UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

Ojiambo Ivan

A Fully Data On-chain Solution for Tracking Provenance of Handcrafted Jewellery

Master's Thesis (30 ECTS)

Supervisor(s): Luciano Garcia-Banuelos, PhD

Tartu 2019

A Fully Data On-chain Solution for Tracking Provenance of Handcrafted Jewellery

Abstract:

Blockchain and Smart Contract have been widely adopted in a number of business domains and some of the recent ones include medical records management, tracking of diamond and many more. The reason for using blockchain, is because it enhances trust through transparency and also the data stored on the blockchain is resilient to tempering. It's based on the above premise that this paper aims at building an application that uses blockchain to track the supply chain of handcraft jewellery. Tracking the supply chain of a product involves storing complex data at each and every stage of production and therefore this may require databases that can store complex data structures in order to capture all the details of the data. However, most of the blockchain platforms can only store data using key-value databases. Using key-value type of databases, data can only be saved using a data key and it's impossible to perform data operation such as data aggregation yet such data operations are of great importance when making business decisions. Hyperledger fabric is an enterprise blockchain that can be extended from using a key-value database to using couchDB, a NoSQL database with the ability to support complex queries. In this paper we investigate the capability of Hyperledger fabric's database by storing the provenance data for the handcraft jewellery onto the blockchain (on-chain). We present a case of Soko, a company that sells handcrafted products and wants to ensure transparency in the supply chain of its products. Finally, we conclude by discussing our findings and comparing our solution with the previous solution where they use both conventional databases and blockchain to store the provenance data.

Keywords: Provenance, blockchain, Hyperledger fabric, Ethereum.

CERCS: P170 - Computer science, numerical analysis, systems, control

Käsitööehete päritolu jälgimine kasutades täielikult plokiahelasse salvestatud andmete lahendust

Abstrakt:

Plokiahela (blockchain) nutilepingute (smart contracts) mehhanisme on laialdaselt kasutatud mitmetes valdkondades, sealhulgas meditsiiniliste andmete haldamises, teemantite teekonna jälgimises ja paljudes teistes kasutusalades. Plokiahela kasutamine on oma läbipaistvuse tõttu usaldusväärne ning plokiahelasse salvestatud andmed on võltsimise vastu resistentsed. Ülaltoodud eelduse põhjal on käesoleva töö eesmärgiks ehitada rakendus, mis kasutab plokiahelat käsitööehete tarneahela jälgimiseks. Toote tarneahela jälgimine hõlmab andmete salvestamist igas tootmisetapis, mistõttu see tegevus vajab andmebaasi, mis suudaks salvestada keerukaid andmestruktuure kõikide üksikasjde jäädvustamiseks. Seevastu enamik plokiahela platvormid suudavad andmeid salvestada ainult relatsioonilistes andmebaasides. Relatsioonilistes andmebaasides saab andmeid salvestada ainult võtmeväärtuste abil ning nendes ei ole võimalik teha andmete koondamise operatsioone, mis omavad suurt väärtust äriotsuste langetamisel. Hyperledger Fabric on ettevõtetetele mõeldud plokiahela raamistik, mida on võimalik laiendada nii, et relatsioonilise andmebaasi asemel on kasutusel mitterelatsiooniline (NoSQL) couchDB andmebaas, võimaldades seeläbi keerukate päringute tegemise. Käesolevas töös uurime Hyperledger Fabric'i andmebaasi võimekust salvestada käsitööehete päritolu plokiahelasse. Elulise näitena toome välja firma nimega Soko, mis müüb käsitöötooteid ning soovib tagada oma toodete tarneahela läbipaistvuse. Töö kokkuvõttes analüüsime saadud tulemusi ning võrdleme omaloodud lahendust varasemaga, kus kasutati relatsioonilist andmebaasi ja plokiahelat toodete päritolu salvestamiseks.

Märkõnad: Päritolu, Plokiahela, Hyperledger fabric, Ethereum

CERCS: P170- Arvutiteadus, arvanalüüs, süsteemid, kontroll

Table of Contents

1 Introduction

1.1 Context

Today, there are billions of products being produced every day, however most of these products the end users (consumers) have very little information availed to them about how, where and when these products were produced yet this information would be of significant importance to consumers when making final decision on what to consume [1]. The process of determining the history of a product i.e. how, where and when a product was produced is what is known as provenance [2].

Today, most of the consumers mainly rely on the manufacturer's labels such as country of origin and trade label of the manufacturer that are placed on the product's packages in order to determine the provenance of a product. Though these labels give insightful information about the products, they still do not give detailed information regarding the supply chain of the product and not to mention many of these labels have been found on fake products.

Because of the little information availed to consumers, many manufacturers have taken advantage of that to produce sub-standard products, use unethical labour, use production methods that have negative consequences on the environment and human life. For example, in 2013, BBC reported that 50,000 tonnes of meat were found to contain horse DNA [3] and in 1996 Nike was found to be using under age children in factories based in Asia [1].

The lack of provenance information about the products does not only affect the consumers but also affects manufactures, for example, manufacturers of genuine products tend to suffer from market competition from those manufacturers who sell fake products since their products are always cheaper. This brings about financial loss to such companies and consequently discouraging innovation. Additionally, keeping the supply chain of these products as a secret limits the various stakeholders such as the government and environment protectionist who can prevent Environmental, health and safety problems that could have been caused during production [3].

The handicrafts industry is no exception to the challenges listed above and sadly, the lack of clear way to identify the provenance of hand craft products affects the collectors (artisans and designers) more since most times consumers who buy handicrafts products base their consumption decisions on the popularity of the collectors (artisan or designer). [4] goes on to state that people who identify with their work set the price based on the audience that buy their products. In most cases the consumers of handicraft products are willing to pay more for handcrafts products produced by their favourite artisans or designers. Unfortunately, with the existing systems it is hard to prove the identity of the artisan or designer especially when the trade is a cross country borders.

The challenges listed above call for more transparency in the supply chain of handicraft products. [5] has proposed and developed a prototype application that uses blockchain to store the provenance data for the handcraft products. Blockchain is a special type of distributed application, which stores data permanently meaning data cannot be changed (immutable) and there is traceability of the data.

The reason for using blockchain in tracking the supply chain of handcraft is because blockchain is an append only database and therefore does not allow deleting or altering data records once stored on the blockchain. Also, every data stored on the blockchain has a digital signature of the person that submitted the transaction and all the other participants

on the network need to agree on the validity of the transaction before the transaction can be committed (saved) to the blockchain [1]. Finally, blockchain is a distributed system and therefore there is no single point of failure.

1.2 Problem statement

This paper aims at improving the provenance solution developed by [5] where he uses Ethereum blockchain as a registry for what is happening in the outside world but the core functionality of the system runs on the conventional system. The application stores the provenance data both on-chain and off-chain. One of the challenges with this type of design is that it's difficult to synchronize data between the two states. Also, the data stored off-chain is still vulnerable for tempering.

Besides the challenges that arise as a result of the application design, there are a number of challenges that arise as result of using Ethereum as the blockchain technology. Smart contract embedded in Ethereum can only use limited data structure and data can only be stored in key-value databases. The challenge is that, it is difficult to aggregate data yet data aggregation is important in any business process.

Because of the limitation in the application design and Ethereum, this thesis proposes an alternative provenance application that uses Hyperledger fabric as the blockchain technology. In our proposed application we use blockchain as the only single source of truth. Because Smart contracts in Hyperledger can be written in rich languages such as Golang, Java and JavaScript we decide to run the application business logic onto the smart contract. Running the business process on the smart contract will allow us to take advantage of all the features advertised by blockchain such as transparency and traceability.

Research Goals

This thesis aims at answering the following questions:

1. What is the impact of a NoSQL database in the architecture design of blockchain application?
2. What is the impact of running business logic of an application onto the blockchain in terms of performance, transparency and resilience?

1.3 Structure

The rest of this paper is structured as follows; the second chapter gives an overview of the state of art about provenance and blockchain technology in general. In the third chapter, we introduce a case study of Soko; a company that sell handcraft jewellery. We identify the problems with the current systems and propose possible solutions. In the fourth chapter, we describe the system design and implementation of our proposed proof-of-concept application and in the final chapter, we draw some conclusions and propose possible future works.

2 Background

In this Section, we explain the concepts and technologies that have been used in this thesis. We also overview a collection of related works that we found relevant to this thesis. By the end of this section, we should have a clear understanding of provenance, blockchain, public blockchain, permissioned blockchain and smart contracts.

2.1 Provenance

Provenance refers to the process of determining the source of origin or the history of ownership of a product, a piece of art or any entity with a value [6]. Provenance helps one to answer questions such as: where the product was manufactured, who manufactured it, which processes were used during the manufacturing, which materials were used during manufacturing, among others. Today the most common way that consumers can use to determine the provenance of a product is by looking at the trade labels placed on the product. However, these trade labels do not contain sufficient information about the entire supply chain of the product and moreover the information on these labels in most cases cannot be verified with any independent source other than the manufacturer.

Having a clear and transparent way of determining provenance of handcraft jewellery will enhance consumer's trust and confidence in the product. Blockchain technology enhances trust and transparency by storing provenance data in a permanent and verifiable way.

2.2 Blockchain

The provenance application suggested in this thesis uses blockchain technology as the underlying technology. In a nutshell, Blockchain is a distributed database that stores data in a permanent and verifiable way [7]. i.e. the data stored on the blockchain is immutable and there is traceability of all the transactions [2].

Blockchain began to attract public attention both in academia and the business world in 2008 after Satoshi Nakamoto published his paper "Bitcoin: A peer-to-peer electronic cash system" which addressed the long-time problem of double spending in electronic commerce [8]. His intention was to create a system that would eliminate unreliable third parties from electronic commerce. Blockchain become popular because of its key characteristics such as decentralization, data persistency, auditability [9] that other decentralized application did not offer.

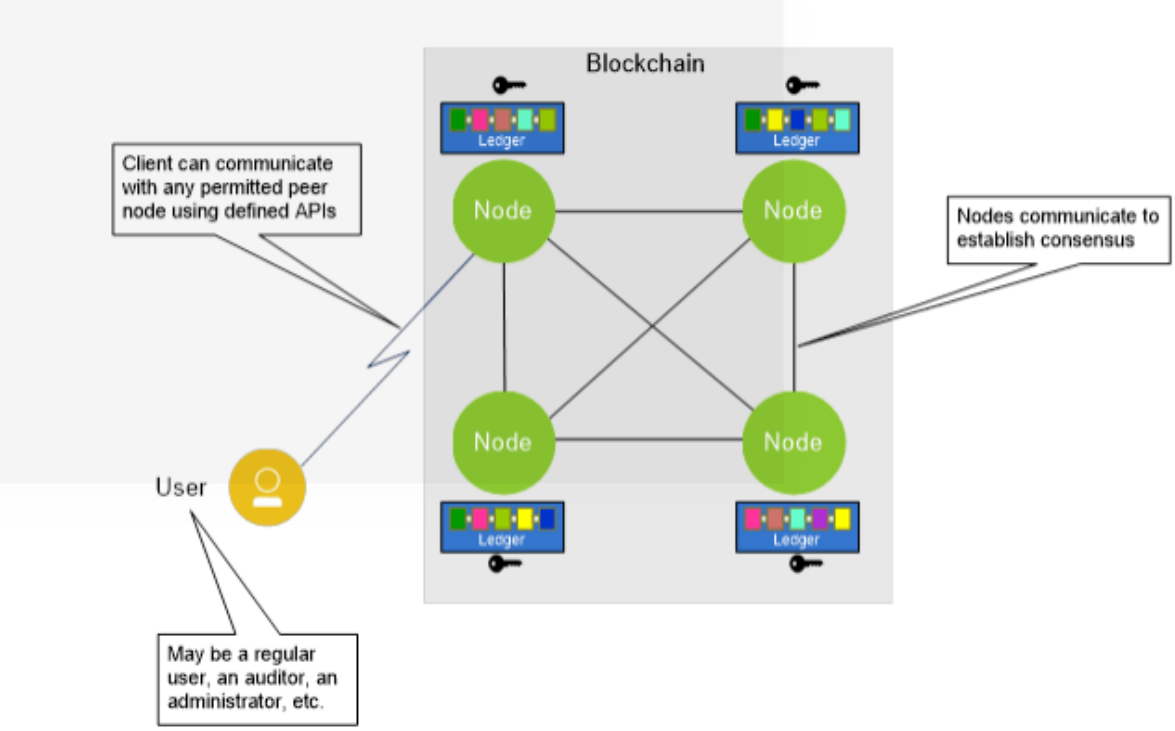


Figure 1.0. blockchain architecture – courtesy of https://www.zignuts.com

Block.

A block can be compared to a page in the ledger. A block records "n" number of approved transactions that have not been recorded in prior blocks. Each block that is created is appended to already existing blocks forming a chain of blocks thus the name blockchain. Each block consists of timestamp when the transaction was made, a cryptographic hash of the previous block and transaction data. The first block is called the genesis block and has no parent block [9] and is always created when the network is setup.

Ledger

In a blockchain application all the transactions that have been approved by the participants are stored on the ledger. In simple words, a ledger is a database residing on the blockchain that stores all committed transactions in sequenced and tamper-resistant manner. A ledger is distributed across the network and each participant has a copy of the same ledger. The transactions recorded in the ledger cannot be modified instead a new transaction (block) is created in the ledger. Each transaction invoked by the smart contract results into a set of asset key-value pair that is stored on the ledger. In Hyperledger fabric, the ledger is comprised of the state database that maintains the current state and the blockchain (" chain") that stores the immutable sequence record in a block.

Digital Signature

Each participant on the blockchain has a public key and private key which are issued by a certificate authority. The private key is kept confidential by the participant and is used to sign transactions [9]. The public key is used by the other participants (nodes) to decrypt and validate the transaction [9].

Consensus.

Before a transaction is committed to the ledger, all or the majority of the participants (peers) have to come to an agreement on whether to accept or to reject the transaction and this is

what we refer to as consensus. By having a consensus, only approved transactions are committed to the ledger and this ensures that the ledger is synchronized across all the participants. The participants approve a transaction by signing the transaction with their private key. Consensus brings about trust and prevents the content of the blockchain from tampering [10]. Blockchain uses different consensus mechanism (algorithms) to validate a transaction. For example, Bitcoin and Ethereum use proof of work algorithms as consensus strategy [9] whereas Hyperledger fabric and Hyperledger Indy use apache Kafka and byzantine fault tolerance (PBFT) respectively.

Permissionless blockchain and Permissioned blockchain.

Permissionless blockchain also known as a public blockchain refers to a blockchain platform where anyone is allowed to join the network, create transactions and participate in the validation process of transactions(mining) without the need to disclose their identity. I.e. the identity of participants on the network remain anonymous and all the participants have the same rights. An example of a permissionless blockchain include Bitcoin and Ethereum.

On the other hand, Permissioned blockchain refers to a blockchain platform where the identity of the participants on the network is known prior to joining the network. With permissioned blockchain, the participants are restricted on which resource they can access on the blockchain. Examples of permissioned blockchain include Hyperledger fabric and Corda. Permissioned blockchains provides a secure way of doing business transactions between organizations or entities that do not fully trust each other yet they have a common goal [11].

Unlike Permissionless blockchain where every participant can participate in the consensus process of the transactions, with permissioned blockchain, only pre-selected set of nodes can participate in the consensus process[1] of the transactions.

On-chain storage and Off-chain storage

On-chain storage is when data is stored on the blockchain itself [12]. If it's a public blockchain, then data stored on-chain will be available to all the participants on the network.

Off-chain storage is when data is stored outside the blockchain (say on the traditional database) and only hashed value of the data is stored on the blockchain. This is used especially when the parties involved do not want to store the entire data set on-chain but only want to use blockchain to validate the correctness of the data or in scenarios where a single transaction involves large volumes of data that may be larger than the permitted block size [12]. Most blockchain platforms have a limit on the amount of data a single block can store, for example in Ethereum, a block has a maximum capacity of 1 megabyte and because of this limitation, most blockchain platforms store the data off-chain and only store a hashed value of the data on the blockchain. The size of the hash is relatively small and therefore the corresponding cost of storage is also very low [12].

2.3 Smart contracts.

Smart contracts are computer programs that can be executed correctly by a network of mutually distrusting nodes without the need of an external authority [13]. Smart contracts contain the business rules that should be enforced before data is altered on the blockchain. The business rules have to be agreed upon by all the participants and they can't be changed by a single participant. Smart contracts deployed on blockchain platforms such as Ethereum require the participant to have a certain amount of crypto currency before they can be invoked.

[1] https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/[Accessed: 12.10.2018]

2.4 Ethereum

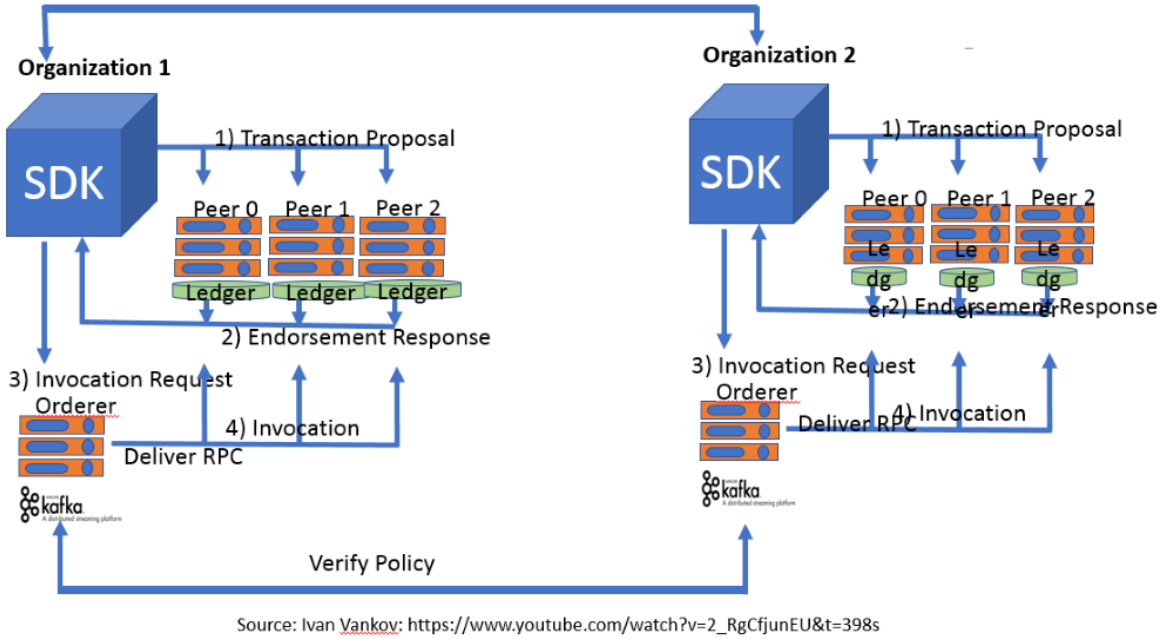
Ethereum is a decentralized computing platform that uses blockchain as the underlying technology. It's a public blockchain and therefore everyone can join the network, create transactions and validate transactions(mining). Just like bitcoin, Ethereum has its own crypto currency called ether and every transaction carried out on the network requires the participant to have certain amount of ether. Ethereum supports the implementation of smart contracts. Unlike Bitcoin's smart contract which are only limited to currency use, Ethereum uses Solidity programing language to develop smart contracts which support different use cases, including managing agreements, tracking supplier chains, and many more. Ethereum uses proof of work (POW) as the consensus protocol [14].

2.5 Hyperledger Fabric

Hyperledger fabric is an open source framework under Linux foundation used for building permissioned blockchain application. Hyperledger fabric has a modular architecture delivering high degrees of confidentiality, resiliency, flexibility, and scalability. Hyperledger fabric is a permissioned blockchain meaning the identity of each of the participants (peers) on the network is known [9] prior to joining the network and the participants enroll through a trusted member service provider (MSP).

Hyperledger fabric network can be partitioned into subnetworks called channels. Participants connected to a particular channel can only access resources that exist on that particular channel. This is important especially when an organization wants to limit access of certain resources on the blockchain to certain users.

Unlike other blockchain frameworks like Ethereum and bitcoin which require crypto currencies for every transaction that takes place on the blockchain and exchange of assets, with Hyperledger fabric, there is no need for a crypto currency and participants can agree on what to use as to exchange digital assets.



Source: Ivan Vankov: https://www.youtube.com/watch?v=2_RgCfjunEU&t=398s

Figure 2.0. Hyperledger fabric architecture design – courtesy of Ivan Vankov.

Orderer

Hyperledger fabric has a special node called Orderer (ordering service) which is responsible for accepting endorsed transactions, ordering them into a block and transfer the block to all the peers[2].

Membership service provider

Each participant or peer on the network has to have a digital identity in order to participate in the network. The membership service provider is responsible for issuing and validating certificates of users, peers and ordering service on the network.

Chain

In Hyperledger fabric, the ledger is comprised of the state database and chain. The chain is the actual "blockchain". It contains the history of all the transactions that have been committed to the blockchain[3]. The chain is structured as a hash-linked blocks with each block containing the hash of the block's transactions and the hash of the prior block. With this, all the transactions are sequenced and cryptographically linked together making it impossible to change the ledger data without breaking the hash links.

State database

State database also known as the world state stores the latest values for all keys stored in the blockchain[4]. The purpose of the state database is to make smart contract (Chain code) transaction extremely faster by looking up the current value of the key from the state database rather than search for the key current value in all the blocks. By default, Hyperledger fabric uses levelDB as the state database but also supports CouchDB. LevelDB stores data as key-value pair whereas CouchDB stores data as a JSON documents. CouchDB supports rich queries when Chain code data is modelled as JSON documents

[2] https://hyperledger-fabric.readthedocs.io/en/release-1.4/orderer/ordering_service.html
[3] https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger.html#chain
[4] https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger.html#state-database

2.6 Relational and NoSQL Databases

A relational database is a type of database which stores data in predefined tables and each record or row in the table contains a unique identifier for the categories defined by the column [15]. The data stored in relational databases can be retrieved or updated using structural query language (SQL) queries. The data stored in relational databases is subject to ACID (atomicity, consistency, isolation and durability) rules and therefore often used to store data which require high precision [15]. Relational databases are complex and for one to store data in the tables, the data must be converted into the table structure [15]. They do not work well in a distributed system because it is difficult to join tables across distributed system.

NoSQL are non-relational and therefore can store data in any structure [15]. NoSQL databases have been adopted because of their flexibility plus they allow developers to develop without having to convert in-memory structures to relational structures. Also, NoSQL databases have a high data access performance when handling large volumes of data [15]. There are 3 types of NoSQL databases; key-value pair databases, document database and column-oriented databases. Example of key-value pair database include levelDB, Redis, BerkelyDB and examples of document database include mongo DB and CouchDB.

2.7 Related works

This section gives a detailed overview of the previous research that has been conducted in the area of provenance using blockchain.

[10] has suggested a decentralized medical record management system that uses blockchain technology to share and manage medical records of patients across different health providers or medical research institutions. The proposed system makes it easy to aggregate a person's medical records in a single place where privacy and security cannot be compromised. The system allows the patients and health providers to have control on which data they would wish to share with other health providers.

The author proposes that the different stakeholders such as researchers, public health authorities will be used as miners. The author also suggests two models that can be used to incentivize the miners to participate in the network. The first one is based on Ethereum's inherent incentivizing model where the transaction fee paid in ether (a cryptocurrency for Ethereum) is paid to the miners. The second model incentivizes the miners (medical researchers and health care authorities) with aggregate anonymized data as mining rewards.

The patient's data is cryptographically hashed and stored on the blockchain. The suggested system uses a designated smart contract that aggregates reference to all patient's medical data in a single point of reference and also designs a set of rules that govern medical records. The system uses a synching algorithm to transfer data stored off-chain (medical provider's database) onto the blockchain. The smart contract defines a collection of data pointers and associated access permission that identify a patient's medical records that are held by different medical providers. Each of data points consist of a query string that when executed on the medical provider's database returns the patient's medical records that the patient has given access to the that particular medical provider.

It's important to note that even though the system in question guarantees transparency in the medical records, the system still depends on the medical provider's database for querying the medical records which results into a single point of failure.

[1] suggests a distributed system that uses blockchain to ensure transparency and traceability in the supply chain of manufactured products. The suggested system collects, manages and

stores key product information on the blockchain throughout the product life cycle. The data stored on the blockchain can then be accessed by all the stockholders' in the supply chain. For example, the consumers, the government and many others. The author also suggests that each product should have an information tag inform of RFID or QR (quick response) code that represents a unique cryptographic identifier that links a physical product to its virtual product on the network.

The author further suggests that as the product moves from the producer to the supplier and finally to the consumer, each of these actors will record the current status of the product onto the blockchain. Each of the system actors have a digital profile on the network. The profile contains key information such as their relationship with the product, location, certificates. Before any of the system actors can access the data on the blockchain, they are required to register on the network using the registrar service that assigns unique identity to each of the actors. After registration, the actor is assigned a private and public key which is used to authenticate the actor on the network and identify the actor on the network respectively. When a product is being added or modified on the blockchain, the actor signs the product using private key. In case of any malicious act by any of the actors, other actors can identify the actor behind the malicious act using his or her public key. The system has a set of rules that govern the network and they are stored on the blockchain. The rules define how the different actors will interact with the system and how data will be shared across the network. The rules cannot be changed once they have been deployed onto the blockchain. To change the rules, all the main actors have to consent and agree to them before they can be changed. Even though the system described above has the same architecture design like ours, it is still a prototype and it has not been implemented.

[16] has suggested and implemented a blockchain based application that supports data accountability and provenance tracking. The application stores the data policy agreement on the blockchain. The system allows data owners to track how companies have used their data without violating the policy agreement and in case the companies violate the policy agreement, then the data owners can always revoke the policy agreement. The data usage policies and provenance tracking information are encoded in a smart contract and stored on a block chain in a privacy-friendly way [16].

[2] has also proposed a data provenance solution that uses blockchain to audit the data operations performed on the cloud data. The proposed solution uses hooks, listeners to monitor and record all user operations performed on the cloud data. After each user operation on the files, a provenance entry is generated which is uploaded to the blockchain by cloud service and a copy of the data is stored on the normal database.

Provenance.org has developed a blockchain application that tracks the supply chain of tuna fish. The process begins by a fisherman sending a text message to register their catch which is registered as an asset on the blockchain. The text message is then accompanied by a unique Identifier that identifies that catch to the blockchain. At every stage of processing the tuna fish, the corresponding process is registered on the blockchain [3]. They use NFS (Near field communication) enabled stickers and QR codes to link the tuna fish with the digital one registered on the blockchain.

Everledger[5] and BlockVerify [17] are using blockchain to trace the origin of diamond. Their solutions are aimed at overcoming the challenges associated with the Kimberley process, a

[5] https://diamonds.everledger.io/ [Accessed: 11.11.2018]

process that ensures that global supply chain of diamond is free from conflicts diamond[6]. The proposed solution records provenance data regarding the diamond such as source of origin, how it was polished and the artisan onto the blockchain. The data is encrypted before its stored on the blockchain.

Modum.io [17] is using blockchain and IOT (internet of things) to ensure that the temperature of pharmaceutical products meets the required temperature while transporting the products. Modum.io uses sensors to record the temperature of pharmaceutical products during shipping. The sensors are placed in the same package as the containers used for transporting the products. The sensors are connected to the mobile application using Bluetooth connectivity that reads the temperature and sends the temperature readings to an HTTP server which also sends the reading to Ethereum blockchain. The Smart Contract embedded inside Ethereum is responsible for validating the temperature. The results are publicly accessible and both the distributor and receiver are notified of the results that have been stored on the blockchain.

In all of the user cases mentioned above, the provenance data is stored both on the traditional database and on the blockchain. In our proposed solution, we use blockchain as the only single source of truth.

[6] https://www.kimberleyprocess.com/en/what-kp [Accessed: 11.11.2018]

3 System overview

In this chapter, we begin by introducing our case study; Soko, an online company that sells handcraft jewellery. We describe its core business process using BPMN diagrams, identify the problems associated with the current system and propose an alternative system

3.1 Case study

The case study is about Soko, a company that sells handmade jewellery online. The company has its headquarters in San Francisco, USA and the production office in Nairobi, Kenya. Soko makes the design of the jewellery and collaborates with the local artisans centered around Nairobi for the production. The artisans use local-sourced and eco-friendly materials such as reclaimed cow horns and bone as the raw material in the production of the jewellery. Soko has web-based application (https://shopsoko.com/) where clients can go to make their sales order. The web application contains the design specification of their products. Once an order has been placed, the application send a purchase order to the artisan who is responsible for the production. The artisans receive the orders through a mobile application. The artisan creates the handcraft based on the design specification in the order. After production, the artisan, delivers the handcraft pieces to Soko offices where they have to perform a quality test before they are accepted. Soko also employs artisans who improve the handcraft by polishing, electroplating and painting. The production process takes 7-10 days before the product is delivered to the customer. Figure 3.0 illustrates the supply chain of the handcraft jewellery.

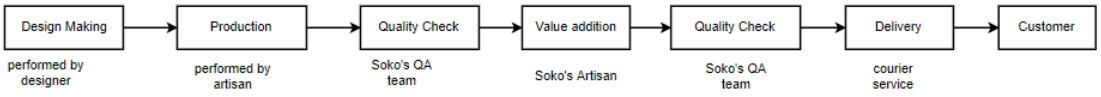


Figure 3.0. supply chain.

As illustrated in figure 3.0, the handcraft jewellery under goes different stages before reaching the final consumer. Soko's business model is centred around providing ethically handmade jewellery to its customers and therefore it's important that they keep the supply chain transparent such that their customers can be confident that they are supporting a genuine course. However, with the current system there is no verifiable way that the consumers can use to prove the authenticity of the jewellery that they buy. The consumers can only rely on the labels placed on the jewellery which do not provide sufficient information about how, where the jewellery was produced.

Because of the challenge mentioned above, [5] has proposed and developed a system that uses blockchain to ensure transparency and traceability in the supply chain of handcraft jewellery. The system records all the processes involved in the supply chain on both a relational database and the blockchain. Figure 4.0 illustrates As-is business process of the system developed by [5].

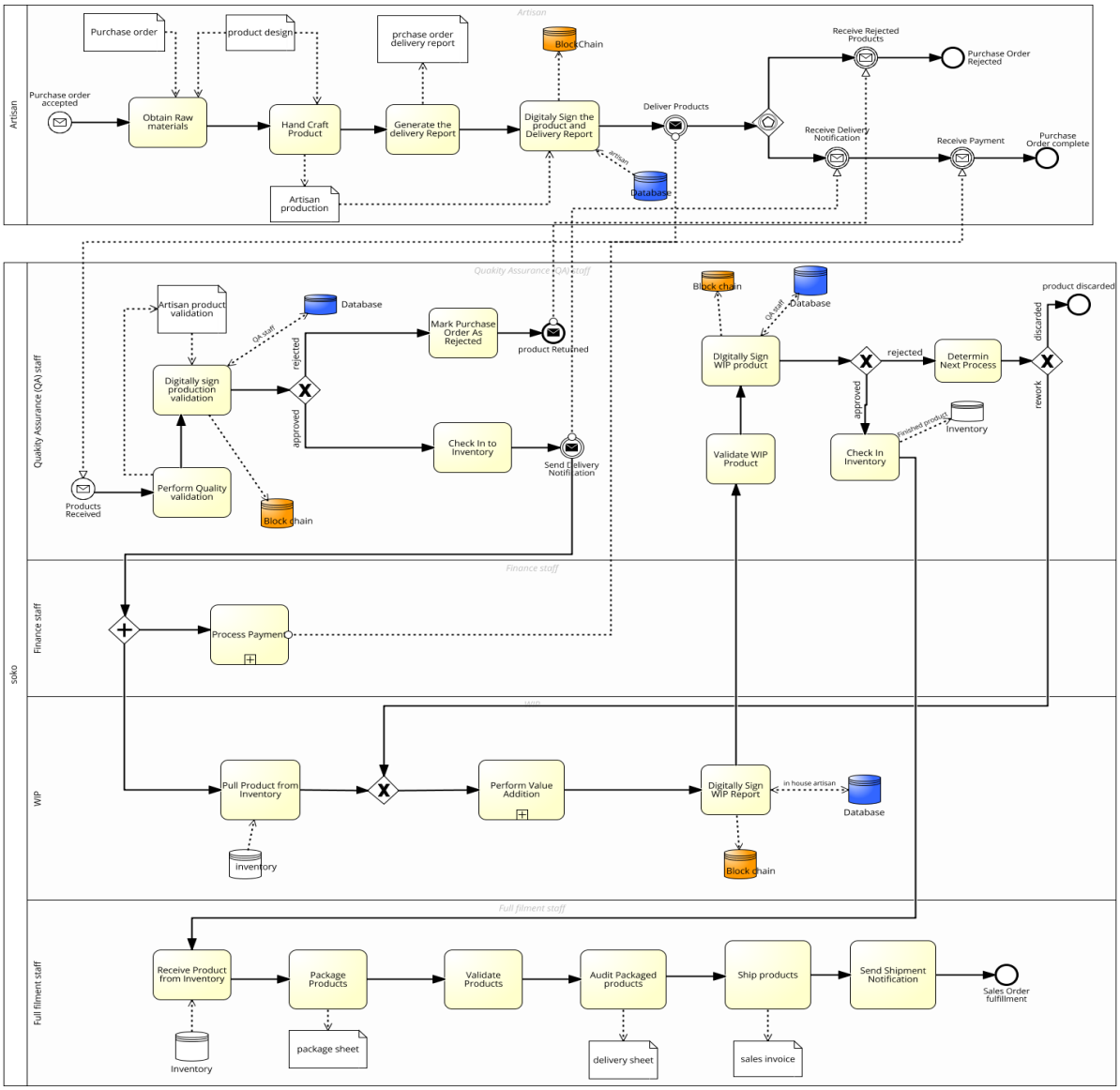


Figure 4.0. As-Is business process.

As shown on figure 4.0, the data stored on the blockchain is cryptographically signed by the responsible actor. This ensures there is data integrity of the data stored on the blockchain and therefore it can be trusted by other parties involved.

3.2 Challenges with existing solution

Even though the system developed by [5] may guarantee transparency in the supply chain of handcraft Jewellery, it makes minimal use of blockchain yet there are issues within the business process that could be solved by blockchain. For example, the artisan exchanges the handcraft to Soko without guarantee that they will be paid back yet blockchain can be used to foster smooth transactions that guarantee trust and transparency. The rest of the issues are as a result in the limitation of Ethereum, the blockchain technology used in the prototype.

1. **Limited querying capability**.
   Smart contract in Ethereum are written in solidity, a programming language which does not have an interface for specifying queries when retrieving data from the ledger. Solidity uses mapping to store data, this means that, data can only be retrieved by scanning through a map using a map key. This makes it hard to retrieve specific information that could be of interest to the final user of the system. For example, if an artisan wanted to know how many pieces of handcrafts jewelry did not pass the quality control process within a given period of time, then retrieving this kind of information may not be possible by merely using a map key. In summary, with the existing system it's difficult to aggregate data yet data aggregation is very critical part of any business when making business decisions.

2. **Risk of data manipulation**.
   Ethereum being a public blockchain any participant can join the network and can participate in the validation process of the transactions (mining process). Since Ethereum uses proof of work algorithm where a consensus is reached by majority vote, then it's possible for malicious users to join the network and obtain more than 51% of the network and therefore being in a position to manipulate the provenance data stored on the blockchain.

3. **High maintaining costs**
   Every transaction on Ethereum that invokes the smart contract requires the participant to pay ether as a transaction fee [16]. The transaction fee is meant for maintaining the network and motivating the miners to participate in the mining process. This may not be feasible since tracking the supply chain of handicraft product may involve invoking the smart contract multiple times and consequently making the solution expensive. Additionally, Ethereum uses the proof of work algorithm as consensus mechanism which requires a lot of computation power and therefore consuming a lot of energy

4. **Low transaction throughput**
   Because Ethereum is a public blockchain and in a public blockchain every nod is required to participate in the mining process. This results into low transaction throughput as the size of the nodes increases.

It's based on the above challenges that we propose an alternative provenance application that uses a different blockchain technology.

3.3 Proposed provenance application

We propose an alternative provenance application that uses Hyperledger fabric, a permissioned blockchain application as the underlying blockchain technology. Since block chain is considered as a database [18], we use it as the application database to store both provenance data and other application data such as purchase order. We also embed the application business logic into the smart contract. The goal is to maximize the benefits of blockchain such as security, temper resistance and transparency.

Why Hyperledger fabric

There are a number of blockchain frameworks and the choice of which framework to use depends on the business requirements. In our proof-of-concept application we chose Hyperledger fabric as the underlying blockchain technology because of the following reasons

1. **It's a Permissioned blockchain**.
   Hyperledger fabric is a permissioned blockchain meaning all participants on the network have a known identity prior to joining the network. With this, we are sure that malicious users can not join the network to influence the validation process of the transaction. Also, participants cannot misbehave because everyone on the network will get to know and the necessary action can be taken.

2. **Highly scalable**
   Blockchain frameworks such as Ethereum uses proof of work algorithm as the consensus mechanism which requires every node connected on the network to participate in the mining process. This leads to low transaction throughput as the as the number of miners increases. However, with Hyperledger fabric, a few trusted nodes can be selected to participate in the consensus process and consequently leading to a high transaction throughput.

3. **Has a modular design.**
   Hyperledger fabric has a modular design which allows extensibility and flexibility [19]. Developers can easily plugin-components that suit that their demands for example developers can include custom identity management into already existing identity management.

4. **Support for complex queries.**
   Hyperledger fabric state database can be extended to use CouchDB, a NOSQL database which allows us to perform complex queries when aggregating data.

To-Be Business process

The following assumptions are taken into account for our to-be business process:

1. Because the trade between the artisan and the Soko is of international nature we assume that payments are done using banks.
2. We assume that letter of credit is accepted as form financial security.

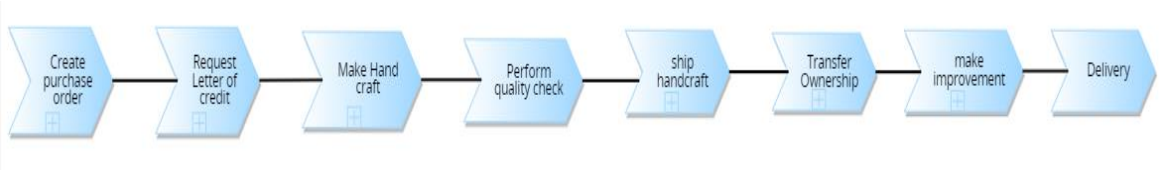


Figure 5.0: provenance application value chain

Our application aims at capturing all the process involved from the time the client makes an order to when an order is delivered as shown on figure 5.0. As soon as a user makes a sales order, Soko sends a purchase order to the responsible artisan for production.

Ideally, the initial production stages such as collecting raw materials may have started way before the order was made but for the purpose of ideation, we shall capture the production after an order has been made.

Purchase order creation

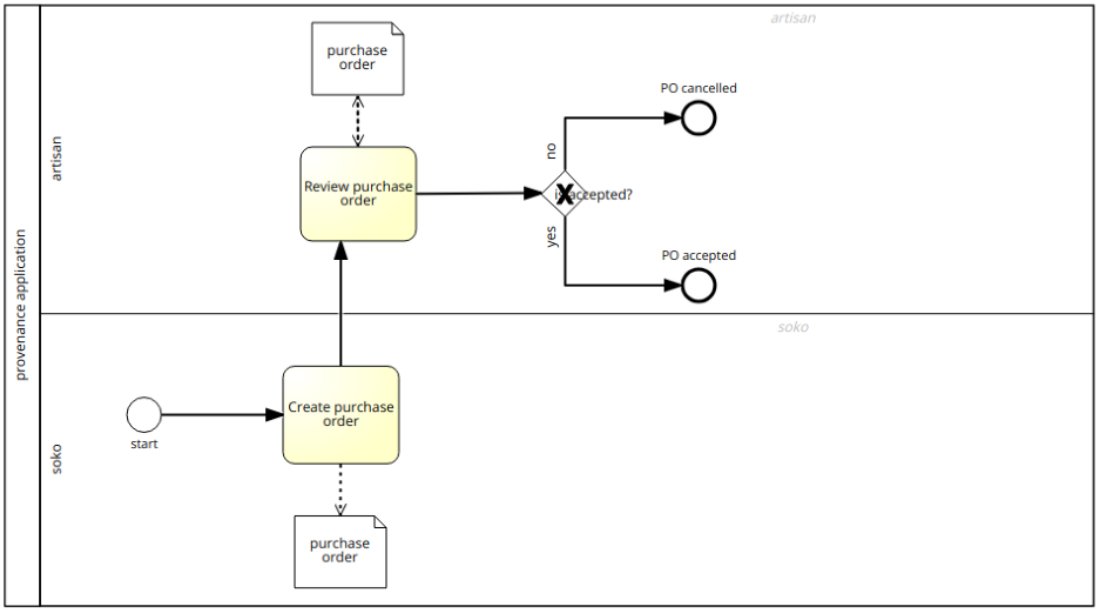


Figure 6.0: purchase order creation.

As shown in figure 6.0, when a purchase order is created, a corresponding purchase order object will be created and stored on the blockchain.

Request Letter of credit process.

Given the international nature of the business, we use letter of credit as means of payment settlement. Instead of using crypto currency as way of settlement for a transaction, we decide to use the letter of credit which is more internationally accepted form of payment for international transactions.

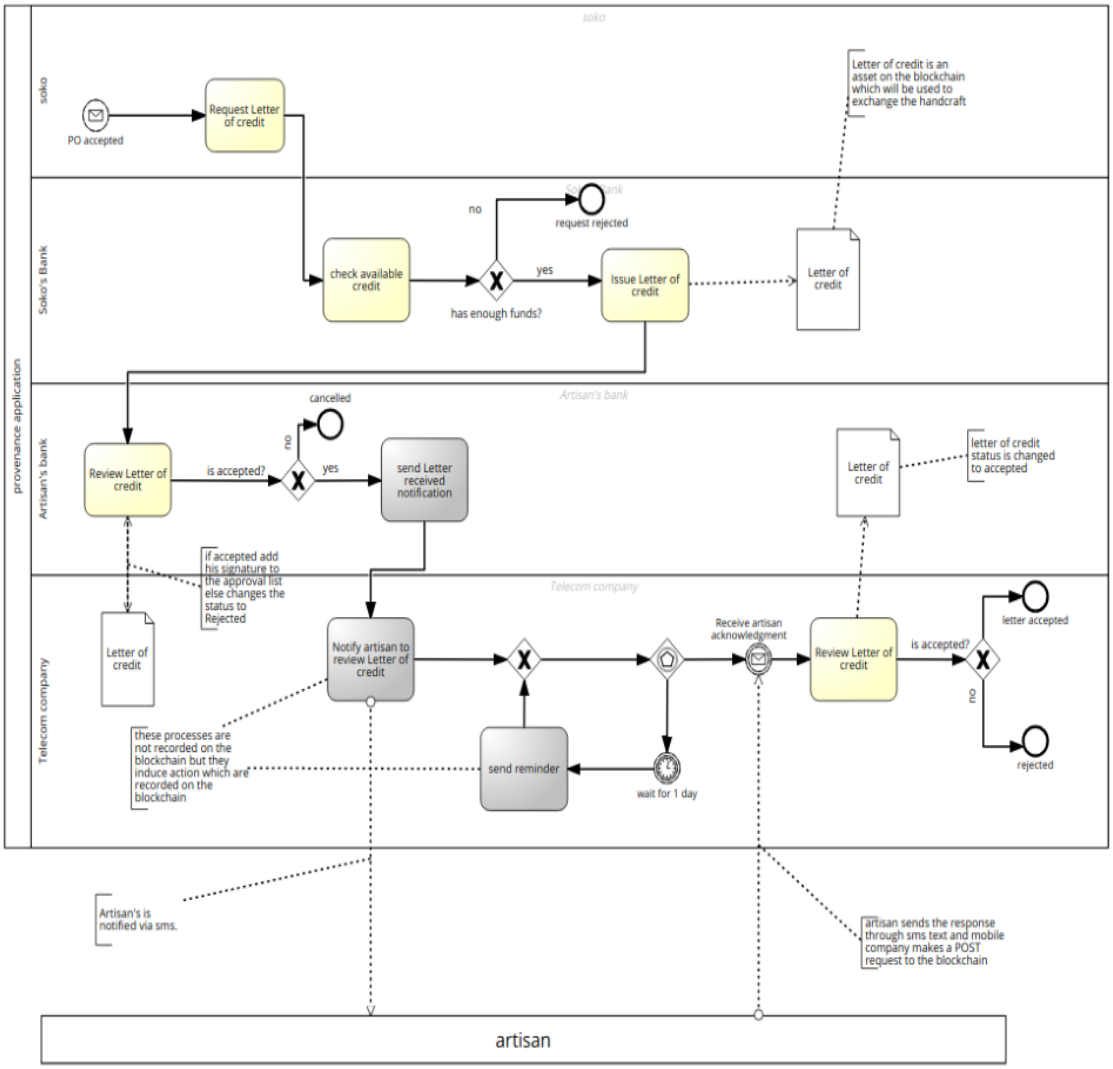


Figure 7.0. Letter of credit process

As shown in figure 7.0, the letter of credit process involves 5 actors (the artisan, the artisan's bank, Soko, Soko's bank and the telecom company). All the actors have to sign the letter using their digital certificate before it can be accepted. Since most people in developing countries like Kenya are not connected to the internet, we shall use the mobile companies as the intermediary between the artisans and their corresponding banks to sign the letter.

Production process

The production process captures both the design process and actual production. Soko employs a designer who is responsible for making the designs. The design is then sent to the artisan for handcrafting as shown in figure 8.0.

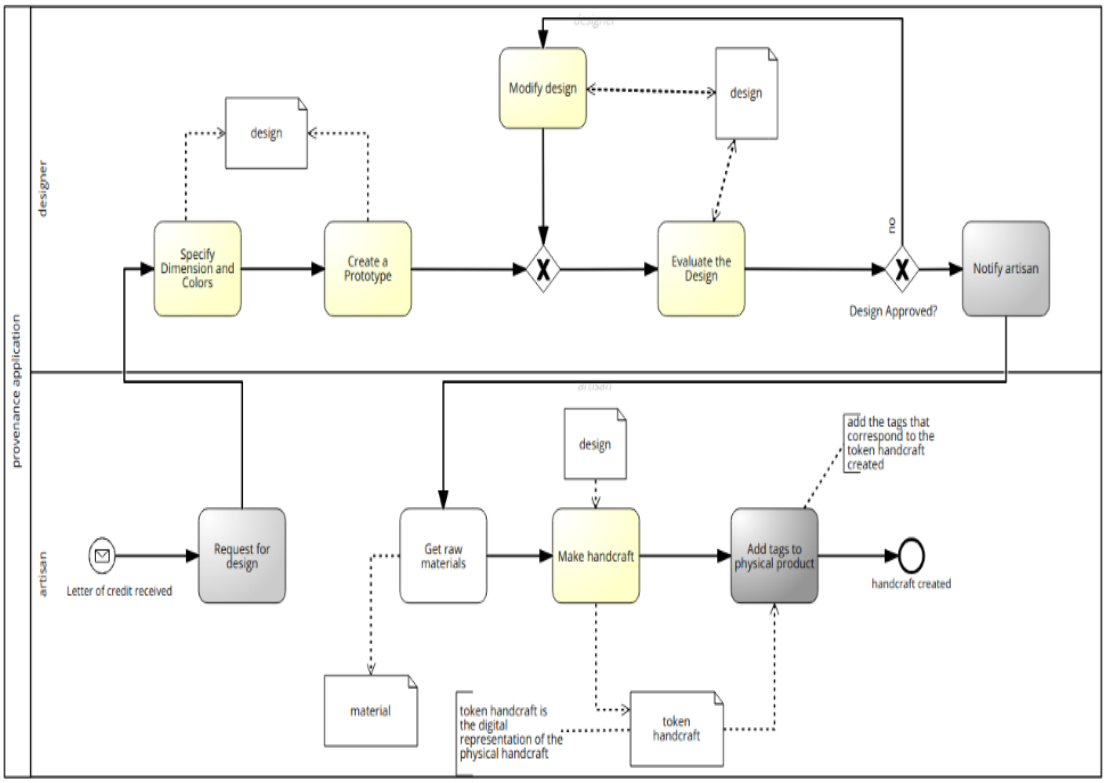


Figure 8.0: production process.

When the design is complete, a design token is created on the blockchain which can be used to trace its provenance. The design token will contain the dimensions, colors to be used and a digital identity of the designer. Similarly, when the production is finished a handcraft token is created on the blockchain which represents the physical handcraft jewellery. The handcraft token will contain the raw material used, the design used and the digital identity of the artisan.

Shipping process

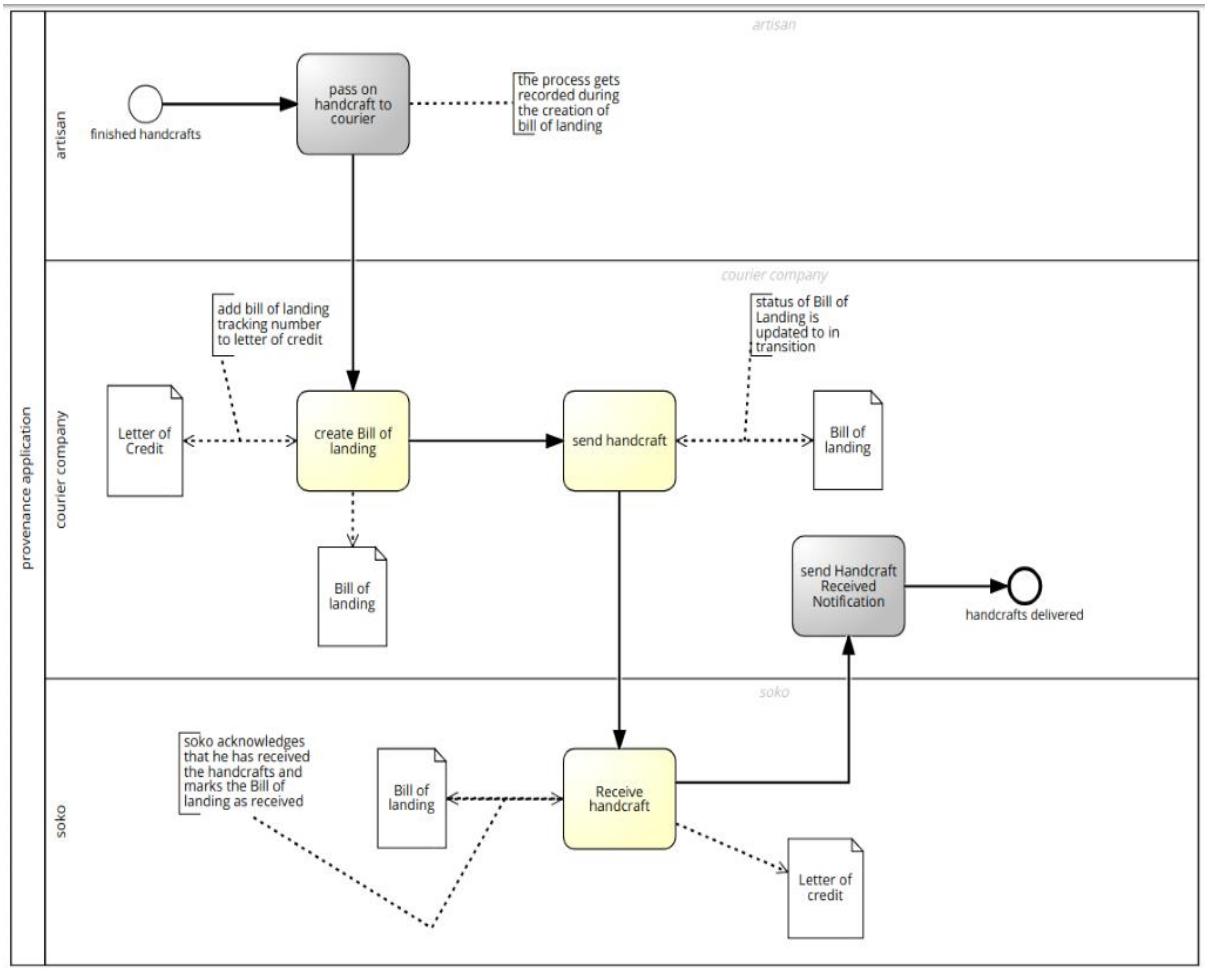


Figure 9.0: shipping process

When the production process is finished, the handcraft is shipped by a shipping company to Soko offices. The shipping company creates a bill of lading which contains a tracking number for the cargo. The bill of landing will be recorded onto the blockchain platform and linked with the letter of credit as shown in figure 9.0. The bill of lading will act as proof of shipment to all the parties involved like the two banks and Soko.

Payment and Transfer of Ownership process

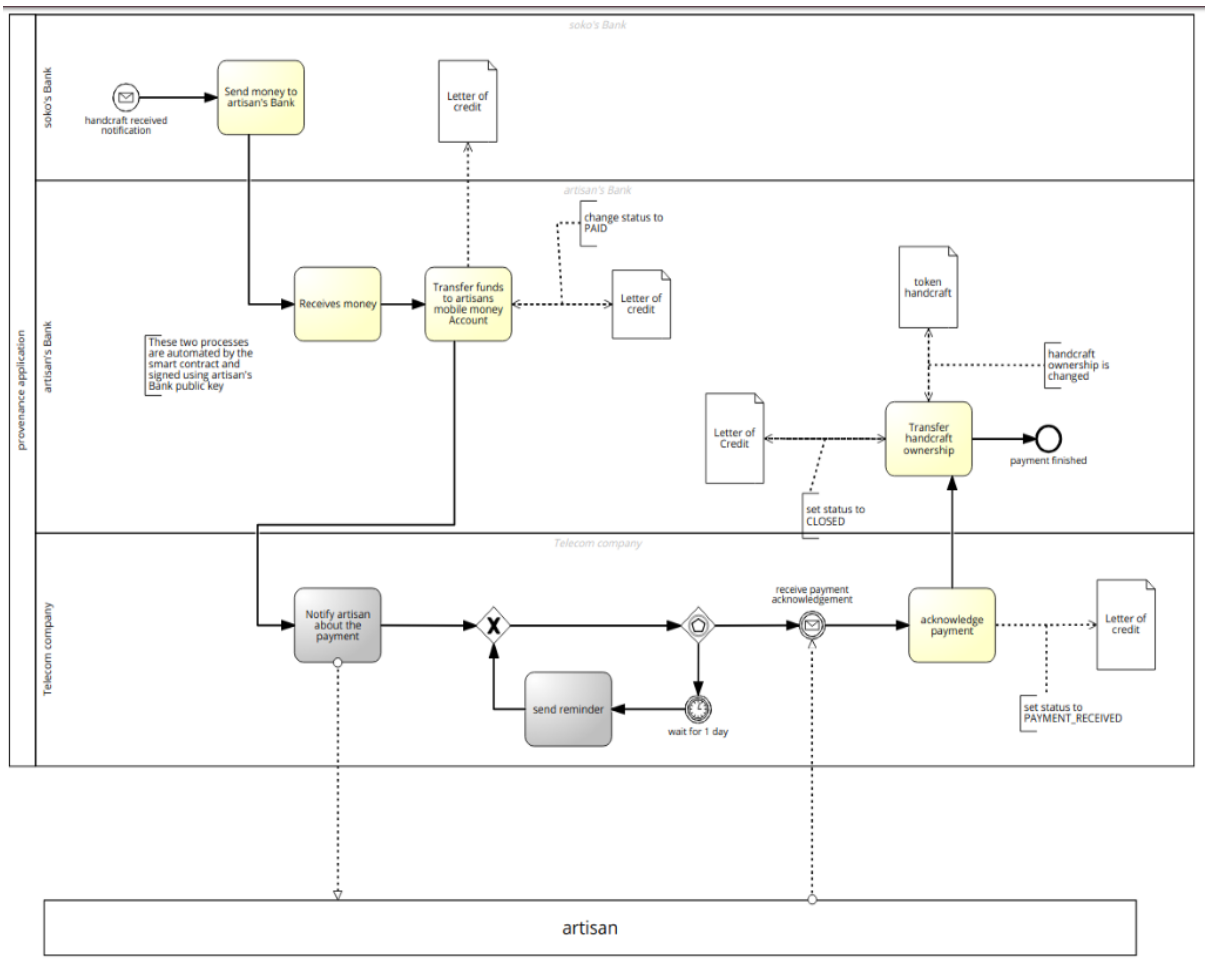


Figure 10.0. payment process.

The payment process is initiated by Soko's bank upon confirming that the handcraft has been received by Soko (figure 10.0). After the payment has been made, the letter of credit is updated and later after the artisan has confirmed the payment the handcraft ownership can be transferred.

Value addition

During value addition, the handcraft is fine tuned to look more elegant. Fine tuning may involve more than one process and each of these processes are recorded on the blockchain as shown in figure 11.0.

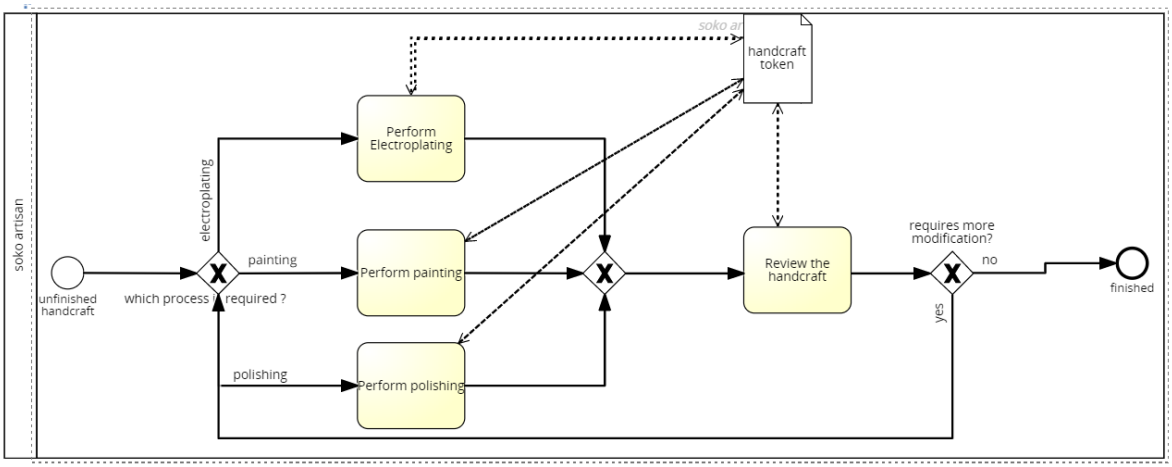


Figure 11.0. Value addition

Domain Model

In the previous section we have been describing Soko's handcraft business from the business process perspective, in this section, we break down the data involved in the application using conceptual class model diagrams as shown in figure 12.0 and figure 13.0.

Purchase order and Letter of credit creation

When a client makes an order from Soko, Soko sends a purchase order to the artisan. The purchase order may contain one or more orders with each order (*PurchaseOrderLineItem*) containing the name of the design, description, price and amount. When the purchase order is accepted by the artisan, Soko sends a letter of credit to the artisan. The Letter of credit will contain the terms and conditions that are to be met before payment can be made. Below (figure 12.0) is a class modal diagram that illustrates the creation of purchase order and Letter of credit.

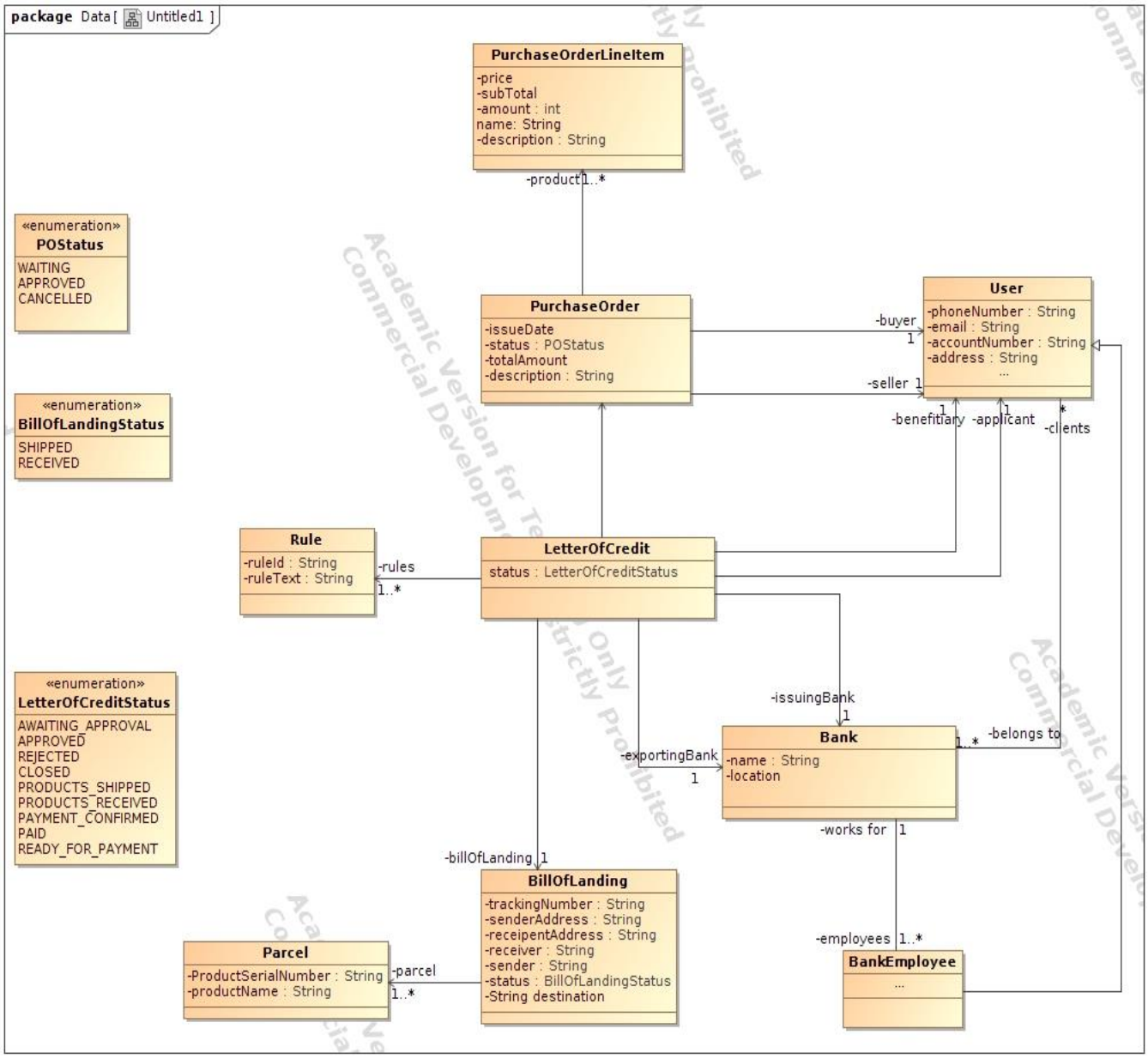


Figure 12.0. purchase order & Letter of credit creation domain model.

Production and value addition

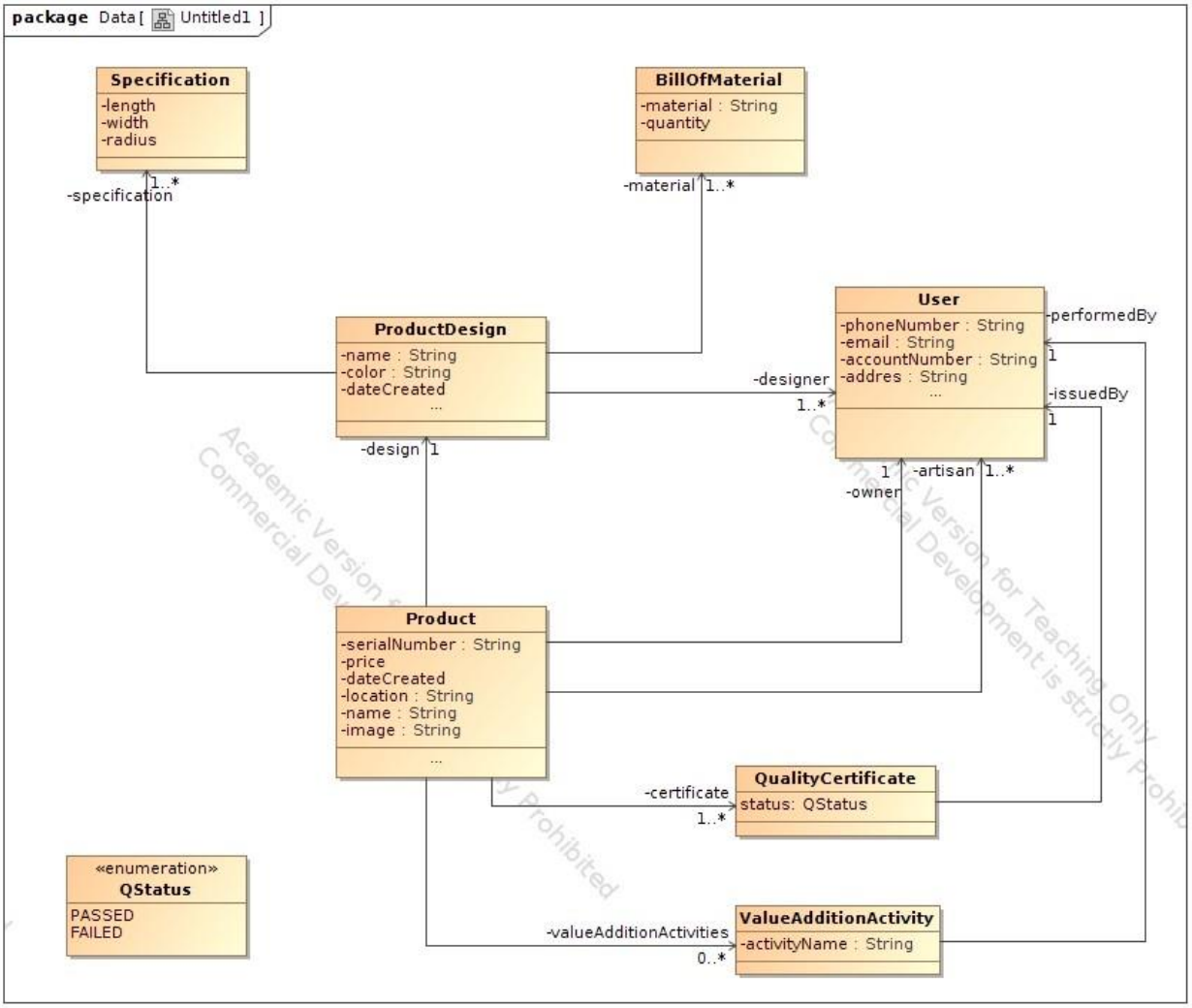


Figure 13.0. domain model

There are 3 primary entities during production and value addition process: *ProductDesign*, *Product and the User* entity (figure 13.0). The *ProductDesign* entity represents the design of the handcraft, the *Product* entity represents the actual physical handcraft jewellery whereas the *User* entity represents the different actors (artisan, designer, quality analyst) that have made changes either to the design or the handcraft.

3.4 Summary

In this chapter we described Soko's handcraft business, identified the problem with their current business model. We went on to describe the proof-of-concept application developed by [5] and its limitation. We introduced our proposed proof-of-concept application that uses Hyperledger fabric as the underlying technology. We also modelled our proposed solution using BPMN diagrams and finally, we modelled our solution using domain modal class diagrams.

4 Implementation

In this chapter, we describe the design and implementation for our proof-of-concept application and we finally conclude by discussing our solution.

4.1 System design

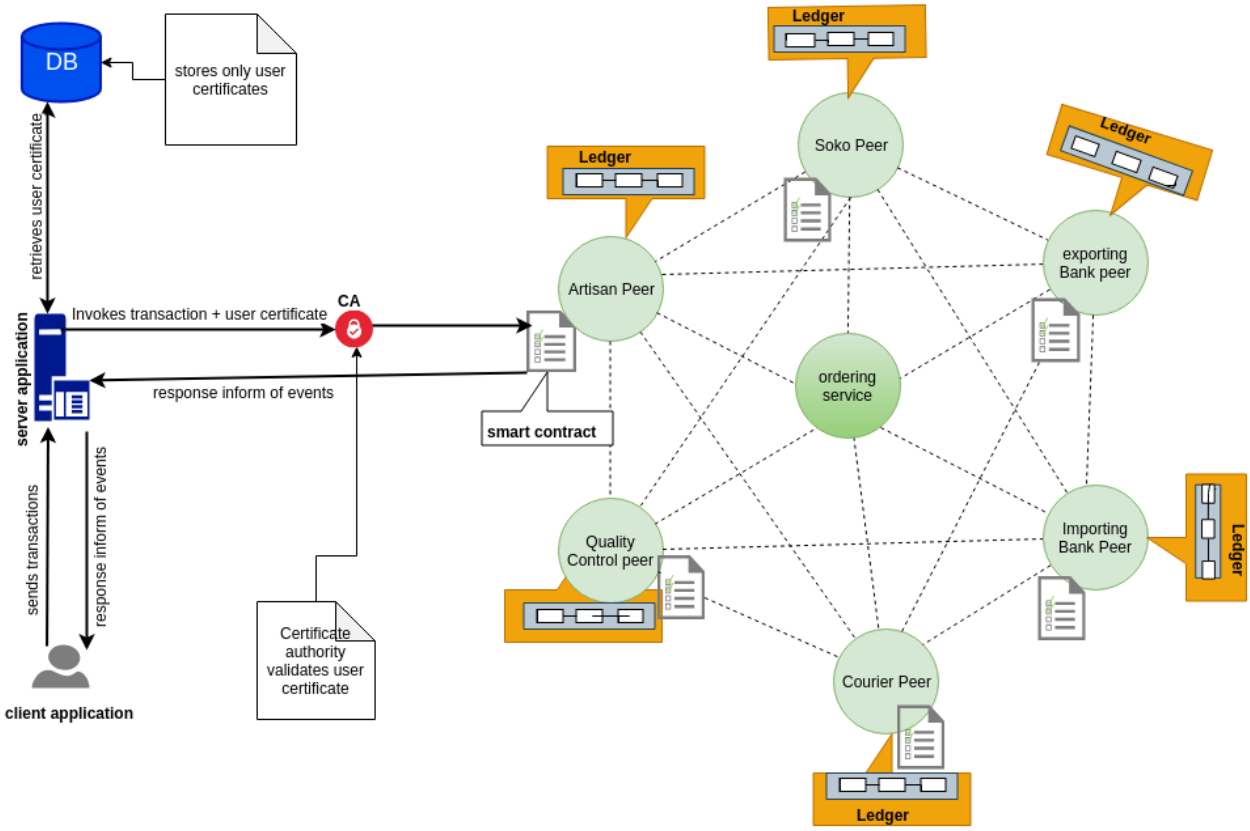


Figure 14.0: proof-of-concept system design

The system is composed of client application, back-end server, a blockchain network and a certificate authority (CA).

Since we are building a permissioned application, each user that login into the system should have their own identity that is known on the network, we therefore store the user certificates on database where the server can retrieve the certificate when signing transactions. With this, a single node can be accessed by different participants of the same organization with different identities. For example, the designer and the artisan may use a single node but their transaction will be signed by their respective user certificates.

4.2 Network setup

As shown in figure 14.0, the network is made up of six independent organization (nodes) i.e. the artisan, trader, courier, quality analyst, exporting bank and importing bank. These six organization reflect the minimum number of participants in an export trade entity of this nature but the number of participants can be more. Each node is made up of a peer, a certificate authority and a state database (CouchDB).

Peer

As mentioned in chapter 2, that there are two types of peers: endorsing and committing peers. In our system, each organization has a single a peer and each of them is an endorsing peer. We use docker container to setup the peers and each peer runs on a separate docker container. More information on how to setup the peers can be found on the GitHub repository[7]

Certificate authority (CA)

The membership provider (certificate authority) is responsible for issuing and validating digital identities of participants and peers. The digital identity is in form of cryptographic certificate. In our system, we use the default certificate authority that comes with Hyperledger fabric to generate certificates. Each organization has a certificate authority that runs on a separate docker container. In production, a certificate authority can be shared by more than one organization. More information on how to setup the CA can be found the Github repository[8].

State database

As shown in figure 14.0, each node is connected to a state database. We use couchDB as state database instead of the default levelDB because it supports more complex queries compared to levelDB. Each instance of the couchDB runs on a separate docker container. For more information on how to setup the state database can be found on the repository[9]

4.3 Endorsement policy

As mentioned earlier, a transaction can only be committed to the ledger when it has been endorsed by appropriate number of endorsement peers. The endorsement policy contains the public certificate id for the endorsing peer and the number of endorsements that a transaction should have before it can be considered to be valid. Below is a code snippet for our endorsement policy

[7] https://github.com/ivangsp/provenance-Jewellary/blob/networkSetup/network/base/docker-compose-base.yaml

[8] https://github.com/ivangsp/provenance-Jewellary/blob/master/network/docker-compose-cas-template.yaml

[9] https://github.com/ivangsp/provenance-Jewellary/blob/networkSetup/network/docker-compose-couch.yaml

```
{
    "identities": [
        {
            "role": {
                "name": "member",
                "mspId": "ArtisanOrgMSP"
            }
        },
        {
            "role": {
                "name": "member",
                "mspId": "SokoOrgMSP"
            }
        },
        {
            "role": {
                "name": "member",
                "mspId": "BankOrgMSP"
            }
        },
        {
            "role": {
                "name": "member",
                "mspId": "RegulatorOrgMSP"
            }
        },
        {
            "role": {
                "name": "member",
                "mspId": "CarrierOrgMSP"
            }
        }
    ],
    "policy": {
        "2-of": [
            {
                "signed-by": 0
            },
            {
                "signed-by": 1
            },
            {
                "signed-by": 2
            },
            {
```

From our endorsement policy, all the peers can participate in the endorsement process and a transaction will be considered valid if it has been endorsed by at least two peers (2-of). With this endorsement policy we are sure that at least two parties are aware of the transaction and have agreed to the transaction.

4.4 Smart Contract

The smart contract contains the business logic of the application. We decided to embed the application business logic in the smart contract to avoid tempering that could have occurred when hosted on a centralized application. We implement the smart contract using Hyperledger composer, an open source development tool for creating blockchain application. Hyperledger composer is built on top of Hyperledger fabric and makes it easy to develop and test blockchain applications. We chose to use composer because of its simplicity plus it comes with a modelling language.

The Smart contract defines the database schema for each of the items stored on the blockchain. Items that can be exchanged or have monetary value are defined using the keyword “asset”. For example, in our smart contract the handcraft token, the design token, purchase

order are defined as assets. The code snippet below shows the database schema for the handcraft token.

```
asset Product identified by serial_number {
  o String serial_number
  o String image
  o String name
  o Double price
  o String location
  --> Artisan [] artisans
  --> ProductDesign design
  --> Person owner
  o QualityCertificate qualityCertificate optional
  o ValueAdditionActivity [] valueAdditionActivities optional
}
```

As it can be seen from the code snippet, each handcraft product has a variable "serial_number" which will be used to uniquely identify the product.

The smart contract also defines the schema for the different actors (artisan, designers, …) participating on the network using key word "participant". Just like "asset", a "participant" is simply an object but with unique identifier. Below is a code snippet how the artisan is defined, more information on how other actors were defined can be found on the repository[10]

```
// PARTICIPANTS
abstract participant Person identified by id {
  o String id
 o String userName
 o String lastName optional
  --> Bank bank
  o String email
  o String accountNumber
  o String address
}

participant Artisan extends Person {
  o Company company optional
}
```

The smart contract does not define the schema for the admin of the business network because the admin is used to create other participants on the network.

The actual business logic of the application is defined as transactions. Transactions are responsible for creating and updating the data on the blockchain. It's important to note that the data on the blockchain is not updated instead a new copy of the data is created in a different block. It's only the state database that is updated. Below is a code snippet on how the handcraft token is created, more information can be found on the repository[11]

[10] https://github.com/ivangsp/provenance-Jewellary/blob/composer/chaincode/trade-network/models/org.trade.com.cto

[11] https://github.com/ivangsp/provenance-Jewellary/blob/composer/chaincode/trade-network/lib/product.js

```
// TRANSACTIONS
transaction CreateProduct {
  o String serial_number
  o String image
  o String name
  o String location
  o Double price
  o String [] artisanIds
  o String designId
  o String ownerId
}
```

```
/**
 *
 * @param {org.trade.com.CreateProduct} transaction - the createProduct transaction
 * @transaction
 */
async function createProduct(transaction) {
    const namespace = 'org.trade.com';
    const factory = getFactory();

    const prodRegistry = await getAssetRegistry('org.trade.com.Product');
    const product = factory.newResource(namespace, 'Product',  transaction.serial_number);

    product.name = transaction.name;
    product.image = transaction.image;
    product.price = transaction.price;
    product.location = transaction.location;

        const artisans = [];
        transaction.artisanIds.forEach(artisanId => {
        artisans.push(factory.newRelationship(namespace, 'Artisan', transaction.artisanIds));
    });
        product.artisans = artisans;
    product.owner = factory.newRelationship(namespace, 'Artisan', transaction.ownerId);
    product.design = factory.newRelationship(namespace, 'ProductDesign', transaction.designId);

    await prodRegistry.add(product);

        // create the product history
    const history = factory.newResource(namespace, 'ProductHistory',  transaction.transactionId);
    history.timestamp = transaction.timestamp;
    history.serial_number = transaction.serial_number;
    history.transaction = transaction.$type;
    history.personInvoking = getCurrentParticipant();
    history.product = product;

         const historyRegistry = await getAssetRegistry('org.trade.com.ProductHistory');
    await historyRegistry.add(history);

    // emit event
    const productInfo = factory.newEvent(namespace, 'ProductEvent');
    productInfo.product = product;
    emit(productInfo);

}
```

4.5 Backend

The client application interacts with the smart contract using a back-end application built in NodeJS. Hyperledger composer comes with a tool called composer-rest-server which helps

us to auto generates a restful API that is based on loop-back technology, a NodeJS framework. The restful API contains all the CRUD operations for interacting with our smart contract. Because our application is a permissioned blockchain application, the smart contract can only accept requests from participants whose identity is known on the network. Therefore, each participant has to register on the network to get a private key which is stored on the database connected to the server (figure 14.0). When a participant sends a transaction proposal, its signed with the user's private key stored on the database.

4.6 Discussion.

Every technology has its own advantages and disadvantages and in this section we shall discuss the impact of using Hyperledger fabric as the blockchain platform in our proof-of-concept application and finally we shall draw systematic comparisons between our proof-of-concept application and that of [5].

Querying for the handcraft provenance data

To get the provenance data regarding the handcraft token, it requires querying for all the blocks that contain handcraft token. At the time of writing this paper, there was no possible way for querying the blocks using the handcraft token identifier. The only possible way around this, was to query the Hyperledger composer historian; a specialized registry that records all successful transactions[12]. Although we can perform queries on the Hyperledger composer historian record to retrieve all the transactions, it is not possible to query for transactions that are specific to a particular handcraft token.

```
41
42  query getProductHistory {
43    description: "returns a list of all transactions"
44    statement:
45      SELECT org.hyperledger.composer.system.HistorianRecord
46        WHERE (transactionType == 'CreateProduct'
47          OR transactionType == 'CreateProductDesign' OR transactionType == 'AddQualityCertificate'
48          OR transactionType == 'AddValueAddition')
49  }
```

As seen in the code snippet, we can only write queries that can filter the transaction blocks based on the transaction id, transaction type. This returns huge amounts of data which will again require manual filtering to get transaction which are specific to a particular handcraft token and consequently making the queries inefficient.

Impact of using NoSQL Database as State database

One of the reasons for choosing Hyperledger fabric as the blockchain technology was the fact that it can use couchDB, a NOSQL database as the state database. The advantage of using a NOSQL database compared to a key-value type of database is that it's more versatile and it can support complex queries. However, it still has a number of limitations when used as a state database. Some of the challenges include:

1. The queries are read only and cannot be used to update the state database.
2. Some SQL commands such as LIMIT, SKIP AND JOIN are not supported.
3. It does not support indexing. It's important to note that CouchDB as database supports indexing when used as a standalone database, however, at the time of writing this paper it was not possible to index the database when writing smart contract using Hyperledger composer.

[12] https://hyperledger.github.io/composer/v0.19/business-network/historian

Impact of Storing User certificates on a central Database.

Since our proof-of-concept is a permissioned application, it stores the users with their corresponding private certificate on a database connected to a server. The purpose is to allow a single organization node to be used by multiple users with each user having their own identity on the network. However, if the database is compromised by malicious users, then the user private certificate can be stolen.

Impact of running the business logic on the Smart contract

We decided to run the application business logic on the smart contract because its more secure and temper resistance. However, this increases the processing time when compared with processing on the central server. The delay in the processing speed is caused by the time taken by the peers to validate the transaction. Because of the delay, transaction request from the client to the server may timeout. To avoid timeout of the transaction we use Web-Sockets to send response from the smart contract to the server and back to the client.

Ethereum based provenance application vs Hyperledger fabric-based provenance application.

In the table below we summarize the major differences between our proof-of-concept application and that developed by [5] where he uses Ethereum as the blockchain platform.

Table 1. difference in the application

Ethereum based provenance application	**Hyperledger fabric-based provenance application**
Uses blockchain as back up for the data stored on the central database.	Uses blockchain as the single source of truth.
data can be accessed by anyone connected on the network.	Only registered members on the network can have access to provenance data.
Requires crypto currency to perform transactions on the smart contract	Does not require crypto currency

5 Conclusion

This thesis had two main goals. First, we wanted to find out the capability of using a NoSQL database in Hyperledger fabric. From our findings, we concluded that using a NoSQL database like CouchDB can help to reduce the response time of a transaction by allowing Smart contract to quickly access data in the blockchain, but it suffers from a number of limitations notably it's limited querying capabilities discussed in section 4.6. It is therefore not the best choice for a database, a more expressive SQL database would be more situated. Investigating the best alternatives is however beyond the scope of our study and has been recommended for further studies.

The second goal of this paper was to find out the impact of running the application business logic onto the smart contract. From our findings, Smart contract in Hyperledger have the capacity to run an application business logic of any kind since they can be written in normal programming languages like Golang, Java and JavaScript. However, the downside to running business logic onto the Smart contract is that it takes more time to process the transactions because all the responsible peers have to validate the transactions.

To conclude, the result of this paper is a permissioned blockchain application that can be used to track the provenance of handcraft jewellery.

One of the challenges faced during implementation of our proof-of-concept application was that, it was not possible to query for specific blocks that contained a particular asset using the asset id or property of the asset. Future works should consider coming up with more efficient way that can be used to query the transaction history of an asset without querying all the transactions blocks. Also, future works should consider integrating a more expressive SQL database inside blockchain architecture.

6 References

[1] S. A. a. M. R. P. Abeyratne, “Blockchain ready manufacturing supply chain using distributed ledger,” 2016.

[2] X. a. S. S. a. T. D. a. K. C. a. K. K. a. N. L. Liang, “Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability,” in *Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing*, 2017, pp. 468--477.

[3] "provenance.org," [Online]. Available: https://www.provenance.org/tracking-tuna-on-the-blockchain. [Accessed 01 November 2018].

[4] A. Ranganathan, “The artisan and his audience: Identification with work and price setting in a handicraft cluster in Southern India,” *Administrative Science Quarterly*, vol. 63, pp. 637--667, 2018.

[5] A. Orengo, "University of Tartu, Institute of Computer Science Graduation Theses Registry," [Online]. Available: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=62203&year=2018. [Accessed 20 August 2018].

[6] H. M. a. L. M. Kim, “Toward an ontology-driven blockchain design for supply-chain provenance,” pp. 18-27, 2018.

[7] Z. X. S. D. H.-N. C. X. a. W. H. Zheng, "Blockchain challenges and opportunities:asurvey," *Internationa Journal of Web and Grid Services*, vol. 14, pp. 352-375, 2018.

[8] S. a. o. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[9] Z. a. X. S. a. D. H. a. C. X. a. W. H. Zheng, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557--564.

[10] A. a. E. A. a. V. T. a. L. A. Azaria, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016.

[11] E. B. A. B. V. C. C. C. K. D. C. A. E. D. F. C. L. G. M. Y. a. M. S. Androulaki, "Hyperledger fabric: a distributed operating system for permissioned blockchains," 2018.

[12] T. a. S. M. a. E. P. a. S. A. a. G. B. Hepp, "On-chain vs. off-chain storage for supply- and blockchain integration," *it-Information Technology*, vol. 60, pp. 283--291, 2018.

[13] N. a. B. M. a. C. T. Atzei, “A survey of attacks on ethereum smart contracts (sok),” in *International Conference on Principles of Security and Trust*, 2017, pp. 164--186.

[14] G. a. o. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1-32, 2014.

[15] N. Leavitt, “Will NoSQL Databases Live Up to Their Promise?,” *Computer*, vol. 43, pp. 12-14, 2010.

[16] R. S. G. a. N.-F. Neisse, "A blockchain-based approach for data accountability and provenance tracking," in *12th International Conference on Availability, Reliability and Security*, 2017.

[17] T. a. R. B. B. a. S. T. a. S. B. Bocek, “Blockchains everywhere-a use-case of blockchains in the pharma supply-chain,” in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 772--777.

[18] S. a. R. H. a. D. M. a. D. Bragagnolo, "Ethereum query language," 2018.

[19] The Linux Foundation projects, “Hyperledger,” [Online]. Available: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf. [Accessed 10 October 2018].

[20] bitsonblocks.net, "Bits on Blocks," [Online]. Available: https://bitsonblocks.net/2016/10/02/gentle-introduction-ethereum/. [Accessed 02 August 2019].

7 Appendix

7.1 License

1. Non-exclusive licence to reproduce thesis and make thesis public

   I, Ivan Ojiambo

   herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   A A Fully Data On-chain Solution for Tracking Provenance of Handcrafted Jewellery Data On-chain Solution for Tracking Provenance of Handcrafted Jewellery

   (*title of thesis*)

   Supervised by Luciano Garcia-Banuelos, PhD

   (*supervisor's name*)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Ivan Ojiambo

14/08/2019