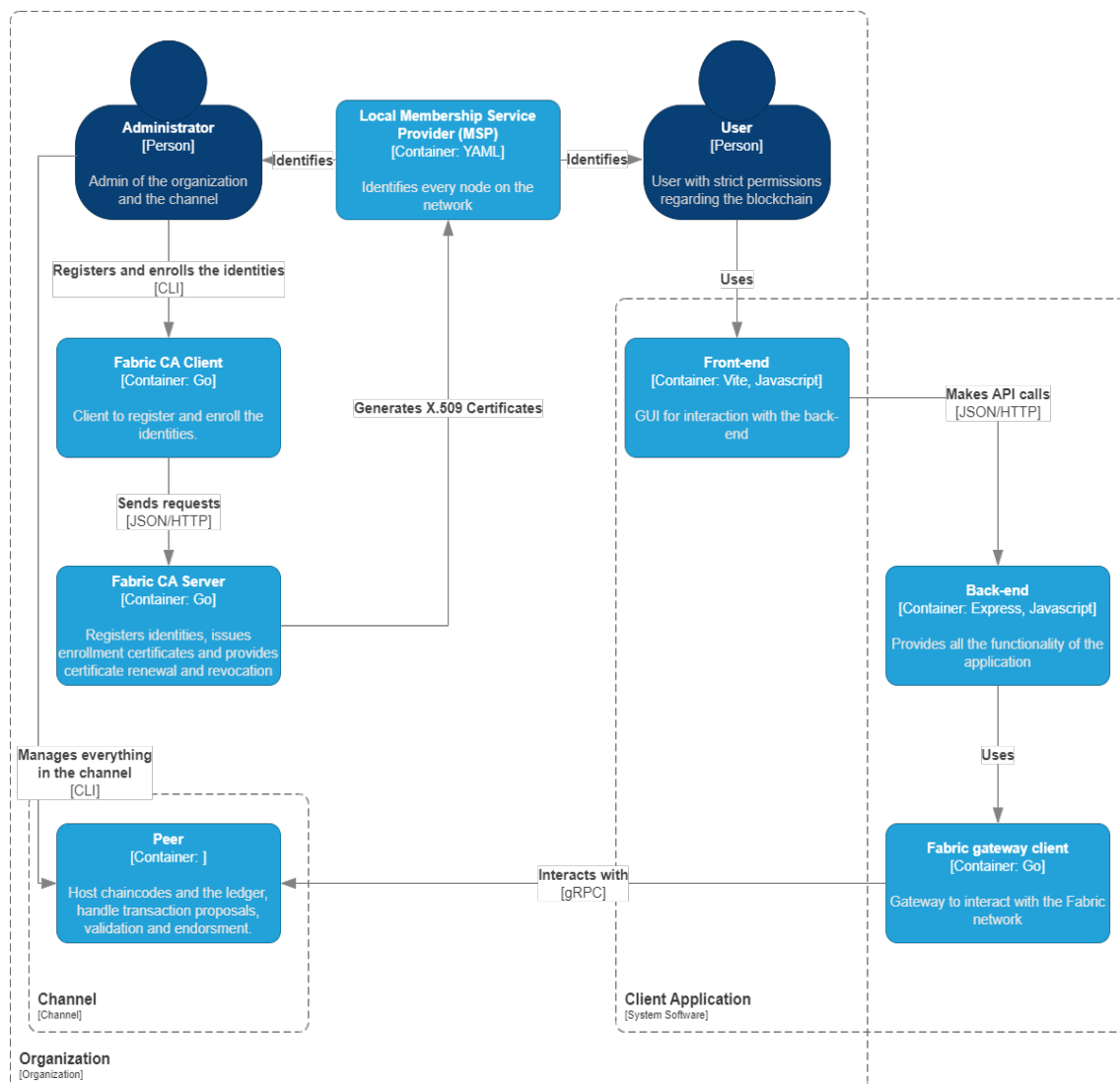


Figure 6.2: Container diagram - Organization only

Representing what will be the main entry point for users on our prototype, the client application is shown in figure 6.3. Initially, the user is presented with a front-end developed in Vue, a JavaScript framework for building modern web User Interface (UI) applications [VueJS, 2023], which, through a REST-API calls the back-end, developed in Express, a minimal and flexible nodejs web application framework that provides a robust set of features [Express, 2023]. Subsequently, the back-end makes use of the Fabric gateway client API, provided by Hyperledger Fabric, which connects to the network and allows the interaction with the smart contract through Google Remote Procedure Call (gRPC)'s.

To store the files associated with each product, the client application allows cloud and database storage. The front-end makes use of the Dropbox API, and the back-end communicates with a PostgreSQL database.

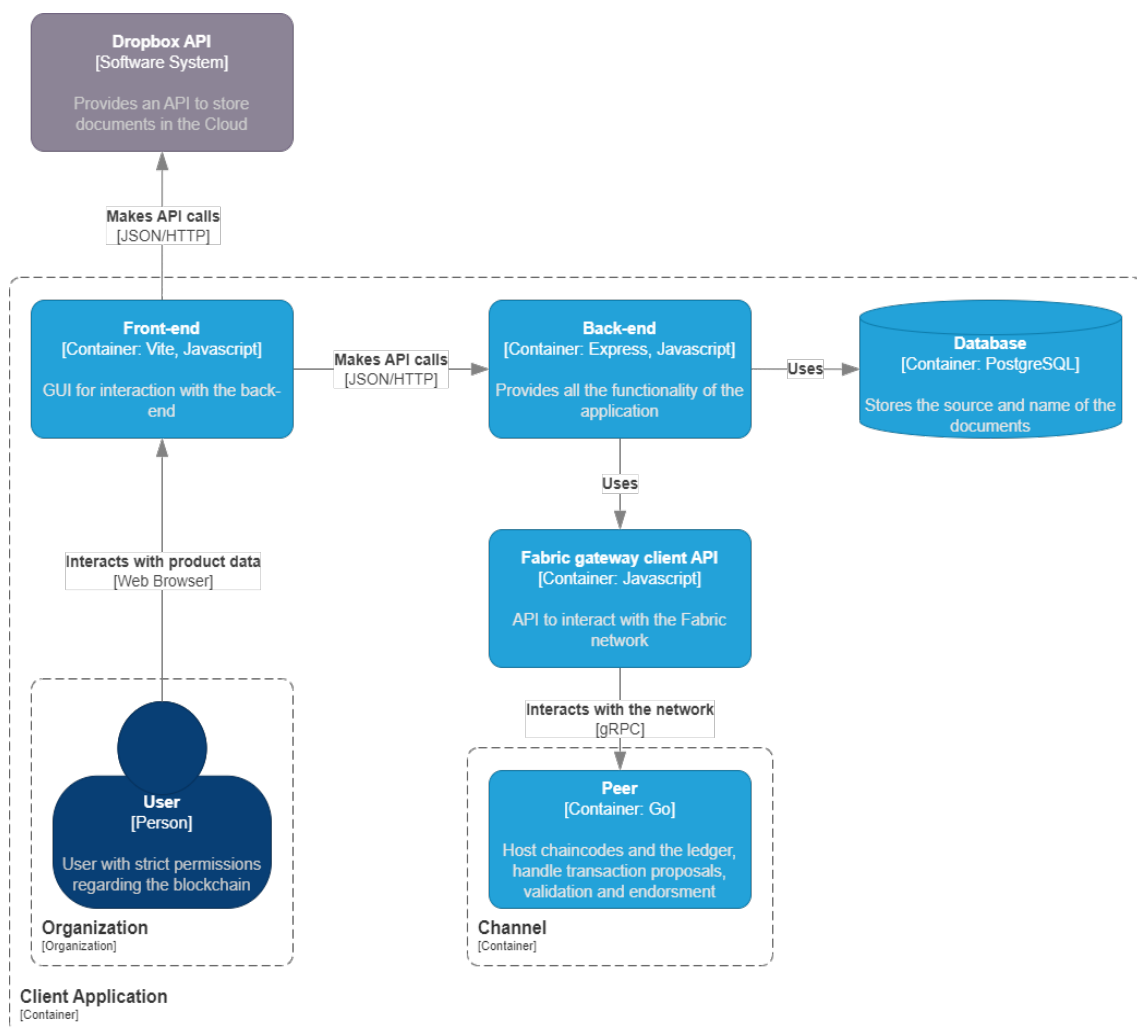


Figure 6.3: Container diagram - Client Application only

Moving on, we have the part of the diagram corresponding to the channel. The seven peers are represented in figure 6.4, representative of each of the organizations, each of which has its own ledger and smart contracts. Below the peers, we have the orderer that will receive the transactions and order them, returning them to the peers with endorsing roles.

At the bottom, we have the channel configuration that contains the permissions and role of each organization in the channel. For example, information about who the endorsers are.

As the last part of the complete diagram, we have the portion referring to the organization of the orderers (figure 6.5). Like all other organizations, it needs an MSP to be identified in the channels in which it operates. For the purpose of ordering the transactions, it has a peer in the channel.

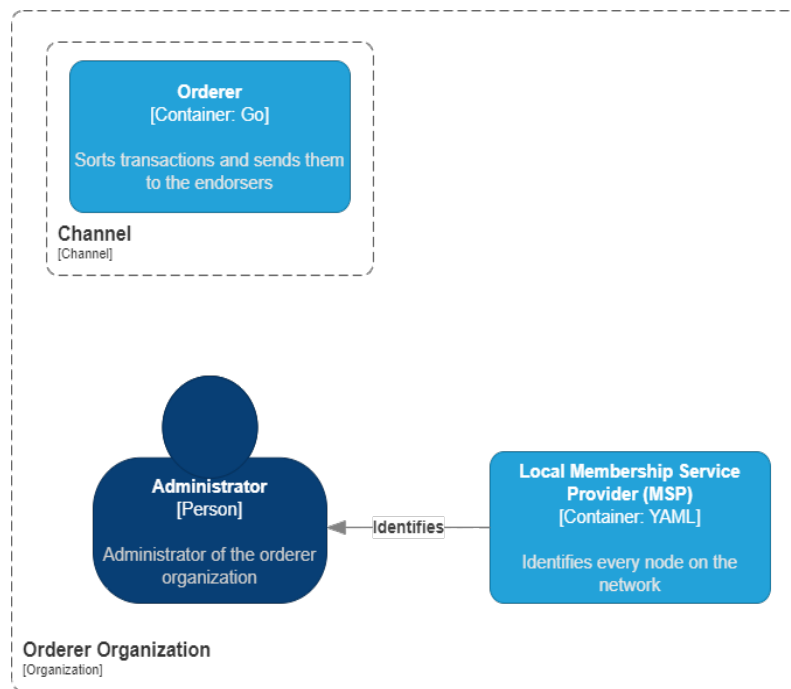


Figure 6.5: Container diagram - Orderer Organization only

6.3 Component layer

Similarly to the previous level, the component diagram will be divided into portions to aid the explanation, and the understanding. Also, as the previous one, only one organization is represented for the same reasons, with the full diagram included in appendix C.

Therefore, to finish defining our architecture, in figure 6.6, we have a representation of what happens in each peer. It is visible that each one contains a copy of the smart contract present on the channel, and the channel configuration for identity verification purposes.

The central point of the peer is the peer node which performs all actions, be it deploying new smart contracts, or introducing new peers into the network. This component is accessed through the peer CLI in the case of the administrator, and the Fabric gateway service otherwise.

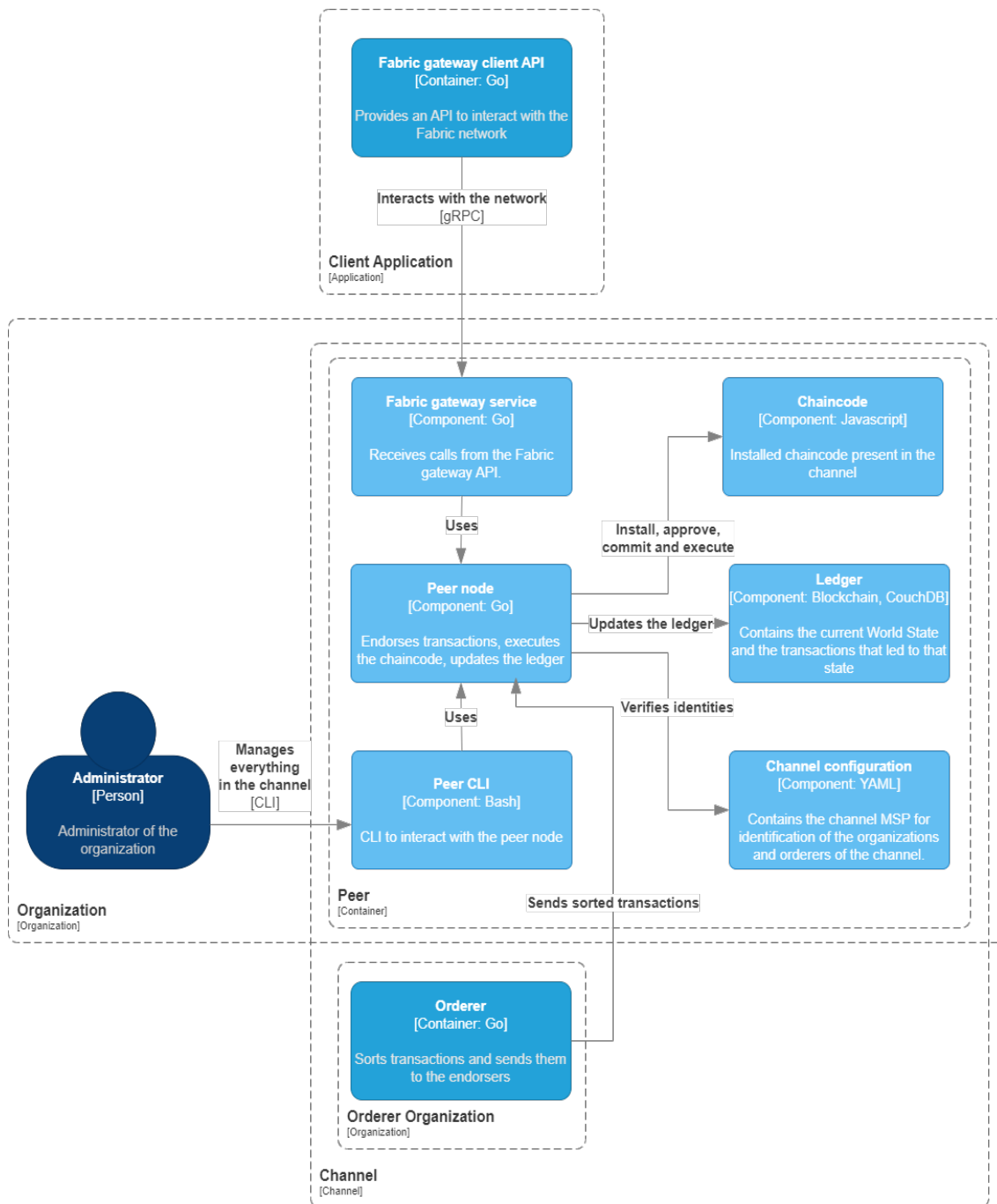


Figure 6.6: Component diagram - Peer only, adapted from [Wrisez, 2023]

6.4 Summary

At this point we come to the end of the definition of the solution, that is, definition of the use case, requirements and architecture. To provide an in-depth presentation of the prototype architecture, the first three levels of the C4 model were used. Having in special detail the functioning of each peer of the network. In the next chapter, we advance to the development of the prototype, starting with the steps to put the platform developed by [Wrisez, 2023] up and running.

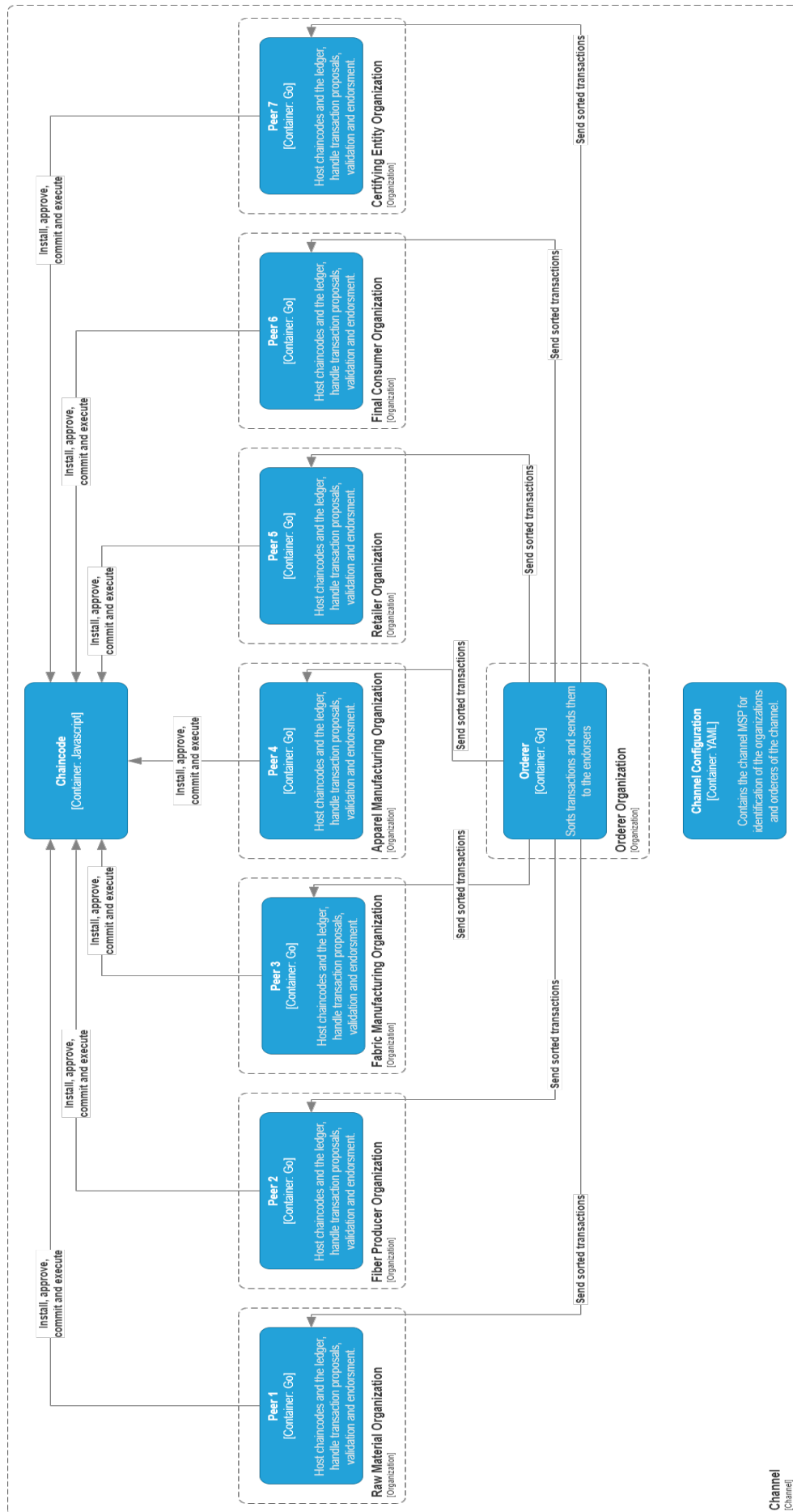


Figure 6.4: Container diagram - Channel only

Chapter 7

Development

Moving on to the development of the prototype, in section 7.1 are shown the steps to set up the platform developed by [Wrisez, 2023]. Next, in section 7.2, are presented some development steps of the prototype, and finally, in section 7.3, the functionalities, some verification measures, and a demonstration of the use case defined in chapter 5 are shown.

7.1 Setting up the platform

As mentioned earlier, this thesis is the follow-up of another one [Wrisez, 2023]. Our prototype appears as an update to the platform developed by [Wrisez, 2023], adapting it to the use case of textiles while also adding new functionalities. Thus, this section summarizes everything that was already implemented prior to the development of this thesis, and presents the prerequisites and the steps to get the platform up and running.

7.1.1 Starting point

Initially, it was necessary to understand the state of the platform implemented by [Wrisez, 2023], so that it would be possible to advance to the development of the prototype and make the necessary adjustments.

Thus, the initial version created by [Wrisez, 2023] consists of a Hyperledger Fabric blockchain with two organizations, a smart contract, a channel, an orderer, and is interacted with via a client application that shows results to the end user.

A smart contract was implemented from scratch and allows a number of features already described in [Wrisez, 2023], which generically consist of inserting products, events and transformations, and querying the inserted data. It is important to mention that through the smart contract it was also possible to insert private data associated with each organization: each organization could insert private information visible only to itself.

To facilitate the visualization of the results, a front-end was implemented where the user is faced directly with the list of products already created, and the various functionalities for creating products, events and transformations. It should be noted that all these functionalities are exempt from verification of writing or reading permissions, with the exception of verifying the user's identity for interaction with the blockchain.

7.1.2 Prerequisites

Before setting up the fabric network, and making use of the back-end and front-end, it is necessary to install some prerequisites, and some dependencies. Listed below are those needed to get the fabric network up and running:

- docker and docker-compose [Docker, 2023].
- jq, a lightweight and flexible command-line JSON processor [Jqlang, 2023].
- fabric CLI tool binaries and docker images.

Since [Wrisez, 2023] used JavaScript it was easier to follow with the same language, so, as to use it, the following dependencies are needed:

- node, version 14.21.2, an open-source, cross-platform JavaScript runtime environment [NodeJS, 2023].
- npm, or node package manager, version 9.5.1, a library and registry for JavaScript software packages [NpmJS, 2023].

Regarding the chaincode, and being aware that no major changes were made in relation to the previously implemented one, the dependencies follow the same:

- json-stringify-deterministic, version 1.0.8, and sort-keys-recursive, version 2.1.8.
- fabric-contract-api, version 2.2.3 [Hyperledger, 2023b].

Moving on to the client application and starting with the backend, below are the necessary dependencies:

- uuid, version 9.0.0, creates unique id's.
- pg, version 8.10.0, postgresSQL client for JavaScript.
- multer, version 1.4.5-lts.1, middleware for handling multipart/form-data.
- fabric-gateway, version 1.1.2, API to interact with the fabric network [Hyperledger, 2023c].

- `express`, version 4.18.2, minimal and flexible web application framework [Express, 2023].

Finally, the dependencies related to the front-end:

- `vue`, version 3.2.45, JavaScript framework for building modern web UI [VueJS, 2023].
- `vuetify`, version 3.1.8, a vue component framework [Vuetify, 2023].
- `dropbox`, version 10.34.0, API to allow storage of files in the cloud [Dropbox, 2023].
- `crypto-js`, version 4.1.1, library to provide hash and encryption algorithms.
- `form-data`, version 4.0.0, library to create readable "multipart/form-data" streams.

7.1.3 Assembling the fabric network

Before explaining how to get the prototype up and running, it is fundamental to explain the process of creating and adding new organizations. Hyperledger Fabric already creates and adds by default two organizations to a new channel when the network is initialized. However, apart from the first two, the process of adding new organizations is not dynamic, and therefore, it is required to do it manually. Still, to add a third organization, Hyperledger Fabric provides the required files to create and add the organization to the channel.

For our use case it is necessary to have seven organizations present in the channel, but with the files provided by Hyperledger Fabric, it would only be possible to add three organizations to the channel: two added by default, and a third for which Hyperledger provides the files.

So, first we will understand how to add the third organization with the files already provided, and later we will understand how to create the files for the remaining organizations. Finally, we will go through the process of installing the chaincode in the seven organizations.

Therefore, to add a third organization to the network, and to the channel, it is necessary to run the file provided by Hyperledger Fabric, present in the following path: `"/test-network/addOrg3/addOrg3.sh"`. This file generates the organization's identification data: the certificate and respective keys, and adds the organization to the network and channel.

Now, taking a fourth organization as an example, let's explain its addition to the blockchain. Unlike the third organization, there is not a file that generates all the data and adds the organization to the channel, so it was necessary to create one. Below is a list with the identification of the steps:

- Copy and paste the `addOrg3` folder, renaming it `addOrg4`, and replacing everything referring to the third organization to a fourth organization, and changing the necessary ports.
- Copy and paste the `org3-scripts` folder, renaming it `org4-scripts`, and replacing everything referring to the third organization to a fourth organization, and changing the necessary ports.
- Change the `envVar.sh` file by adding the fourth organization information, allowing to act as this organization going forward.
- Change the file `setAnchorPeer.sh` by adding the fourth organization information, allowing to set the organization's peer as an anchor in the channel.

Following the same process, it is possible to create the remaining organizations, that is, from the fifth to the seventh. It is important to remember that organizations are only accepted when the majority accepts, therefore, it is also necessary to change the `updateChannelConfig.sh` file whenever needed, as this is where the acceptance of organizations is verified. Taking the case of a fifth organization as example, it was indispensable to add code so that the third or fourth organization, previously added, would accept the organization's addition.

With all organizations inserted in the channel, it is necessary to install the chaincode in all of them, so that they can execute it. Thus, the `deployCC.sh` file, present in the following path "`.../test-network/scripts/`", was changed, which installs the chaincode in all organizations. From this point on, the blockchain is ready to be interacted with.

7.2 Prototype development

With the platform developed by [Wrisez, 2023] ready to move forward with the development, this section presents the new data model, since changes were necessary, the permissions that each organization has on the network, how private data is stored, and how we can associate files to products, and respective events and transformations.

7.2.1 Data structure

To support all the intended functionalities, and to store information relating to the life cycle of textiles, it was necessary to change the data structure defined by [Wrisez, 2023]. Therefore, in appendix C is included an example of the data structure. Below are the respective description of each of the parameters of the data structure, with a distinction between the existing parameters and those that were added.

Existing parameters:

- **Product**, represents the product, consisting of an id, name, respective events and transformations, and the id's of the products that originated it.
- **Event**, represents minor changes to products, such as transportation from one location to another, coloring of fabrics, and cutting of wood.
- **Transformation**, represents major changes associated with a product. For example, a fabric is used for the production of a sweater, or the product is in the hands of the next supply-chain phase.

Events normally represent all processes to which a product is subjected until it is used for the production of another (e.g., use of fibers for the production of fabrics). Therefore, taking the Raw material organization as an example, these processes could be: planting the tree, cutting the log, and cutting the log into various parts.

The transformations do not always indicate that there were physical changes to the product, and this is because when the products pass from one phase to another it is necessary to create new instances of products to represent the new phase.

For example, suppose a log of wood, represented by the Raw material organization, was shipped to a pulp and textile fiber manufacturing plant. There is a need to create a new product on the prototype to represent the Fiber producer phase, since from the moment the wood trunk arrives at the fiber production factory, the Fiber producer organization will insert the data on the prototype.

This detail will be addressed again when the real use case is demonstrated, where it is possible to visualize the life cycle of the products and better understand how the transformations work.

Moving forward, below are the parameters that were necessary to add:

- **Events and transformations id**, identify events and transformations to ease the search for files associated with each one.
- **Originated**, contains the id's of the products that originated the one in question. For example, multiple fibers can create a single fabric, and therefore, the id's of the fibers must be stored in the fabric originated array.
- **Files**, contains objects consisting of name, description, and the hash of the files associated with an event or transformation.

7.2.2 Permissions

As previously mentioned in section 3.4, each network actor must be accompanied by an X.509 certificate and the respective public and private key to be able to sign its transactions, and be identified as a valid actor on the blockchain.

Therefore, upon initialization of the prototype, the X.509 certificates and key pair for each organization are generated. In this phase, the definition of the permissions for each organization takes place. At the time that the organization's registration request is made, these requests carry two associated attributes, one for permissions related to data access and the other related to data insertion. It should be noted that after this registration request, the X.509 certificates contain information on these attributes, thus making a distinction between each organization.

Data insertion permissions

First, for the data insertion permissions there's the "write" attribute that defines the organizations that can insert data on the blockchain. The value is true, if writing is allowed, and false, otherwise. Below we have a table with the identification of this attribute for each represented organization.

Data insertion permissions	
Organization	Data insertion
Raw Material	True
Fiber Producer	True
Fabric Manufacturing	True
Apparel Manufacturing	True
Retailer	True
Final Consumer	False
Certifying Entity	False

Table 7.1: Data insertion permissions

Data access permissions

Similar to data insertion permissions, for data access there's the "read" attribute. The difference is that this one can have more values, differentiating the type of information that each organization can have access to. Below, there is a list explaining what each value of this attribute allows to visualize, followed by a table identifying the permissions attributed for each organization represented:

- **1**, has access to all information, products, respective events and transformations, and associated files.
- **2**, has access to the products and respective events and transformations, but does not have access to the files.
- **3**, has access to the events of each product, with restrictions on the available information about the events. For example, it does not have access to the hashes of the files or the creation date of the events.

These values are only used in the front end code to identify the permissions of each organization. Therefore, to facilitate the perception of the permissions that each organization has, in the following table the cross is used instead of the values.

Data access permissions				
Organization	Products	Events	Transformations	Files
Raw Material	X	X	X	X
Fiber Producer	X	X	X	X
Fabric Manufacturing	X	X	X	X
Apparel Manufacturing	X	X	X	X
Retailer	X	X	X	X
Final Consumer	X	X		
Certifying Entity	X	X	X	

Table 7.2: Read permissions

Permissions of new organizations or peers

Before explaining how to define permissions for new organizations or peers, it is essential to understand who controls the blockchain, that is, who has permissions to add new organizations, smart contracts, etc. So, in order not to have a single point of control, or in other words, to have one organization controlling the entire blockchain, all organizations have this role. In this way, all organizations can perform these actions on the network, more specifically, the administrators of each one.

However, any action that one of the administrators intends to perform, (e.g., adding new organizations) must be accepted by the majority - at least four in the case of our network with seven organizations.

And now the question arises "How to assign permissions to new organizations after the prototype is already running?". In the same way it was done for the other organizations, it is necessary to generate the certificate and key pair, passing the two permission attributes. These attributes are defined by the organization admin who makes the request, and after the certificate is generated, this new organization can now join the channel. Bear in mind that at least four organizations must accept this request for the new organization to join successfully.

From this point on, the organization becomes part of the group of organizations responsible for controlling the blockchain, and therefore, for a new organization or smart contract to be inserted in the network, at least five organizations must accept it.

7.2.3 Private data

As discussed earlier, it was already possible to insert private data for each organization. However, the purpose of this section is to explain how it is possible to insert private data that can only be seen by specific organizations rather than just the one that inserts it. This is possible through the so-called collections, discussed in section 4.4.

To fulfill this need, a json file was created that contains all the collections to be incorporated into the ledger. Figure 7.1, shows a collection called "collection12"

where the organizations Raw Material and Fiber Producer are included. Both the organizations can insert and read the data placed in the collection.

```
[
  {
    "name": "collection12",
    "policy": "OR('Org1MSP.member', 'Org2MSP.member')",
    "requiredPeerCount": 1,
    "maxPeerCount": 1,
    "blockToLive": 1000000,
    "memberOnlyRead": true,
    "memberOnlyWrite": true
  }
]
```

Figure 7.1: Collections configuration file

The downside is that due to the need to verify the organization that is trying to read or insert the data in the collection, the code becomes static. That is, there is the need to check the name of the collections, and if more are added, arises the need to change the smart contract. In that situation, the code is changed and the smart contract is updated in the channel, with the need to be approved by the majority of the organizations.

7.2.4 Files

The prototype allows the association of files, whether its images, text files or any other type of file when creating products, events or transformations. Thus, below is the description of how the storage is carried out for the two possibilities, cloud and locally, respectively.

Cloud

For storage in the cloud, the Dropbox API is used, which, through an access key, allows storing files in a folder previously created. In this case, only the file is stored without any metadata for identification, and it is stored in a specific folder to facilitate the search when it is necessary to show the files to the user.

Thus, it was necessary to define a structure for the path of the files. To make it easier to understand, below is an example of how files are stored in the cloud:

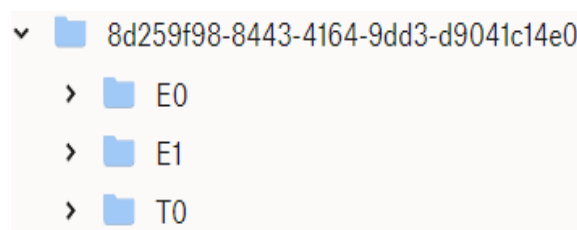


Figure 7.2: Dropbox storage example

The first value refers to the id of the product to which the files are associated, and the second one is the identification of the event/transformation according to the character "E" or "T", accompanied by the respective id. In the example, we have a product with two events of id 0 and 1 and a single transformation with id 0.

Finally, and in order to distinguish files with the same name, a prefix is added to the name, as it is visible in the figure below. Be cognizant that all files have the prefix and not only the ones with the same name.

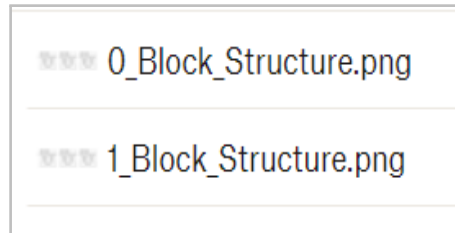


Figure 7.3: Dropbox file example

Local

Moving on to the second option and serving as a backup for cloud storage, local storage arises with the help of a PostgreSQL database. In the same sense that it was necessary to define the structure to store files in the cloud, the same approach was used for local storage. The difference is that a database is used to store the path and name of the file.

When a product, event or transformation is created, the associated files are sent to the back-end where a middleware to handle files, called *multer*, comes into play. As soon as the backend receives the files, the middleware is activated, automatically storing the files in the provided path (figure 7.4).

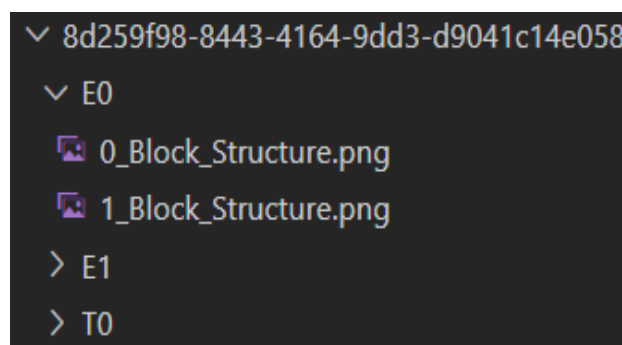


Figure 7.4: Local storage example

After the middleware finishes its work, the respective path and name of the file are stored in the database (figure 7.5).

	id [PK] integer	src text	name text
1	167	/home/hugo/tese/implementation/client-app/backend/images/8d259f98-8443-4164-9dd3-d9041c14e058/E0	0_Block_Structure.png
2	168	/home/hugo/tese/implementation/client-app/backend/images/8d259f98-8443-4164-9dd3-d9041c14e058/E0	1_Block_Structure.png
3	169	/home/hugo/tese/implementation/client-app/backend/images/8d259f98-8443-4164-9dd3-d9041c14e058/E1	0_collections.png
4	170	/home/hugo/tese/implementation/client-app/backend/images/8d259f98-8443-4164-9dd3-d9041c14e058/T0	0_Container_Diagram_Orderer.png

Figure 7.5: PostgreSQL storage example

Blockchain

Regarding what is stored on-chain, and as discussed in the data structure section, an array of objects associated with each event and transformation is used.

Each object consists of the file name, a brief description entered by the user via input when uploading the file, and the respective hash.

Blockchain immutability is used to our advantage: when there is a need to show the files to the user, the values that are in the blockchain are first displayed. This is because there may be problems in the cloud or locally, such as file corruption or deletion. Thus, we guarantee that we are always showing all the files associated with the product in question.

When the user wants to download one of the associated files, a request is made to the Dropbox API to send the file. Here, the hash in the blockchain is verified with that of the received file, and if there are no problems, the download is carried out successfully.

7.3 Prototype overview

Finally, this last section shows the functionalities of the prototype with a simple example, some verification measures to ensure data integrity, and then demonstrates the use case created jointly with the case company, and defined in chapter 5.

7.3.1 Prototype user interface

To better demonstrate the prototype's functionalities, we will take the place of a user belonging to the Raw Material organization. The intention is to demonstrate the steps to create products, events, transformations, and navigate through the products life cycle.

Login

Initially, the user is presented with a login page (figure 7.6) , where they must enter the credentials so as to interact with the blockchain. In this example, as

mentioned, the user from the organization Raw material, whose username and password are "1", is going to be used.

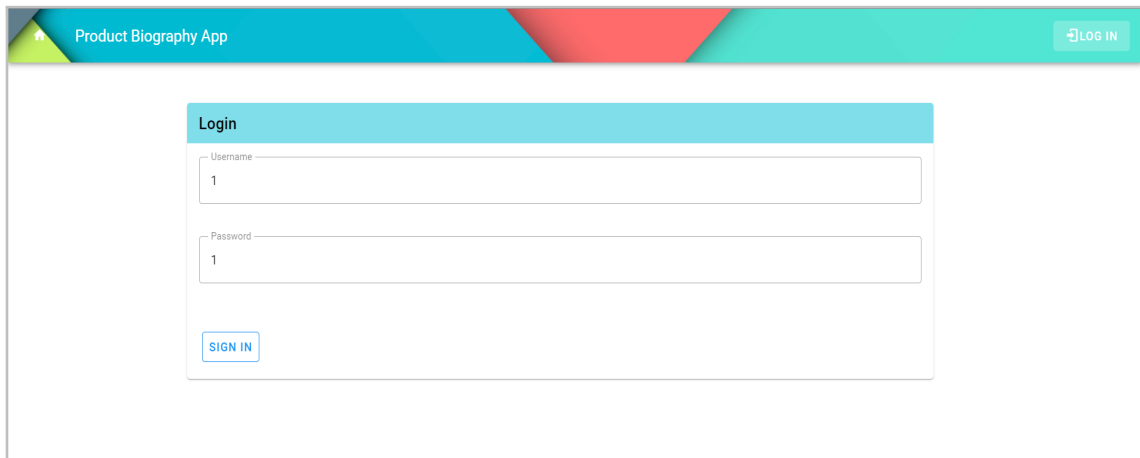


Figure 7.6: Login

After the login is verified and accepted, the user is presented with the list of products registered on the blockchain, along with a navigation drawer on the left side. The navigation drawer contains the main functionalities of the prototype, and the identification of the organization the user is representing.

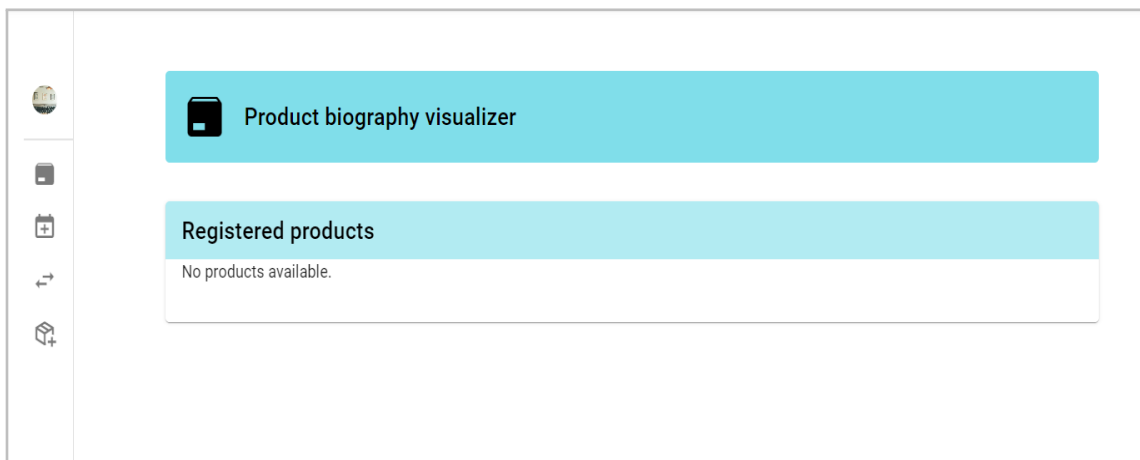


Figure 7.7: Products page

Add product

By choosing the first option on the navigation drawer, the user has the possibility of adding new products (allowed for Producers). The fields that the user must fill in to successfully create the product are as follow:

- **Name**, to identify the new product.
- **Add event**, add the first event of the product, normally its creation. If this checkbox isn't selected the product is created without the possibility of adding new events or transformations in the future.

- **Add quantity**, choose the quantity of the product, and the unit of measurement.
- **Private**, allows the user to turn the product visible to only their organization.
- **Private on collection**, allows the Raw material and Fiber producer user to store the product in the collection named "collection12", discussed in section 7.2.3.

As a result of selecting the add event checkbox, the user can also insert the event description, its location and coordinates, new attributes such as color or weight, and associate files, as shown in figure 7.8. If the product is successfully created, the user can view it in the homepage in the part related to the registered products (figure 7.7).

The screenshot shows a web form titled "Add product". The form is enclosed in a light blue border. At the top, there is a blue header bar with the text "Add product". Below the header, there are several input fields and buttons. The first input field is labeled "New product(s) name" and contains the text "Wood1". Below this is a checkbox labeled "Add event" which is checked. Underneath the checkbox are three more input fields: "Event description" containing "Creation", "Event location" containing "Coimbra", and "Latitude" containing "30". Below the "Latitude" field is a "Longitude" field containing "100". There are three buttons: "ADD ATTRIBUTE", "ADD FILE", and "ADD ATTRIBUTE". Below these buttons are three checkboxes: "Add quantity", "Private", and "Private on collection". At the bottom of the form are two buttons: "SUBMIT" and "CLOSE".

Figure 7.8: Add new product

Add product from others

An interesting and key feature for storing the life cycle of products is the possibility of creating new products, originating from existing ones. For example, take two existing fabrics and use them to create a sweater.

It shares most of the fields of the add product functionality, with the exception of the selection of products that will generate the new one. In this example, another product, named Wood2 was created, which together with Wood1, created in the previous functionality, will originate a fiber named Fiber2.

The form is titled "Add product from others" and contains the following fields and controls:

- New product(s) name:** Text input field containing "Fiber2".
- Select products:** Dropdown menu showing "Wood2, Wood1".
- Add event:** Checked checkbox.
- Event description:** Text input field containing "Creation".
- Event location:** Text input field containing "Porto".
- Latitude:** Text input field containing "50".
- Longitude:** Text input field containing "100".
- ADD ATTRIBUTE:** Button.
- ADD FILE:** Button with an upload icon.
- Add quantity:** Unchecked checkbox.
- Private:** Unchecked checkbox.
- Private on collection:** Unchecked checkbox.
- SUBMIT:** Button.
- CLOSE:** Button.

Figure 7.9: Add new product from others

Add event

When a product is created, there is the possibility of adding new events that occur throughout its life cycle. For example, in the case of a log of wood, it can be cut into smaller portions to facilitate the processes that follow. Thus, in figure 7.10, the event of cutting the wood is created, which has the same fields to fill as the addition of a new product when the add event checkbox is selected.

Add event to Wood1

Event description
Woodcutting

Event location
Coimbra

Latitude
30

Longitude
100

ADD ATTRIBUTE

ADD FILE

SUBMIT CLOSE

Figure 7.10: Add new event

Add transformation

Another action available to perform on products that have already been created is to add new transformations. These transformations refer to physical alterations of the products, in this case the use of wood to create fibers, and the change of supply-chain phase.

The fields to be filled in are all similar to those for events, with the exception of the "generate new product". This field serves to identify the products that this log of wood (product used in this example) originated.

When the transformation shown in figure 7.11 is created, the Fiber1 product is created with a first event, sharing the same characteristics as the transformation.

Add transformation to Wood1

Note: Attributes only get registered in created products.

Transformation description
Creation of fibers

Transformation location
Coimbra

Latitude
30

Longitude
100

ADD ATTRIBUTE

GENERATE NEW PRODUCT

Product name
Fiber1 REMOVE

ADD FILE

SUBMIT CLOSE

Figure 7.11: Add new transformation

Private information

To demonstrate an example of private products, two products were created with the user of the organization Raw material. The first one was private for this organization, and the second one was created and inserted in the collection shared by the Raw material and Fiber producer organizations.

Figure 7.12, shows the point of view of three different organizations to demonstrate what each one has access to. The Raw material organization has access to the two products it created, while the Fiber producer organization only has access to the product inserted in the collection. The third organization, which in this case is the Fabric manufacturing but could be any of the other ones, does not have access to any of the private products.

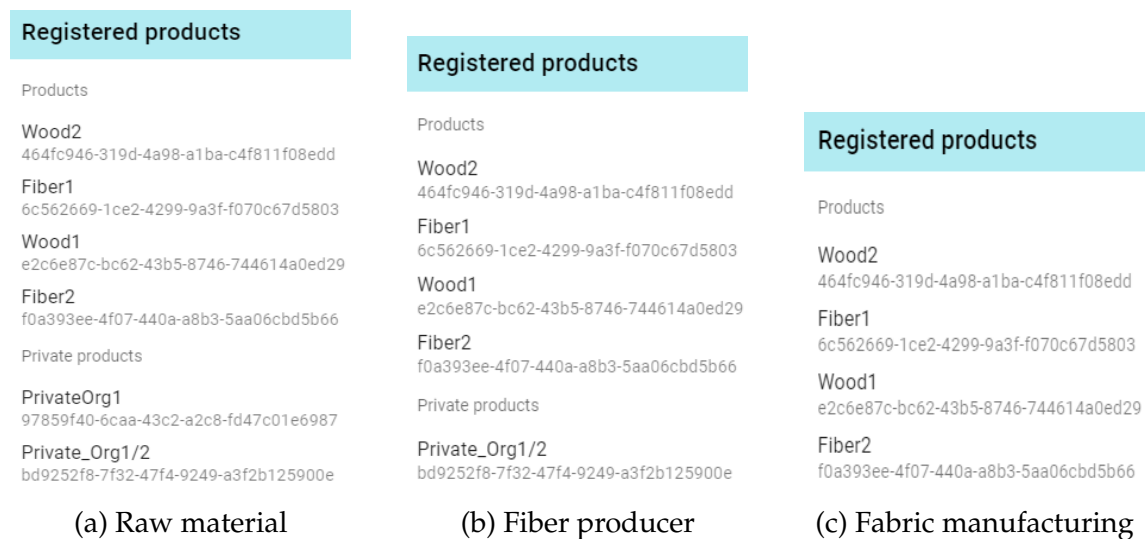


Figure 7.12: Example of storage of private products

Files

Files can be attached to products, events and transformations, and are associated with a short description in case the filenames are not quite friendly. Figure 7.13, demonstrates how to associate these files and their descriptions.

For this example, an event was created that represents the transport of fibers from one location to another. Two images were associated, one when the fibers left the starting point, and another of when they arrived.



Figure 7.13: Add files

After creating the product, event or transformation, it is possible to view the hashes of each associated file on the respective card, e.g. figure 7.14a. Be cognizant that this organization has full read access to the products.

If the user wants to view the associated files, they just need to click on the "Files" button on the event or transformation card, and a list of files will be displayed (figure 7.14b).

In the case where the user intends to download a file, they only need to click on the one intended for download. For this action to be completed successfully, there is a verification of the hashes, the one present in the blockchain and the one that is calculated with the file that the dropbox API returns.



(a) Event card example

(b) List of the files

Figure 7.14: Example of the files

Products history

To conclude this small demonstration of the functionalities, let's understand how it is possible to navigate through the life cycle of products. With the features already presented, a new product was created, Fabric1, which was originated by the products Fiber1 and Fiber2, created throughout this demonstration.

Figure 7.15, shows a first example of the life cycle of this new Fabric1 product. The existence of the first event, in this case its creation, and two buttons that allows the user to move backwards and forwards in the product's life cycle is visible.

Given that this Fabric1 product was originated by the two fibers previously created, using the "Previous phase" button, it is possible to verify that in fact the two fibers are present (figure 7.16a). On the contrary, as this product has not yet originated any products, the "Next phase" button will show an empty list.

To complement, figure 7.16b shows what appears if the user is in the Fiber2 life cycle and the "Next phase" button is clicked. When clicking on one of the products that appears on the list, the life cycle of the product that was clicked will appear.

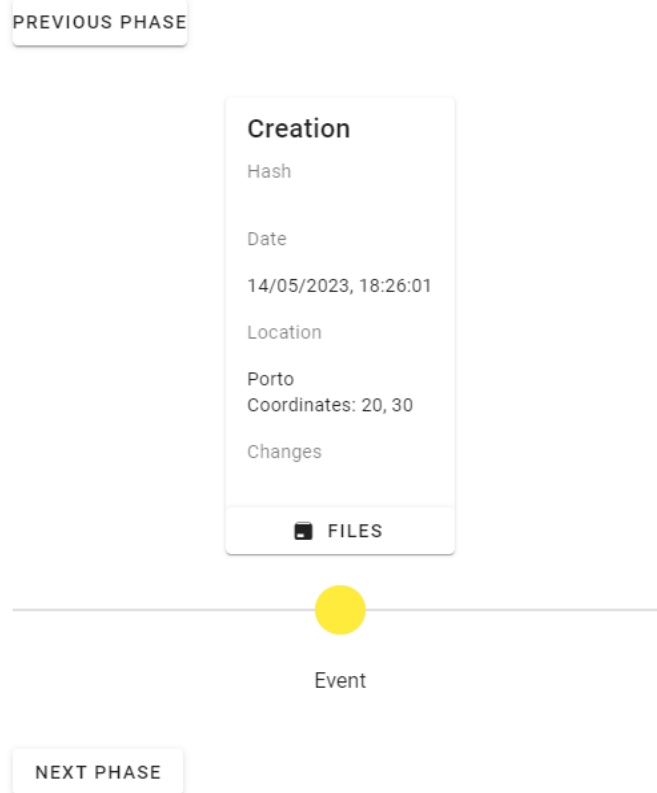


Figure 7.15: Product Fabric1 history



Figure 7.16: Example of the life cycle of products

Although only the navigation of the life cycle between two products is demonstrated, the prototype allows navigating the entire life cycle, from the raw materials phase, to the final consumer.

Figure 7.17 shows the representation of the life cycle of the product Wood1 created at the beginning of the demonstration to give a better illustration of what is presented for a product that has several events and transformations. Accordingly, it is noticeable the event of its creation, the event that was later added related to the cutting of the wood, the first transformation that originated Fiber1, and the transformation that originated Fiber2.

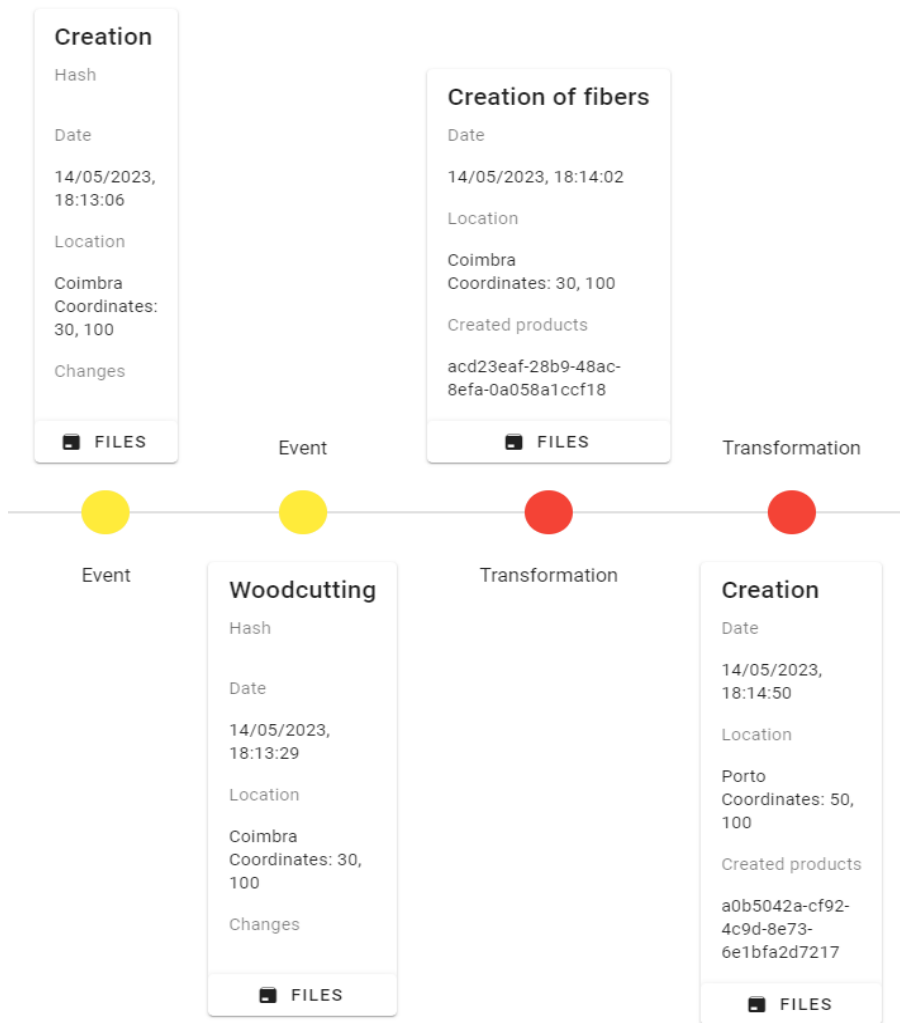


Figure 7.17: Product Wood1 history

7.3.2 Verification measures

It is necessary to ensure that the insertion of information in the blockchain is only done by organizations with write permissions, blocking this functionality for any other organization present on the network, or external users.

Therefore, first it will be demonstrated what happens internally, when a new product is added by an organization with permissions to do so. For this, a tool made available by Hyperledger Fabric is used to monitor everything that happens within the various containers that make up the network. By running the "monitordocker.sh" file, it is possible to view all the changes made to the network by any of the actors.

Figure 7.18 shows a portion of what is logged in the console when inserting a new product. Initially, it is possible to visualize the reception of the new block by all peers in the network, and its subsequent validation according to the parameters presented in section 4.4. After validation, and if everything is correct, each peer will add this new block to its ledger.

```

StoreBlock -> Received block [22] from buffer channel=mychannel
StoreBlock -> Received block [22] from buffer channel=mychannel
StoreBlock -> Received block [22] from buffer channel=mychannel
StoreBlock -> Received block [22] from buffer channel=mychannel
StoreBlock -> Received block [22] from buffer channel=mychannel
StoreBlock -> Received block [22] from buffer channel=mychannel
StoreBlock -> Received block [22] from buffer channel=mychannel
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms
[validator] Validate -> [mychannel] Validated block [22] in 0ms

```

Figure 7.18: Monitor docker output

Moving forward to network organizations without write permissions, the prototype has several preventive checks since the veracity and credibility of the data present in the blockchain is one of the main pillars to create an environment of trust among the stakeholders. Thus, if a user does not have write permissions on the blockchain, a check is made to only show the registered products, without the navigation drawer.

Also, a verification was developed in case the previous one is surpassed through the Uniform Resource Locator (URL) or another type of attack. Therefore, if an attempt is made to insert products, a check is performed, showing a message to the end user informing the lack of permissions to carry out the intended functionality.

This verification is done in the smart contract, as shown in figure 7.19, where the value of the "write" attribute, present in the user's certificate, is compared with the value "True". If the values are equal, the smart contract inserts the product, otherwise, it returns the error explained before. It should be noted that this verification is carried out for all functions related to data insertion, that is, insertion of products, events and transformations.

```

// Gets org ID to verify the permissions
function CheckPermissions(ctx){
    return ctx.clientIdentity.assertAttributeValue('write', 'True')
}

```

Figure 7.19: Smart contract permissions function

To make the prototype more robust to malicious insertion attempts, the channel policies were changed. As discussed in section 4.4, policies are rules that can be established within the network. The fabric network has a default reader policy that allows to control which organizations can send block events in the blockchain (e.g., insert products, events, and transformations). Thus, if a user belongs to an organization that doesn't fulfill the policy, an error appears, informing that it was not possible to satisfy the policy in question. Figure 7.20, identifies the id of the proposed transaction, which is logged and can be used for auditing purposes.

```
details: 'Failed evaluating policy on signed data during check policy on channel [mychannel] with policy [/channel/Application/Readers]: [signature set did not satisfy policy]',
metadata: Metadata { internalRepr: [Map], options: {} }
},
transactionId: '6071dad5d2f6817c9e418a26de79243e5d20f9d46103b07e89230e9aec8980e8'
```

Figure 7.20: Reader policy permission denied

To prevent information from being inserted by users who have not logged in (access the home page through the URL), a check is made that results in a blank page. Still, all other checks are performed if, for some reason, the user manages to insert information through the front end.

Continuing, and taking special attention to data access, to prevent unauthorized organizations from having access to the data, the write policy, which is already defined automatically by Hyperledger Fabric, was changed. This policy defines which organizations can invoke the chaincode, preventing users without permissions from interacting with prototype. It is possible to see in figure 7.21 what is displayed in the console when a network organization without those permissions tries to invoke the chaincode.

On the other hand, by accessing via URL, the user is not associated with any organization, resulting in a message similar to the one shown in figure 7.21.

```
details: [
  {
    address: 'peer0.org2.example.com:9051',
    message: 'Failed evaluating policy on signed data during check policy on channel [mychannel] with policy [/channel/Application/Writers]: [signature set did not satisfy policy]',
    mspId: 'Org2MSP'
  }
],
```

Figure 7.21: Writer policy permission denied

One of the main goals of the development was to ensure that there were different permissions in relation to the type of data that each organization has access to. It was already demonstrated how this distinction is made, and therefore, now the intention is to show the verification that achieves it.

Although Hyperledger Fabric allows the definition of the read and write permissions of the network and of the respective channels, it does not allow the definition of the type of access that each user has to the data (e.g., full access or event only). Still, it would be possible to make this distinction by creating a channel where only the relevant information for each type of permission would be stored, e.g., a channel that would store all the data, from which Producers would

read, a second channel where the data would be stored relevant to the Certifying Entity organization, and one relating to the Final Consumer organization. The problem is that it is not practical to be tripling the information and overloading the network with so much redundant data.

So, the decision was to do the permissions check on the front end. Each organization has an associated type of permission when the certificate and key generation request is made. This attribute is requested to the smart contract and is used to decide which data to present to the user (figure 7.22).

```
<!-- Timeline for permissions of type 1 and 2 -->
<v-timeline class="my-4" direction="horizontal" v-if="productDetails && permission !=3">...
</v-timeline>
<!-- Timeline for permissions of type 3 -->
<v-timeline class="my-4" direction="horizontal" v-if="productDetails && permission ==3">...
</v-timeline>
```

Figure 7.22: Front end code for permissions distinction

Since the previously verification distinguishes Producers and the Certifying Entity organization from the Final Consumer organization, the verification included in figure 7.23, distinguishes Producers from the Certifying Entity organization. The verification made is to only show the images to users with all permissions.

```
<v-btn
  v-show = "permission == 1"
  block
  prepend-icon="mdi-package"
  color="blue-lighten-7"
  @click="getFiles('E', ev)">
  Files
</v-btn>
```

Figure 7.23: Front end code for documents permissions

7.3.3 Real use case demonstration

To finish this chapter referring to the development of the prototype, and to evaluate the solution with a real example of what the life cycle of a textile product looks like (figure 7.24), a real use case will be presented. Given that the case company is responsible for the production of paper pulp and textile fibers, this use case has a greater emphasis on the fiber production phase.

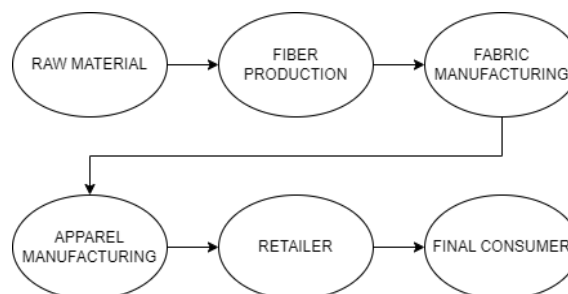


Figure 7.24: Textile life cycle, adapted from [Eckhardt, 2011]

Before moving on to the use case demonstration, it is necessary to explain some concepts specific to the textile sector:

- **Forest Stewardship Council (FSC)**, is an organization that aims to promote environmentally beneficial forest management, issuing certificates that guarantee that products come from well-managed forests [FSC, 2023].
- **Elemental chlorine free (ECF)**, is a technique that uses chlorine dioxide for the bleaching of wood pulp.
- **Lyocell**, is a semi-synthetic fiber used for apparel production and obtained through the dissolution of pulp.

Figure 7.25, identifies the products that are part of a sweater's history. Each of these products will be analyzed in detail, visualizing its life cycle and perceiving the process since the early stages of planting a tree to creating the sweater. This example does not in any way represent all the steps in the lifecycle of the products, it only serves to demonstrate some of them using their correct names, evaluating to a certain extent the functioning of the prototype.

Thus, the sequence of products is as follows, (1) Eucalyptus with bark, (2) Eucalyptus ECF, (3) Lyocell fiber, (4) Lyocell fabric, and (5) Lyocell sweater.

Registered products	
Products	
LYOCELL FIBER	53a1d6a2-6874-4c0f-8acc-92f93ea9ce8c
LYOCELL FABRIC	77ede3dc-2dc8-4653-b647-40dd523a810f
LYOCELL SWEATER	7d92a5c9-eadd-4227-9083-d86f6d0ff419
EUCALYPTUS ECF	ba043dd7-1593-4458-80d7-fc2d896e25a2
EUCALYPTUS WITH BARK	e6127e4b-dbef-4030-af27-4ca2750a714d

Figure 7.25: Created products

Raw material

Starting with the first phase, referring to the Raw material organization, there is the product Eucalyptus with bark containing a first event related to its planting. After its planting, the wood certification is carried out, associating the certificate in the files, and defining the date on which this FSC certificate was issued. From this moment on, all blockchain participants with full permissions can view the associated certificate.

Subsequently, the eucalyptus is cut into several pieces and is sent to the fiber production factory. When the arrival at the factory is validated (by the fiber production factory), the factory will register a new product, Eucalyptus ECF, automatically causing a transformation in the Eucalyptus with bark life cycle, referring to

the arrival. Here, it is evident that the transformations do not always refer to the physical alteration of the products, but to the phase change as well.

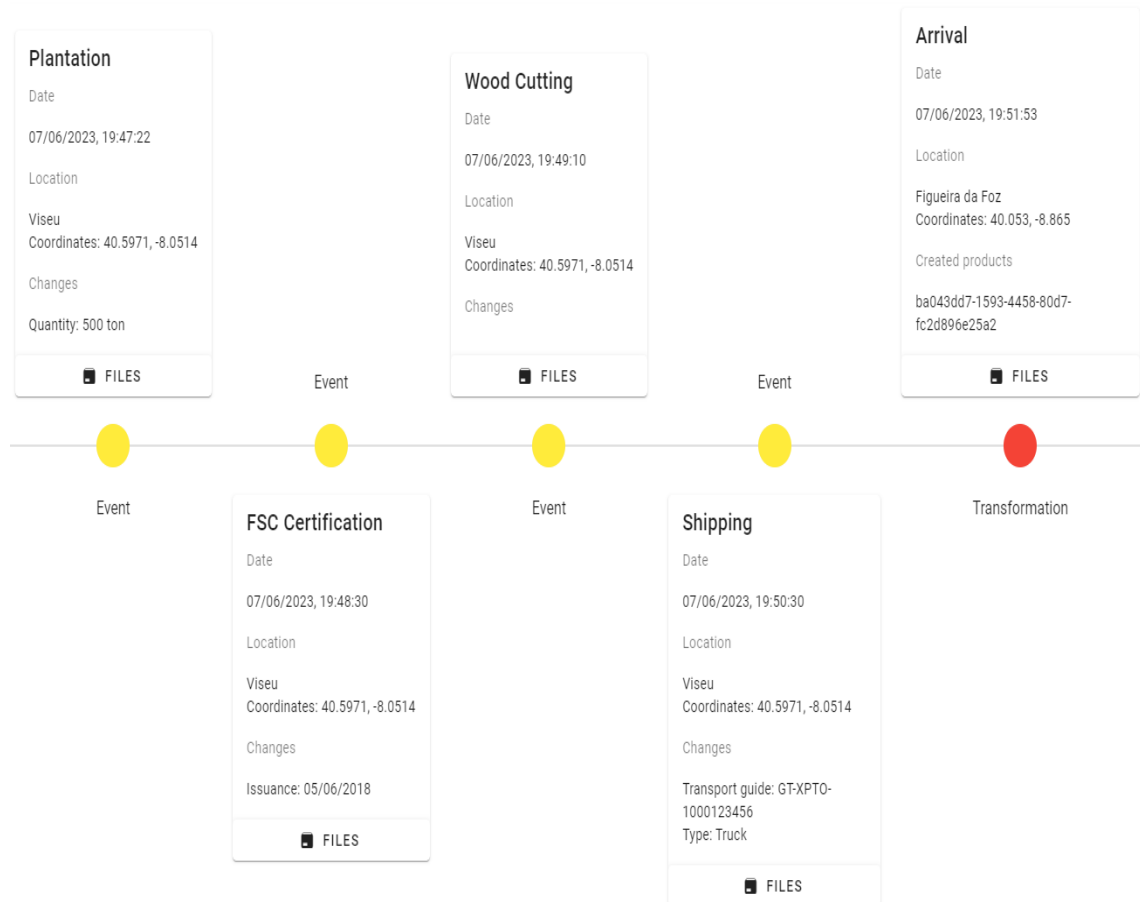


Figure 7.26: Eucalyptus with bark

Fiber producer

It is important to be aware that just because the new product is called Eucalyptus ECF, which is a pulp, does not mean that the product is already in that state. When the eucalyptus pieces arrive, and as they are under the responsibility of the fiber production company, there is a need to create the new product so that the life cycles are independent.

Upon arrival, the eucalyptus pieces are subject to the ECF process, where a date is defined in addition to the one automatically registered when creating the events. This date is to allude to the idea that the creation of events is not always carried out precisely at the time of the process, that is, the eucalyptus pieces can start the ECF process but the creation of the event is only done later manually by a company employee. When the process is finished, and represented as ECF Process Output (figure 7.27), the pieces of eucalyptus become pulp.

With the pulp created, it is necessary to transform it into textile fibers, hence the need to feed this pulp to the production lines. Here, as there will be a physical change in the product, a new one is created, called Lyocell fiber.

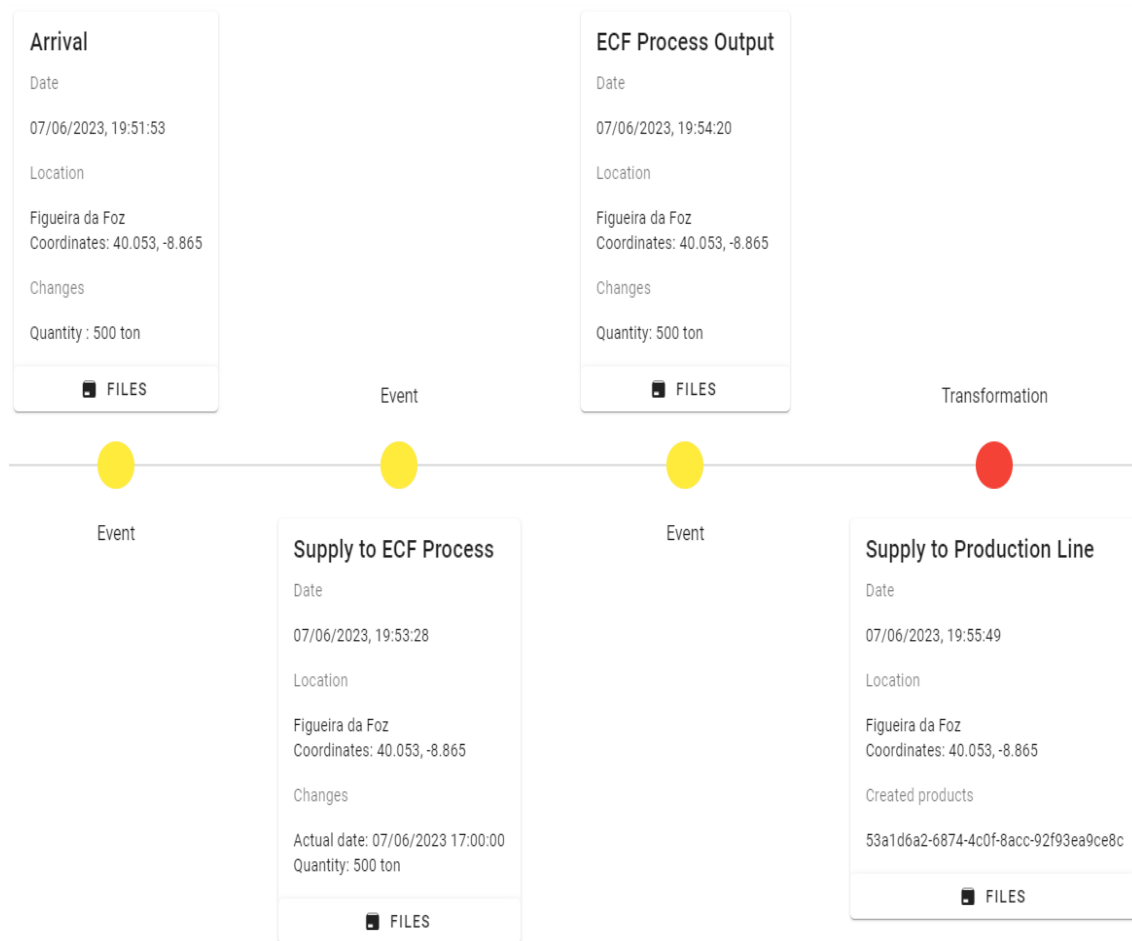


Figure 7.27: Eucalyptus ECF

Given that the company that has accompanied the development of this prototype is in charge of the production of pulp, and textile fibers, there is no phase change, that is, both the pulp (Eucalyptus ECF) and the fiber (Lyocell fiber) are represented in the fiber production phase. This possibility of having several products within the same phase is possible for the Fiber producer organization, as for the rest, indirectly creating sub phases within the main phases.

Therefore, after leaving the production line, the pulp is now in the form of a lyocell fiber that can be used for fabric production. Subsequently, these fibers are sent to the fabric factory, where it is possible to define the transport guide, and the type of transport, e.g., truck, plane, boat. These attributes serve only as an example, meaning that there is the possibility to define a variety of relevant attributes for the events and transformations.

Fabric manufacturing

In the same way that was carried out when sending the pieces of eucalyptus, a transformation is created that represents the arrival of the fibers at the fabric production factory. Upon arrival, the fibers are fed to the production line, which will originate the lyocell fabrics. Once production is complete, the lyocell fabrics are sent to the apparel production factory.

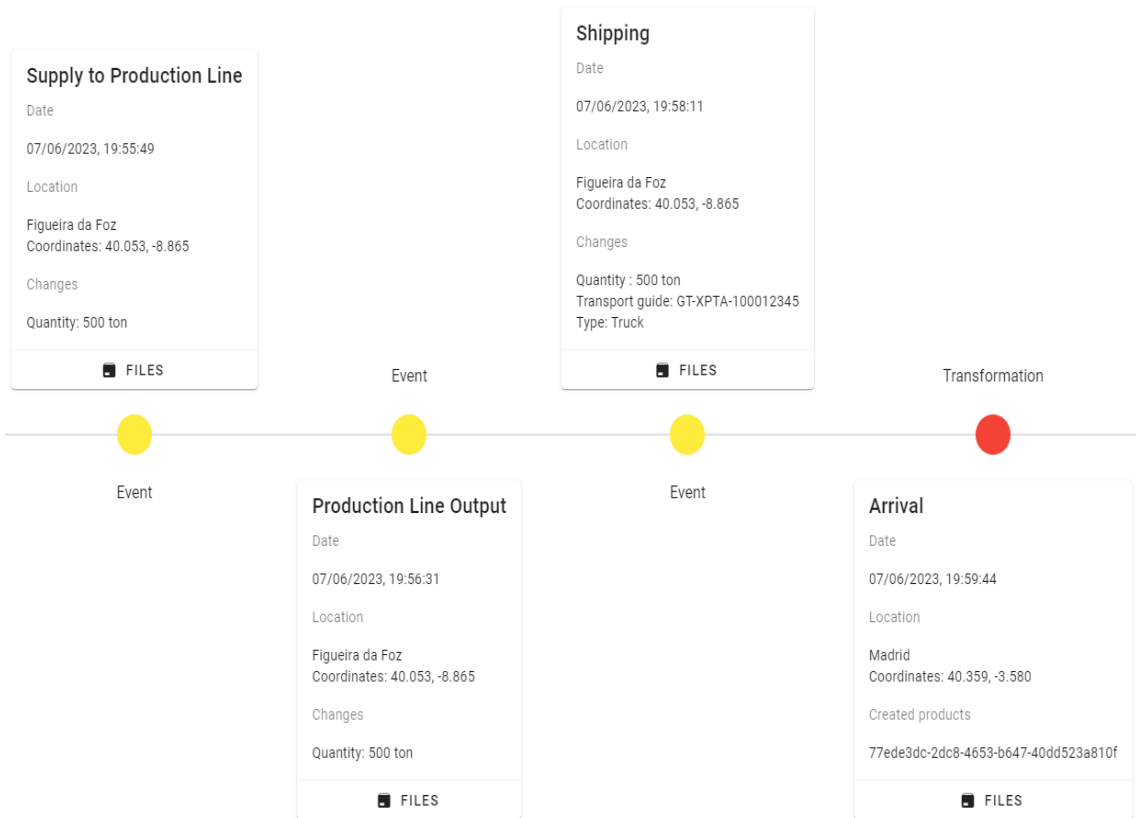


Figure 7.28: Lyocell fiber

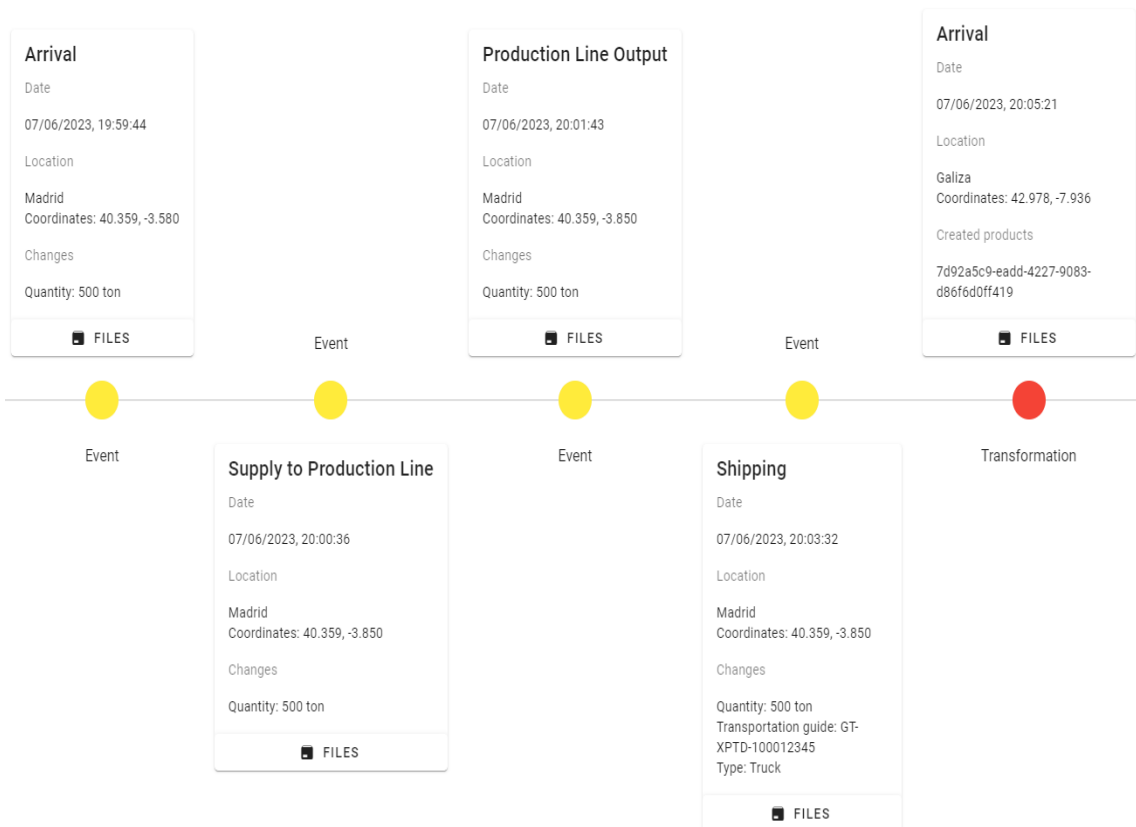


Figure 7.29: Lyocell fabric

Apparel manufacturing

To finish off this use case, comes the apparel production phase, where lyocell fabrics are received and used to create sweaters (figure 7.30). From this point onwards, these garments can be shipped to retailers, who in turn will sell them, thus, continuing the product life cycle. It is visible that the product life cycle is accessible to all stakeholders through a simple front end, meeting the traceability and transparency needs that the textile sector is looking for.

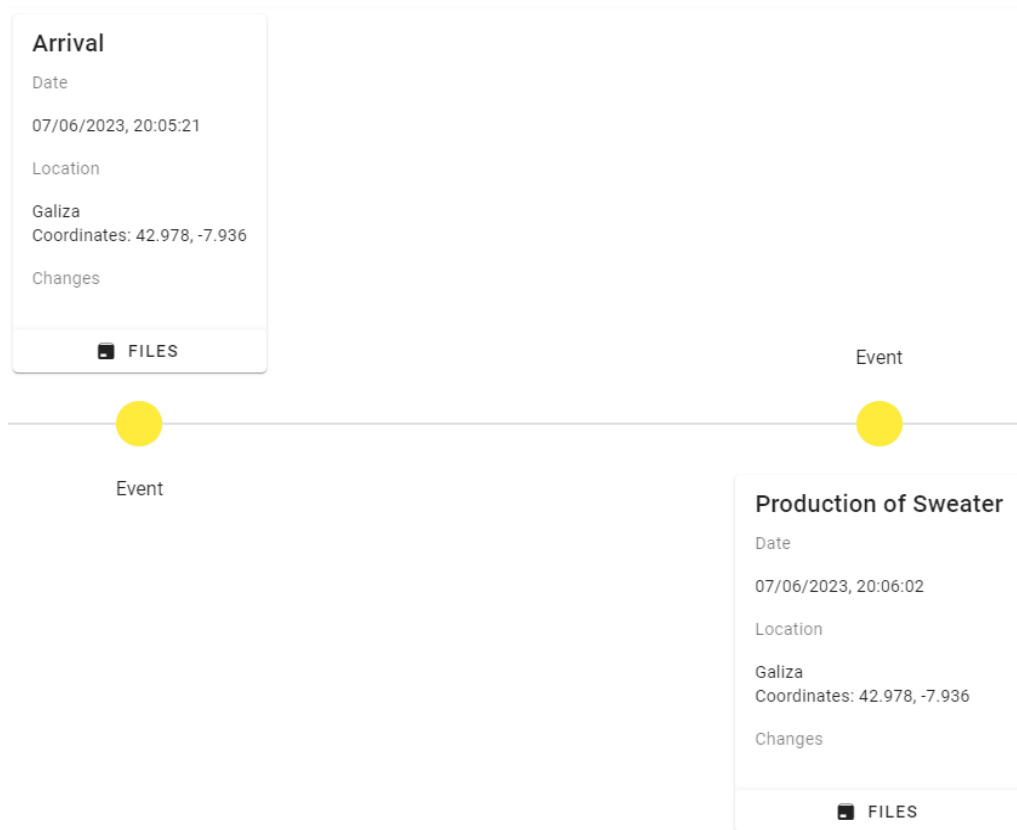


Figure 7.30: Lyocell sweater

7.4 Summary

Section 7.1 started by presenting the status of the prototype developed by [Wrisez, 2023], which had already developed a platform where it was possible to register information of products. Next, it was provided the prerequisites to be able to set up, and interact with the platform. To finish the section, it was presented the necessary steps to place all the organizations represented in chapter 5 in the Hyperledger network and install the chaincode in each of the peers.

Later, in section 7.2, some steps performed during the development were explained. It was necessary to change the data structure defined by [Wrisez, 2023] to achieve the defined requirements. Following, it was shown the permissions assigned for each of the organizations, and how the storage of files and private information works.

Finally, in section 7.3, all the platform's functionalities were first presented following the role of a user of the Raw material organization. Then, the verification measures implemented to guarantee the integrity and veracity of the data were explained, and finally, a real use case of the textile sector, created together with the case company, was demonstrated.

Chapter 8

Evaluation and testing

After displaying and explaining the functionalities of the developed prototype, this chapter starts by evaluating the requirements defined in chapter 5. Next, follows the definition of the test cases, along with the expected results, and if the prototype passed or failed those tests. Finally, this chapter ends with a brief summary.

8.1 Evaluation

The prototype's functionalities were evaluated together with the company that has been monitoring the development of the thesis when the use case presented in chapter 7 was created. The feedback received was very positive, given that the state of the prototype responded to the concerns and needs received from the company throughout the development of the thesis.

The security of the prototype was also evaluated in relation to problems associated with actors' certificates. Certificates function as a hierarchical tree, with the root certificate corresponding to the Fabric CA, which is public. Subsequently, each organization has its own CA with a certificate signed by the root CA, where, finally, each admin, peer and user of the organization have their certificates signed by the organization's CA.

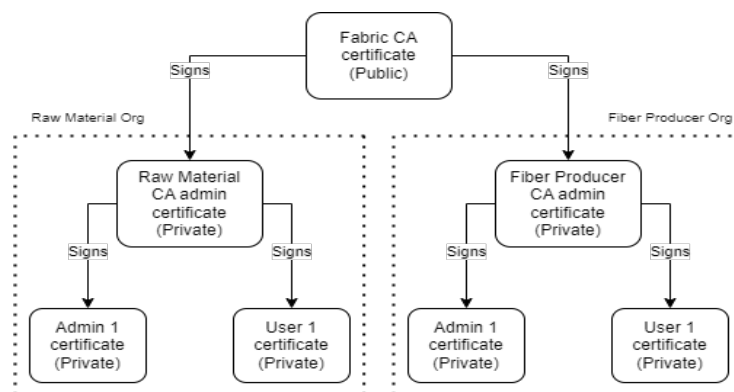


Figure 8.1: Hierarchical tree example

Image 8.2 shows an example of a directory with an organization's certificates. In more detail, it is visible the "ca" folder that contains the organization's CA certificate, and the "users" folder that contains sub-folders referring to the actors present in the organization. Focusing on the user's sub-folder we can verify the presence of the certificate and the private key.

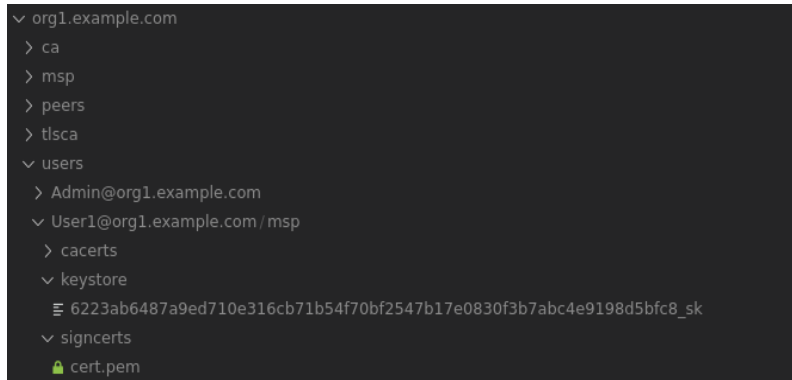


Figure 8.2: Directory with the certificates and private keys

Since certificates and private keys are the main components for an actor to be able to interact with the prototype, it was necessary to ensure that when one of these was changed, deleted, or even revoked, the actors in question would lose all their permissions to guarantee the integrity of the data. However, despite the certificate of a user, or admin, being revoked, changed, or deleted, it must be possible to generate new certificates and private keys so that these actors can regain access to the permissions they are entitled to.

Therefore, using the functionalities provided by Hyperledger Fabric it was necessary to evaluate the generation of the new certificates for users, admins, and for the CA admin. Be aware that when generating a certificate for the CA admin, there will be the need to generate certificates for all users, admins, and peers of the organization.

Thus, when it becomes necessary to generate new certificates for a new user, the following commands are used:

- `fabric-ca-client register -caname ca-org1 -id.name user2 -id.secret user2pw -id.type client -id.attrs 'write=True:ecert,read=1:ecert' -tls.certfiles "$PWD/organizations/fabric-ca/org1/ca-cert.pem"`.
- `fabric-ca-client enroll -u https://user2:user2pw@localhost:7054 -caname ca-org1 -M "$PWD/organizations/peerOrganizations/org1.example.com/users/User1@org1.example.com/msp" -tls.certfiles "$PWD/organizations/fabric-ca/org1/ca-cert.pem"`.

The first one refers to the request for registration of new entities, passing the attributes of writing and reading permission, and the username. The second one is the subsequent addition of the entity to the network. After the enrollment is done, it is still necessary to update the channel configurations, so as to recognize the new user.

When the enrollment request for the new user is made, the path of the previous one is chosen, avoiding the need to change the backend of the client application, which needs the path of the certificates and of the private key for chaincode invocation. Obviously this does not eliminate the possibility of choosing another path and changing the backend to have two users represented by the same organization.

In the case of an admin, the process is similar, being only necessary to change the type of actor, visible in the registration request command, the username, and the password. After updating the channel configuration, it is then possible to use the new admin to interact with the network.

Finally, the situation becomes more complex when there is a need to generate a new certificate for the organization's CA admin. There are two possibilities for generating certificates, a simpler one, and a more complex one. The first one uses the file presented in section 7, which generates and adds organizations to the network and channel, e.g., the "addOrg3.sh" file. The second one consists of generating all the certificates, that is, the CA admin, admin, and user, manually.

A similarity that both possibilities share is the need to change the docker-compose file that originates the CA admin certificate. This is because when an identity is revoked, the username of the identity can no longer be used for the registration of another actor.

Figure 8.3, shows the docker-compose configuration file that starts the Fabric Manufacturing organization's CA container. As visible, in the part relating to the command, it is necessary to change the username to admin2, with this name being just an example.

```
services:
  ca_org3:
    image: hyperledger/fabric-ca:latest
    labels:
      service: hyperledger-fabric
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-org3
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=11054
    ports:
      - "11054:11054"
    command: sh -c 'fabric-ca-server start -b admin2:admin2pw -d'
    volumes:
      - ../fabric-ca/org3:/etc/hyperledger/fabric-ca-server
    container_name: ca_org3
```

Figure 8.3: Docker compose CA configuration file

After changing the docker-compose file, it is still necessary to change the "registerEnroll.sh" file that enrolls and registers the CA admin, and subsequently enrolls an admin and a user on the network, all of which are represented by the Fabric Manufacturing organization.

From this point on, there is an admin CA with a valid certificate, and therefore, we

can proceed with the registration and enrollment of the admin of the organization and the user who will interact with the client application.

By verifying the result of these changes in the prototype (the revocation of the certificates and posterior generation of new ones), it was concluded that the actors are able to recover access to their data and permissions.

8.2 Testing

In this section, follows the definition of the test cases, along with the prerequisites, the list of steps to run them, the expected result, and whether the prototype can achieve it or not.

Before moving forward, and to avoid redundant information, as a prerequisite it is always necessary that the user logs in as a Producer, except in cases where it is specifically said which organization has to log in to run the test case.

Therefore, table 8.1 represents the tests cases carried out for the functional requirements related to the insertion of information in the blockchain. Next, in table 8.2, the test cases relating to information access are represented, and finally, table 8.3 shows the test cases carried out on non-functional requirements.

ID	Test case	Test pre-requisites	Test step	Expected results	Pass /Fail
TI1	Insert new product	None	1. Fill functionality "Add product" fields 2. Click submit	The product is inserted in the blockchain and displayed in the front end	Pass
TI2	Add new event to a product	A product must be registered in the blockchain	1. Fill functionality "Add event" fields 2. Click submit	The event is associated to the product, inserted in the blockchain, and displayed in the front end	Pass
TI3	Add new transformation to a product	A product must be registered in the blockchain	1. Fill functionality "Add transformation" fields 2. Click submit	The transformation is associated to the product, inserted in the blockchain, and displayed in the front end	Pass

TI4	Generate a new product when adding a new transformation to an existent product	A product must be registered in the blockchain	<ol style="list-style-type: none"> 1. Fill functionality "Add transformation" fields 2. Select "generate new product" and choose a name for the product 3. Click submit 	The transformation is associated to the product and inserted in the blockchain, the new product is generated, and both are displayed in the front end	Pass
TI5	Insert new product originated from one or more existent in the blockchain	A product must be registered in the blockchain	<ol style="list-style-type: none"> 1. Fill functionality "Add product from others" fields 2. Select the products that will originated the new one 3. Click submit 	The product is inserted in the blockchain and displayed in the front end	Pass
TI6	Associate a file with an event	A product must be registered in the blockchain	<ol style="list-style-type: none"> 1. Fill functionality "Add event" fields 2. Select "Add file" 3. Choose files to associate and fill description 4. Click submit 	The event is associated to the product and inserted in the blockchain with the file name, description, and hash. The file is stored in the cloud and locally	Pass
TI7	Associate a file with a transformation	A product must be registered in the blockchain	<ol style="list-style-type: none"> 1. Fill functionality "Add transformation" fields 2. Select "Add file" 3. Choose files to associate and fill description 4. Click submit 	The transformation is associated to the product and inserted in the blockchain with the file name, description, and hash. The file is stored in the cloud and locally	Pass

TI8	Insert private data in the organization personal collection	None	1. Fill functionality "Add product" fields 2. Select "Private" 3. Click submit	The product is stored in the organization personal collection, being visible only to itself	Pass
TI9	Insert private data in the Raw material and Fiber producer collection	None	1. Login as Raw material or Fiber producer user 2. Fill functionality "Add product" fields 3. Select "Private on collection" 4. Click submit	The product is stored in the Raw material and Fiber producer organizations collection, being visible only to themselves	Pass

Table 8.1: Test cases of data insertion related functional requirements

ID	Test case	Test prerequisites	Test step	Expected results	Pass /Fail
TA1	Access to all products, events, transformations and associated files	A product must be registered in the blockchain with at least one event, transformation and associated file	1. Select a product 2. Check events and transformations accessibility 3. Select "Files" button 4. Check files accessibility	The user is presented with the events and transformations of the product, also having access to the associated files	Pass
TA2	Download of an associated file	A product must be registered in the blockchain with at least one associated file	1. Select a product 2. Select "Files" button on the event or transformation card 3. Click on a file	The file is downloaded to the user device	Pass

TA3	Only access to restricted fields of the events	A product must be registered in the blockchain with at least one event, transformation, and associated file	<ol style="list-style-type: none"> 1. Login as Final consumer 2. Select a product 3. Check access to event fields 	The user is only presented with restricted fields of the events, not having access to transformations or associated files	Pass
TA4	Only access to products, events, and transformations	A product must be registered in the blockchain with at least one event, transformation, and associated file	<ol style="list-style-type: none"> 1. Login as Certifying entity 2. Select a product 3. Check access to products, events, and transformations 	The user is only presented with products, events, and transformations, not having access to associated files	Pass
TA5	Log in with correct credentials	None	<ol style="list-style-type: none"> 1. Input credentials 2. Click "log in" 	The page with the registered products and various functionalities is presented	Pass
TA6	Log in with incorrect credentials	None	<ol style="list-style-type: none"> 1. Input credentials 2. Click "log in" 	An error appears indicating that the credentials are incorrect	Pass
TA7	Navigate through the products life cycle	There must be at least two products connected with each other	<ol style="list-style-type: none"> 1. Select a product 2. Click "next phase" to move forwards on the products life cycle 3. Click "previous phase" to move backwards on the products life cycle 	The different products life cycles are shown	Pass

Table 8.2: Test cases of data access related functional requirements

ID	Test case	Test prerequisites	Test step	Expected results	Pass /Fail
TN1	Check data insertion when logged with an organization without write permissions	None	<ol style="list-style-type: none"> 1. Log in as Certifying Entity 2. Fill functionality "Add product" fields 3. Click submit 4. Check result 	An error appears indicating that the user doesn't have the permissions for such functionality	Pass
TN2	Check data insertion without logging in	None	<ol style="list-style-type: none"> 1. Navigate to the home page via the URL 2. Fill functionality "Add product" fields 3. Click submit 4. Check result 	The product is not inserted and an error is logged	Pass
TN3	Check data access without logging in	A product must be registered in the blockchain	<ol style="list-style-type: none"> 1. Navigate to the home page via the URL 2. Check access to data 	A blank page is shown and an error logged	Pass
TN4	Check data insertion when user certificate is deleted, changed or revoked	None	<ol style="list-style-type: none"> 1. Fill functionality "Add product" fields 2. Click submit 	The product is not inserted and an error is logged	Pass
TN5	Check data insertion when user private key is deleted or changed	None	<ol style="list-style-type: none"> 1. Fill functionality "Add product" fields 2. Click submit 	The product is not inserted and an error is logged	Pass
TN6	Check data insertion when a CA admin certificate is deleted, changed or revoked	None	<ol style="list-style-type: none"> 1. Fill functionality "Add product" fields 2. Click submit 	The product is not inserted and an error is logged	Pass

TN7	Check data access when user certificate is deleted, changed or revoked	A product must be registered in the blockchain	1. Log in as any organization 2. Check data access	The data is not shown and an error appears	Pass
TN8	Check data access when user private key is deleted or changed	A product must be registered in the blockchain	1. Log in as any organization 2. Check data access	The data is not shown and an error appears	Pass
TN9	Check data access when a CA admin certificate is deleted, changed or revoked	A product must be registered in the blockchain	1. Log in as any organization 2. Check data access	The data is not shown and an error appears	Pass

Table 8.3: Test cases of non-functional requirements

8.3 Summary

Throughout this chapter, it was firstly possible to evaluate the response of the prototype to the needs and concerns of the case company, and the DPP's. Also, that in the event of a change, deletion, or revocation of a certificate, it is possible to provide new certificates to actors so that they can return to act in the network. This was very important for this phase of the process, since blockchain has not yet been cemented as one of the optimal solutions for storing the life cycle of products, be they from the textile sector, as from any other sector.

Next, followed the execution of the test cases, where it was possible to understand that the prototype manages to fulfill all the requirements defined in chapter 5. As this chapter comes to an end, in the next one follows the presentation of the conclusions of this thesis along with future steps.

Chapter 9

Conclusions

This thesis aimed to develop a prototype to support a DPP in the textile industry with blockchain as foundation, and to provide a proof of concept of a DPP in the same industry.

Due to the novelty of this topic, it was necessary to carry out several studies to understand the status of these digital passports, a priority for the CE in the EU, still lacking practical examples to guide the industry. Notwithstanding the very recent and important proposals for the integration of DPPs in the textile industry [Jansen et al., 2022], this thesis sought to test a solution in which blockchain technology emerges as the primary source of storage for these passports.

Blockchain, being a type of DLT offers many benefits for storing DPP's, such as immutability, transparency, security, and privacy of data. After a re-analysis, it was concluded that the decision taken by [Wrisez, 2023] to adopt Hyperledger Fabric was still valid for our use case.

The development process and subsequent validation showed that blockchain is a promising solution for storing DPP's in the textile sector, as it guarantees data integrity and security, preventing the insertion of invalid information from malicious users. Derived from these values that the blockchain guarantees, the problem of lack of transparency along the supply-chain is improved, having as a strong point the construction of an environment of trust between all the entities of the life cycle of the textile product.

Furthermore, by creating a prototype where it is possible to link products of different types (e.g., linking wood to fibers, fibers to pulp, pulp to fabrics, and so on), keep an history of the events, transformations and files of each of the products, restrict write and read access to the products information, allow for verification of products compliance with legal obligations, and allow users to make more informed purchase decisions, it was possible to respond to the expectations of what a DPP is in the textile sector. Due to the prototype's ability to support these and other functionalities, the feedback received by the case company was very positive.

Future steps

For any solution found and put into practice, there is always a rising number of new problems, and this thesis is no exception. So, to end this chapter, possible future steps will be presented that can complement the work developed during this thesis.

The impracticality of verifying the information that each organization is inserting in the blockchain (e.g., a user of the Raw Material organization can insert a new product representative of the fiber production phase) is a limitation of the prototype. It would be necessary to limit the options of all open writing fields, presenting the user with a dropdown of previously defined choices. Even so, it would be necessary to verify which organization the user would be representing to validate the choices.

The adoption of multi-blockchains can contribute to create new solutions to address this limitation [Ahmed balaghi, 2021]. A multi-blockchain system would consist of each peer having its own blockchain, where each blockchain would communicate with each other whenever needed.

Two more possible steps, which were not developed in this thesis, are the representation of the recycling and reuse phases, which would reintroduce the products into the economy, and the development of an application that through a QR code could access the prototype. However, despite not having been developed, the addition of these two functionalities does not imply changes in the core of the prototype.

Since this thesis and the conclusions were focused on the textile sector, emerges the possibility of testing the prototype architecture in other sectors.

Finally, and with the presentation of a proof of concept of a DPP, it is possible to specify which processes and attributes have to be defined when adding events and transformations to the life cycle of the products.

References

- Adisorn Thomas, Tholen Lena, Götz Thomas.* Towards a Digital Product Passport Fit for Contributing to a Circular Economy // *Energies* 2021, Vol. 14, Page 2289. 4 2021. 14. 2289.
- AgileBusiness .* Chapter 10: MoSCoW Prioritisation. 2022.
- Agrawal Tarun Kumar, Kumar Vijay, Pal Rudrajeet, Wang Lichuan, Chen Yan.* Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry // *Computers Industrial Engineering.* 4 2021. 154. 107130.
- Ahmed Al balaghi.* A multichain approach is the future of the blockchain industry. 2021.
- Androulaki Elli, Barger Artem, Bortnikov Vita, Muralidharan Srinivasan, Cachin Christian, Christidis Konstantinos, Caro Angelo De, Enyeart David, Murthy Chet, Ferris Christopher, Laventman Gennady, Manevich Yacov, Nguyen Binh, Sethi Manish, Singh Gari, Smith Keith, Sorniotti Alessandro, Stathakopoulou Chrysoula, Vukolić Marko, Cocco Sharon Weed, Yellick Jason.* Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains // *Proceedings of the 13th EuroSys Conference, EuroSys 2018.* 4 2018. 2018-January.
- Armin Ibitz.* Digital Product Passports for a Low-Carbon Circular Economy? 2021.
- Barata Joao, Pereira Vasco, Coelho Miguel.* Product Biography Information System: A Lifecycle Approach to Digital Twins // *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics.* 10 2020. 2020-October. 899–904.
- Billon .* The Unified Enterprise DLT System. 2023.
- Brown Phil.* A sustainable future: Using blockchain for digital product passports - Ledger Insights - blockchain for enterprise. 2022.
- Builtin .* What Is the Internet of Things? How Does IoT Work? | Built In. 2022.
- Bybit .* Consortium Blockchain | Bybit Learn. 2022.
- C4Model .* The C4 model for visualising software architecture. 2022.
- CIRPASS .* CIRPASS – Digital Product Passport. 2023.

Cem Dilmegani. Blockchain in Supply Chain: Benefits Top Use Cases in 2023. 2023.

Chopova-Leprêtre Paulina, Montfort Jean-Philippe, Johnson Kismet. EU Proposal on New Ecodesign Requirements for Sustainable Products | Perspectives Events | Mayer Brown. 2023.

Circularise . Digital product passports (DPP): what, how, and why? 2022.

Circularise . Achieving visibility into the Porsche supply chain. 2023.

Commission European. EU Strategy for Sustainable and Circular Textiles. 2022a.

Commission European. Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. 2022b.

Commission European. Funding tenders. 2023.

Cura Kirsti, Jain Sheenam, Niinimäki Kirsi. Transparency and traceability in the textile value chain. 2022.

DIGITALEUROPE . Digital Product Passport (CIRPASS) - DIGITALEUROPE. 2023.

Dasaklis Thomas K., Voutsinas Theodore G., Tsoufias Giannis T., Casino Fran. A Systematic Literature Review of Blockchain-Enabled Supply Chain Traceability Implementations // Sustainability 2022, Vol. 14, Page 2439. 2022. 14. 2439.

Docker . Docker Compose overview | Docker Documentation. 2023.

Dropbox . HTTP - Developers - Dropbox. 2023.

ECOLabel . EU Ecolabel paper products. 2022.

Eckhardt Jappe. Taking a Firm Stance: The Political Economy of Business Lobbying in EU Trade Policy towards China. 2011.

Ellen MacArthur. What is a circular economy? | Ellen MacArthur Foundation. 2022.

Ethereum . Home | ethereum.org. 2022.

Express . Express - Node.js web application framework. 2023.

FSC . Home Page Portugal | Forest Stewardship Council. 2023.

GS1 . Home | gs1.eu. 2022.

GS1 . GS1 | The Global Language of Business. 2023.

Gliffy . What is a C4 Model? How to Make C4 Software Architecture Diagrams | Gliffy by Perforce. 2021.

Guinard Dominique. A Decentralized Blueprint for Digital Product Passports. 2023.

- Hader Manal, Tchoffa David, Mhamedi Abderrahman El, Ghodous Parisa, Dolgui Alexandre, Abouabdellah Abdellah.* Applying integrated Blockchain and Big Data technologies to improve supply chain traceability and information sharing in the textile sector // *Journal of Industrial Information Integration.* 7 2022. 28.
- Haegglblom Jonna, Palmer Cecilia.* CIRCULARITY.ID. 2019.
- Hasib Anwar.* Hyperledger Sawtooth Vs. Fabric: How Are They Different? 2021.
- Hyperledger .* A Blockchain Platform for the Enterprise — hyperledger-fabricdocs main documentation. 2022a.
- Hyperledger .* How Walmart brought unprecedented transparency to the food supply chain with Hyperledger Fabric. 2022b.
- Hyperledger .* ScanTrust Case Study – Hyperledger Foundation. 2022c.
- Hyperledger .* Benchmarking Hyperledger Fabric 2.5 Performance – Hyperledger Foundation. 2023a.
- Hyperledger .* fabric-chaincode-node | Hyperledger Fabric Node.js Smart Contracts issues in JIRA please <https://jira.hyperledger.org>. 2023b.
- Hyperledger .* fabric-gateway | Go, Node and Java client API for Hyperledger Fabric v2.4+. 2023c.
- IBM .* What are smart contracts on blockchain? | IBM. 2022.
- IBM .* IBM Supply Chain Intelligence Suite - Food Trust | IBM. 2023.
- Investopedia .* Public, Private, Permissioned Blockchains Compared. 2022.
- Jansen Maike, Gerstenberger Bastian, Bitter-Krahe Jan, Berg Holger, Sebestyén János, Schneider Jonas, Jansen M, Berg J.* Current Approaches to the Digital Product Passport for a Circular Economy An overview of projects and initiatives. 2022.
- Jqlang .* jq. 2023.
- Kathleen Wegrzyn, Eugenia Wang.* Types of Blockchain: Public, Private, or Something in Between | Blogs | Manufacturing Industry Advisor | Foley Lardner LLP. 2021.
- King Melanie R.N., Timms Paul D., Mountney Sara.* A proposed universal definition of a Digital Product Passport Ecosystem (DPPE): Worldviews, discrete capabilities, stakeholder requirements and concerns // *Journal of Cleaner Production.* 1 2023. 384. 135538.
- Lewe Elina.* What is the Digital Product Passport for textiles? – Finix. 2022.
- LogRocket .* CouchDB vs. LevelDB: Comparing state database options - LogRocket Blog. 2022.
- Lucidchart .* Introduction to the C4 Model for Visualizing Software Architecture | Lucidchart Blog. 2023.

- LyondellBasell* . Digital Product Passport Solution Prototype | LyondellBasell. 2023.
- Journal Pre-proof A blockchain architecture for industrial applications A Blockchain Architecture for Industrial Applications. // . 2022.
- McGinty David*. How to Build a Circular Economy. 2020.
- Morris Nicky*. ScanTrust's anti-counterfeit solution isn't just about blockchain - Ledger Insights - blockchain for enterprise. 2018.
- Nakamoto Satoshi*. A. The Bitcoin Whitepaper by Satoshi Nakamoto - Mastering Bitcoin, 2nd Edition [Book]. 2018.
- NodeJS* . Node.js. 2023.
- Nokelainen Marika, Tikkanen Saana, Köykkä Sami, Kieksi Lauri, Pulkkinen Anu, Roschier Solveig, Markkula Annu, Luoma Päivi, Jyrälä Minna, Bergman Leo*. Digital Product Passport SOLITA. 2023.
- NpmJS* . npm. 2023.
- Olson Kelly, Bowman Mic, Mitchell James, Amundson Shawn, Middleton Dan, Montgomery Cian*. Sawtooth: An Introduction. 2018.
- Parliament European*. The impact of textile production and waste on the environment (infographic) | News | European Parliament. 2020.
- Peppers Ken, Tuunanen Tuure, Rothenberger Marcus A., Chatterjee Samir*. A design science research methodology for information systems research // Journal of Management Information Systems. 12 2007. 24. 45–77.
- Phuwanai Thummavet*. Demystifying Hyperledger Fabric (1/3): Fabric Architecture | by Phuwanai Thummavet | Coinmonks | Medium. 2019.
- PwC* . Internet of Things | Temas Atuais | PwC Portugal. 2022.
- Queiroz Maciel M., Telles Renato, Bonilla Silvia H*. Blockchain and supply chain management integration: a systematic review of the literature // Supply Chain Management. 2 2020. 25. 241–254.
- Santander* . Linear and circular economies: What are they and what's the difference? 2021.
- Sawtooth* . Hyperledger Sawtooth. 2022.
- Shrimali Bela, Patel Hiren B*. Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities // Journal of King Saud University - Computer and Information Sciences. 10 2022. 34. 6793–6807.
- Simplilearn* . Merkle Tree in Blockchain: What is it and How does it work | Simplilearn. 2022.
- Spring Martin, Araujo Luis*. Product biographies in servitization and the circular economy // Industrial Marketing Management. 2017. 60.

- Stahel Walter R.* The circular economy // Nature 2016 531:7595. 3 2016. 531. 435–438.
- Stefan Sipka.* Digital product passports: What does the Sustainable Products Initiative. 2022.
- Sunny Justin, Undralla Naveen, Pillai V. Madhusudanan.* Supply chain transparency through blockchain-based traceability: An overview with demonstration // Computers and Industrial Engineering. 12 2020. 150.
- Formalizing and Securing Relationships on Public Networks. // . 1997.
- Trace4Value .* Trace4Value – The Trace4Value project brings together partners from several industries to tackle the complex challenge of sustainable system transformation and the shift to climate-neutral and circular production with resource-efficient and resilient value chains. 2023.
- TrueTwins .* EU to require Digital Passporting. 2022.
- VueJS .* Vue.js - The Progressive JavaScript Framework | Vue.js. 2023.
- Vuetify .* Vuetify — A Vue Component Framework. 2023.
- Wahab Abdul, Memood Waqas.* Survey of Consensus Protocols. 2022.
- Wang Xu, Zha Xuan, Ni Wei, Liu Ren Ping, Guo Y. Jay, Niu Xinxin, Zheng Kangfeng.* Survey on blockchain for Internet of Things. 2019.
- Wrisesz Jason Pimenta.* Biographies of things using blockchain. 2023.
- Ya Wang Ting.* Research Note Digital Product Passport in Fashion Industry | by Wang Ting Ya | Section 12 | Medium. 2022.
- Yang Xinting, Li Mengqi, Yu Huajing, Wang Mingting, Xu Daming, Sun Chuanheng.* A Trusted Blockchain-Based Traceability System for Fruit and Vegetable Agricultural Products // IEEE Access. 2021. 9. 36282–36293.
- Zeeve .* Guide to Hybrid Blockchain, Benefits and Use Cases. 2022.
- Zhu Hegui, Guo Yujia, Zhang Libo.* An improved convolution Merkle tree-based blockchain electronic medical record secure storage scheme // Journal of Information Security and Applications. 9 2021. 61. 102952.
- eCycle .* O que é economia linear e seus impactos? - eCycle. 2022.
- Šarac Marko, Paolović Nikola, Bacanin Nebojsa, Al-Turjman Fadi, Adamović Saša.* Increasing privacy and security by integrating a Blockchain Secure Interface into an IoT Device Security Gateway Architecture // Energy Reports. 11 2021. 7. 8075–8082.

Appendices

Appendix A

Context diagram

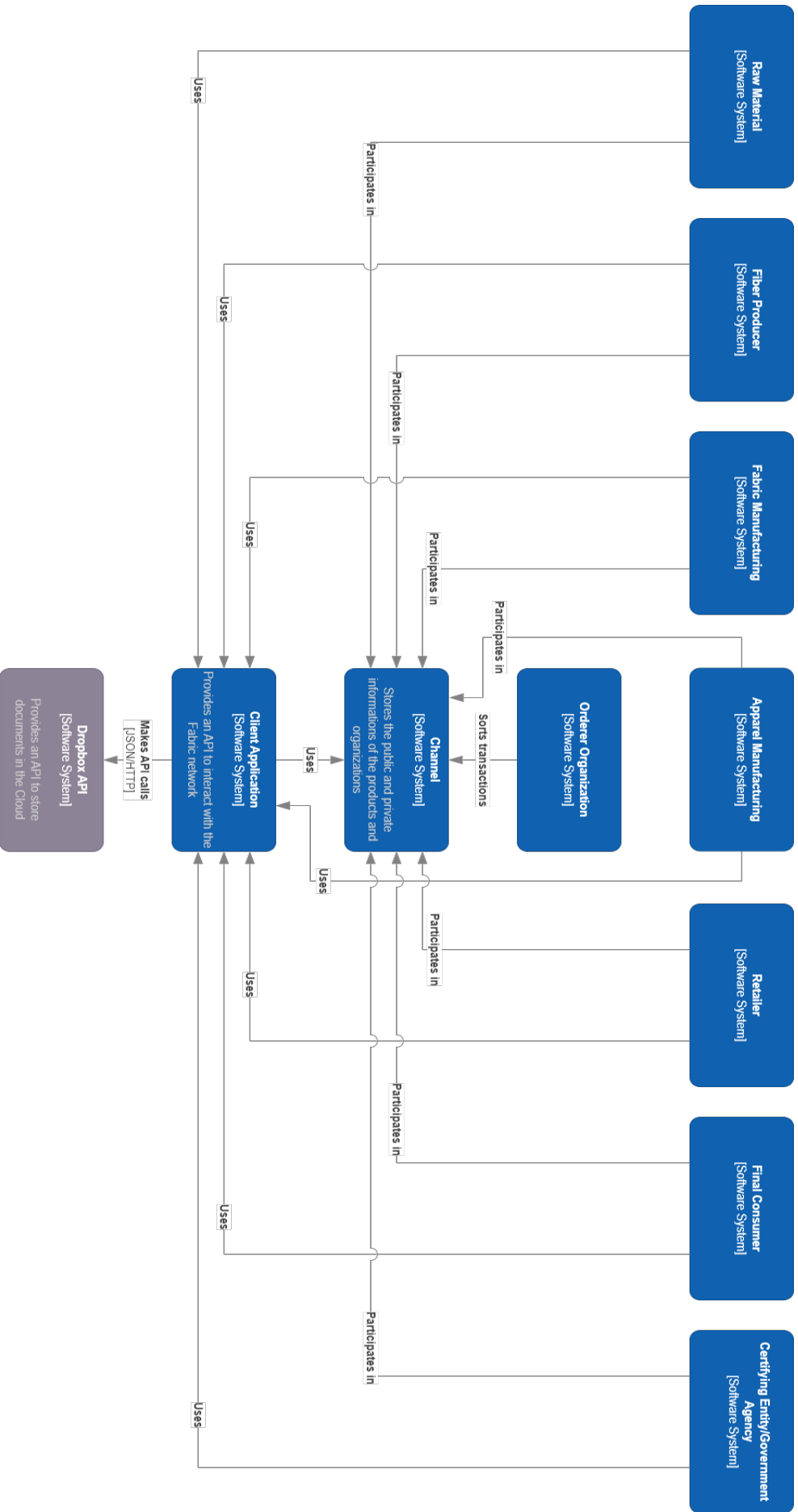


Figure A.1: Context diagram of the architecture

Appendix B

Container diagram

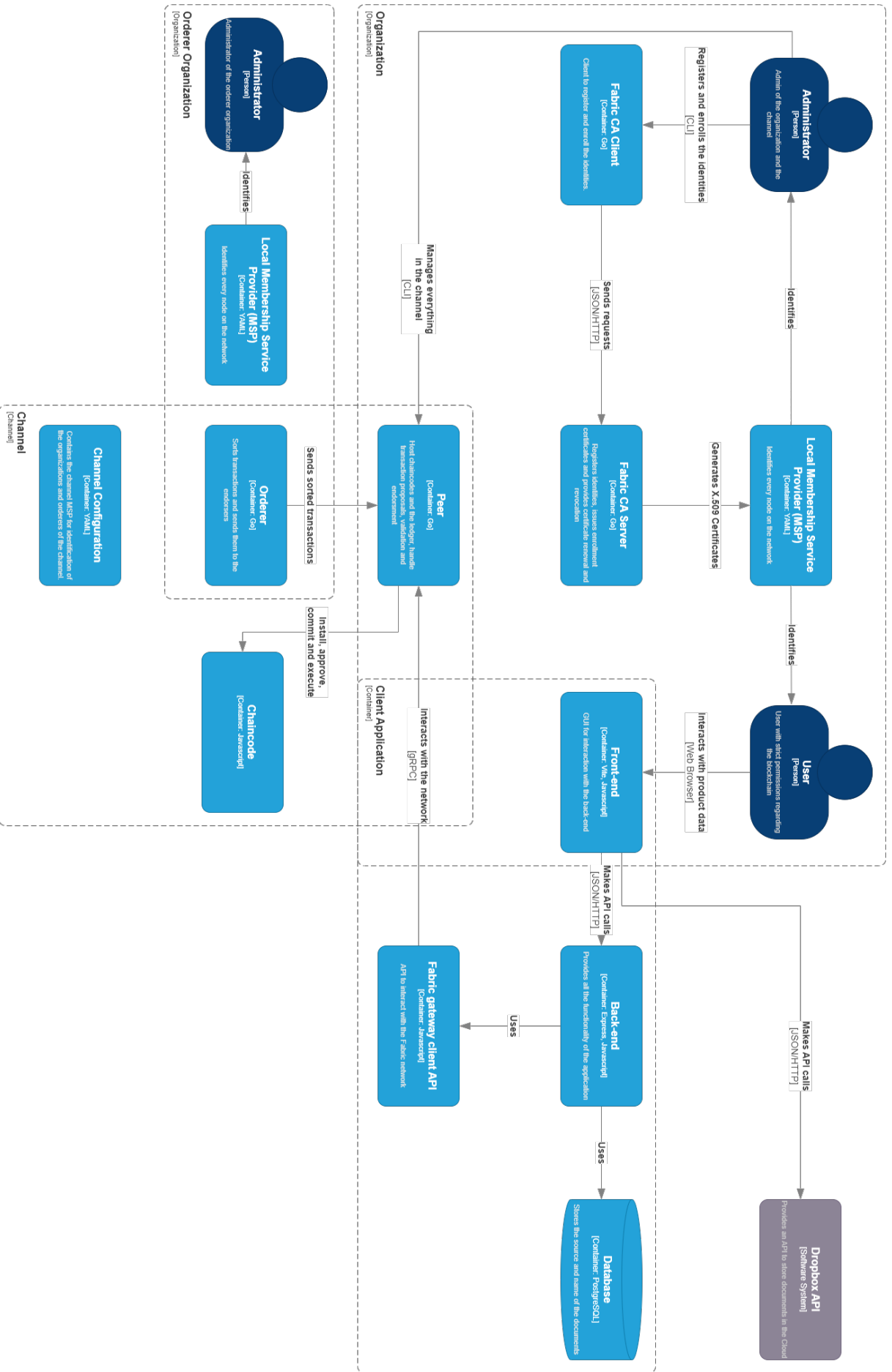


Figure B.1: Container diagram of the architecture

Appendix C

Component diagram

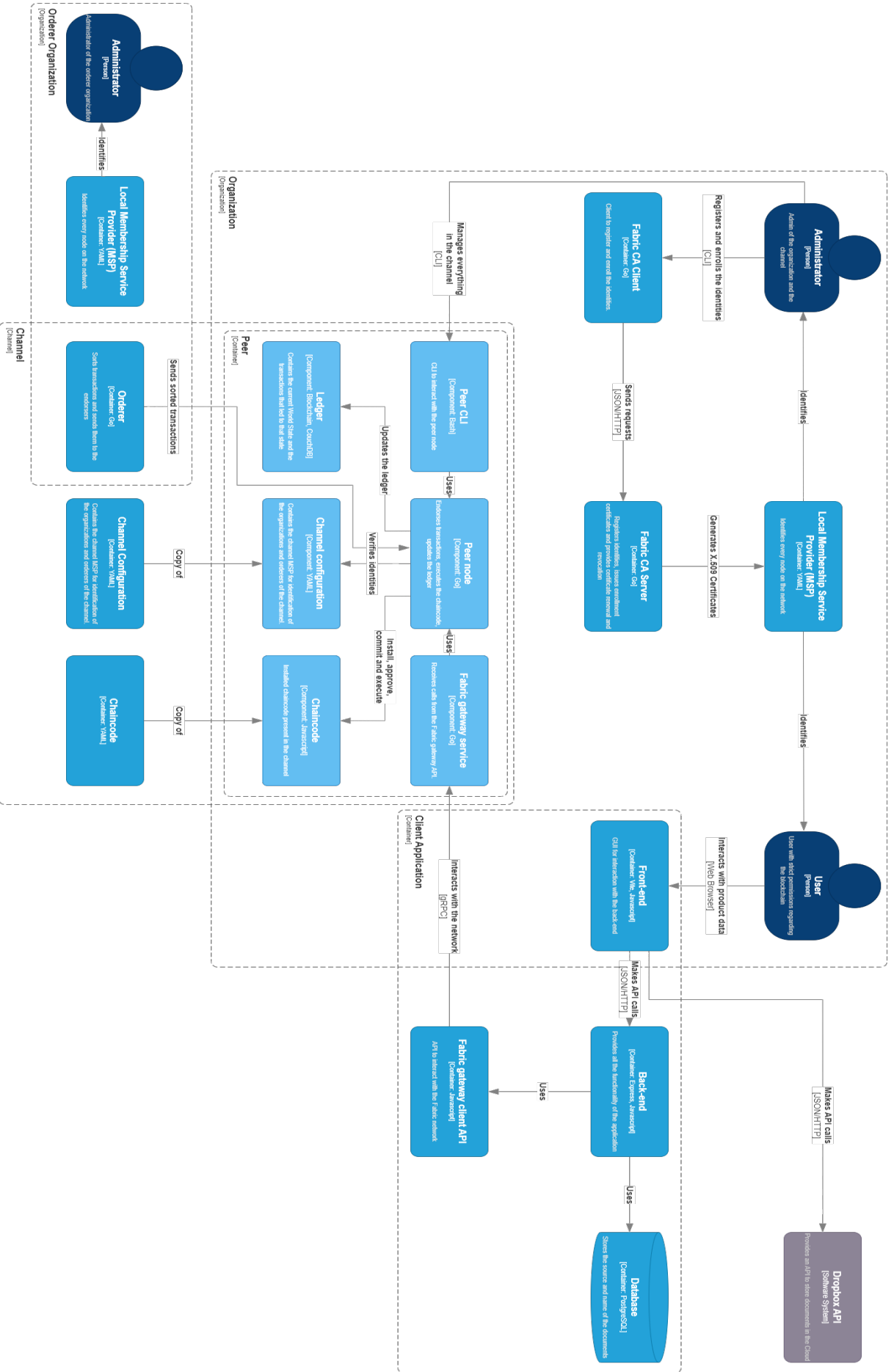


Figure C.1: Component diagram of the architecture

Appendix D

Data structure

- **Product**, represents the product, consisting of an id, name, respective events and transformations, and the id's of the products that originated it.
- **Event**, represents minor changes to products, such as transportation from one location to another, coloring of fabrics, and cutting of wood.
- **Transformation**, represents major changes associated with a product. For example, a fabric is used for the production of a sweater, or the product is in the hands of the next supply-chain phase.
- **Events and transformations id**, identify events and transformations to ease the search for files associated with each one.
- **Originated**, contains the id's of the products that originated the one in question. For example, multiple fibers can create a single fabric, and therefore, the id's of the fibers must be stored in the fabric originated array.
- **Files**, contains objects consisting of name, description, and the hash of the files associated with an event or transformation.

```
{
  "id": "1234-5678-90ab-cdef",
  "name": "Fabric1",
  "events": [
    {
      "id": 0,
      "description": "Production",
      "files": [
        {
          "name": "File0.png",
          "description": "File about Y",
          "hash": "09891b7itfg651cdec3485da435fild9"
        }
      ]
    },
    {
      "location": {
```

```
        "name": "Coimbra",
        "latitude": 40.20564,
        "longitude": -8.41955
    },
    "changes": {
        "color": "Blue",
        "quantity": 100,
        "unit": "g"
    }
}
],
"transformations": [
    {
        "id": 0,
        "description": "Cloth production",
        "files": [
            {
                "name": "File1.png",
                "description": "File about X",
                "hash": "be391b7itfg651cdec3485da435fild9"
            }
        ]
    },
    "location": {
        "name": "Porto",
        "latitude": 41.14961,
        "longitude": -8.61099
    },
    "productsId": [
        "cdef-5678-1234-90ab"
    ]
}
],
"originated": [
    "abcd-5678-1234-90ab",
    "cdef-1000-1234-90ap",
    "jff-5678-9000-977f"
]
}
```