A Blockchain-Based Fake Product Identification System

Yasmeen Dabbagh, Reem Khoja, Leena AlZahrani, Ghada AlShowaier, and Nidal Nasser

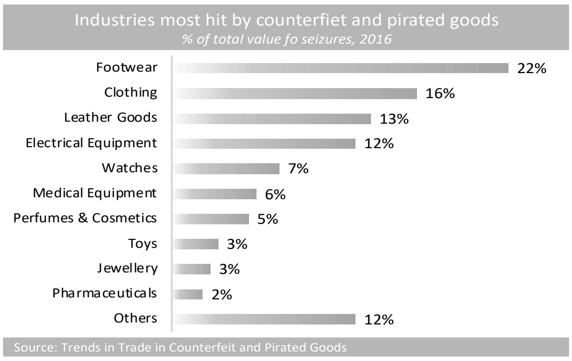
College of Engineering, Alfaisal University, Riyadh 11533, Saudi Arabia

***Abstract*—Counterfeit goods have become a global problem as consumers are being deceived into buying unauthentic goods with no way to validate the authenticity. Recently, blockchain has become more popular as it fosters trust between untrusting participants. This paper uses blockchain technology to combat the sale of counterfeit products. We use blockchain to allow manufacturers to add authentic product serial numbers onto the ledger; consumers can then use the serial number to verify the authenticity of a product before purchasing it. Blockchain plays a pivotal role in ensuring that data was not tampered with – creating a trusted environment.**

Keywords*—*blockchain, counterfeit.

I. Introduction

By definition, a counterfeit product is a low-quality imitation of an original item – the aim is mainly to mimic a luxurious product at a cheaper price to allow consumers to save money. Nowadays, the quality of these items has become almost entirely identical to the original product. According to the Organization for Economic Co-operation and Development (OECD), the global trade in counterfeit goods has risen steadily over the last couple of years and now represents 3.3% of global trade. [1]. Counterfeiting revenue takes away revenue from original brands, and depending on the type of goods (i.e., medicine and beauty products), can seriously harm the health of consumers.



Source: Trends in Trade in Counterfeit and Pirated Goods

Fig. 1: Percent of seized counterfeit goods in each industry

Many online retailers are investing in solutions to reduce the number of counterfeit products found on their websites. For instance, Amazon has reportedly blocked ten billion attempts to list counterfeit products and destroyed two million fake goods in its warehouse after starting an initiative called Project Zero [2]. Project Zero is a machine-learning technology that removes flagged products as potentially fake. Amazon has spent over $700 million and recruited 10,000 individuals to protect their website from fraud [2].

Within the European Union, nearly 1 in 10 consumers have been deceived into buying a counterfeit product that they thought was a genuine, authentic item, and 33% stated that they were skeptical of the authenticity of their purchase [3]. Counterfeit products have made consumers more resistant to buying goods online due to a lack of trust. In turn, authentic brands will be affected as they lose business. On the other hand, uninformed consumers will unintentionally help counterfeit brands gain profit while reducing profits and success of authentic brands.

These issues have created an immense need for a reliable way to authenticate products before being bought, especially if they are being exchanged between consumers (second-hand). Today, blockchain technology provides a reliable solution to provide a platform of trust between consumers. By creating a distributed ledger with consensus, provenance, immutability, and finality, to allow consumers to ensure the authenticity of their products, sales of counterfeiting products will be significantly reduced while allowing consumers to shop without uncertainty.

The rest of this paper is organized as follows. In Section II, we review the literature. In Section III, we define our problem statement. In Section IV, we describe our system. In Section V, we conclude our paper.

II. Literature Review

A. Blockchain Overview

Blockchain is a decentralized, shared, replicated, and permissioned ledger that establishes trust between entities as blocks, which are committed to the ledger, cannot be modified or removed. This allows a continuous track of transactions that have occurred on an asset. Cryptography is also implemented to ensure the privacy of the data by only allowing relevant participants to see parts of the ledger that are relevant to them [4].

Immutability is achieved by the connection between blocks in a single blockchain application. Each block is appended to the end and includes the previous block's hash (Fig. 2). A hash is a mathematical function that converts an input (of any size) to a hash (of fixed size) to encrypt the data; the decryption of a hash is nearly impossible as the algorithms use hash lengths of 256-bits or more. Therefore, one must have the initial input to recalculate the hash and compare if the input maps to the stored hash. Within the block, transactions are included (a single block can have zero to n transactions). If a transaction within a block has been tampered with, the hash value will change, and the next block will no longer be chained as there will be discrepancies between the hash values.

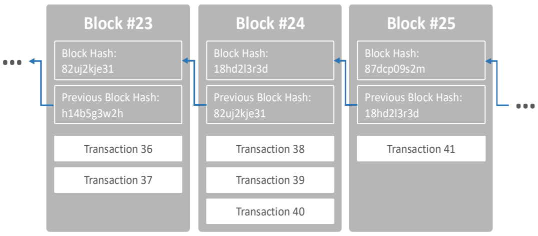


Fig. 2: Connection between blocks in a blockchain

2022 5th Conference on Cloud and Internet of Things (CIoT) | 978-1-6654-7964-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/CIOT53061.2022.9766493

978-1-6654-7964-6/22/$31.00 ©2022 IEEE

B. Blockchain Characteristics

A blockchain's characteristics allow the establishment of a level of trust between all participants. Reference [5] defined the characteristics of a blockchain:

1) *Private, Public or Permissioned Blockchain:* the system of ledger sharing and permission of participants in a system depends on the type of blockchain. A **private** blockchain is a closed network with a predefined group of participants. These participants must be identified, verified, and authorized to contribute and use the system. **Public** blockchains (like Bitcoin) is open P2P network that allows anyone to contribute to the network by running the consensus mechanism. As it is a public network, anonymous and untrusted participants can join and leave the system at any time. A **permissioned** blockchain is a combination of both private and public blockchains. The main participants are identified, verified, and authorized. An additional layer of access control is involved, allowing only certain participants to perform defined actions. All three types have specific characteristics in common in that they all operate on P2P networks, multiple nodes maintain the ledger, and data is stored on blocks in the blockchain.
2) *Decentralization:* Each blockchain entity is responsible for verifying and storing the transactions on their ledger. No third party is responsible for doing so.
3) *Validity:* each block is validated by other nodes. There is no requirement that each node must execute the validation. Therefore, the falsification of any transaction will be detected.
4) *Auditability:* all transactions are recorded and timestamped. Therefore, it is easy to trace an asset's movements and previous records.

C. Hyperledger Fabric Consensus Implementation

Reference [4] describes the consensus implementation by completing three phases: endorsement, ordering, and validation. The three-phase model (Fig. 3) allows the distribution of untrusted code in an untrusted environment.

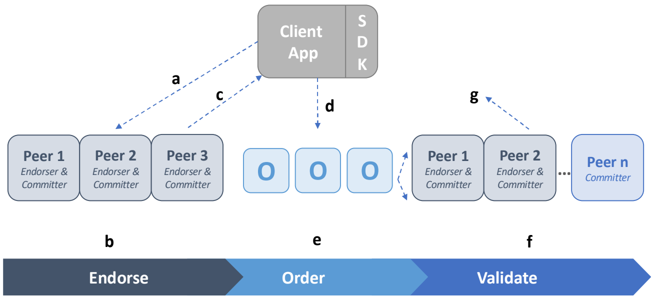

Client App
SDK
a
c
d
g
Peer 1 Endorser & Committer
Peer 2 Endorser & Committer
Peer 3 Endorser & Committer
O
O
O
Peer 1 Endorser & Committer
Peer 2 Endorser & Committer
...
Peer n Committer
b
e
f
Endorse
Order
Validate

Fig. 3: Transaction lifecycle

1) *Phase 1:* Endorsement
   a) A client submits a proposed transaction to selected endorsing peers based on the endorsement policy.
   b) The endorsing peers will run a simulation of the proposed transaction against the smart contract to check that all rules defined in the smart contract are met. The endorsing peers will then capture the RW set; the read set shows the current data in the world state database, and the write set shows what the data will be after the transaction is completed. The endorser then signs the transaction proposal.
   c) Once the endorser has signed the transaction, it is sent back to the client along with the RW set. The client should ensure that all endorsers send the same RW set to continue the process.
2) *Phase 2:* Ordering
   d) The client then sends endorsers' responses and the transaction to the orderer.
   e) The orderer collects the transactions in sequence and places them in blocks to be transmitted to the peers of the channel.
3) *Phase 3:* Validation
   f) Committing peers receive the blocks and validate the various transactions before committing the block to the ledger. The use of the Validation System Chaincode (VSCC) helps the process. The RW set and signature are checked, and two outcomes are possible: 1) the RW set and signature is valid in which the committer will commit the changes to the world state database and add the transaction to the ledger, or 2) the RW set and signature is not valid in which the committer will add the transaction to the ledger but will not update the world state database.
   g) Committing peers will then notify applications about the success or failure of a transaction.

III. Problem Statement

A fake product identification system is long overdue. Consumers need a single, reliable platform that can authenticate a luxurious product anywhere in the world. This platform will help prevent consumers from paying the price of a "real product" when in fact they are getting a counterfeit. Our system will take on the role of being the leader in the industry of counterfeit identification systems.

Our objectives are to help preserve product authenticity - by storing the serial numbers of authentic products and the transactions that have taken place on each product - and having a single trusted source for authentication - the system will act as a single point of product authentication amongst consumers.

IV. System Model

A. *Network Participants:* our blockchain is composed of two main participants with the following main roles:

1) *Manufacturer:* adding new sellers' information to allow them to upload the sold pieces, adding the product information (serial numbers and product category/description) of all products that have been manufactured. The manufacturer can retrieve a list of all consumers that have bought a product from a seller.
2) *Seller:* update the sold product with the consumers' information and query all consumers that have previously bought from the seller. In addition, the seller can query a list of products that are in their possession to see a list of items they have for sale.

B. *Operation Flow:* Fig. 4 shows the sequence of events in our system as follows: the manufacturer is responsiblefor creating a list of sellers that can receive its products. After that, the manufacturer should add the products that are scheduled to be manufactured to the blockchain. Once the product has been manufactured, it is transferred to the corresponding seller to sell at his/her shop. If the seller sells the product to a consumer, they will be required to update the product's information to include the buyer's information. Finally, consumers will be able to authenticate the product's authenticity by using the serial number or phone number to ensure the product is not a counterfeit. Each transaction will be saved on the blockchain to create the product's records.

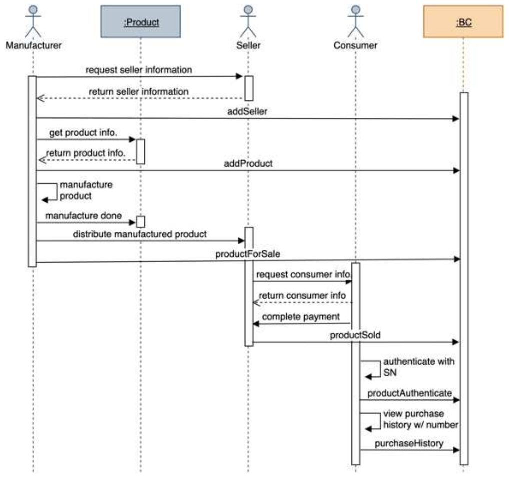


Fig. 4: Sequence diagram of our system

C. *Network Architecture:* Our system follows a three-layered architecture. Fig. 5 shows the three layers along with their components. In the presentation layer, the user runs a website to connect to the system using the application layer, which consists of a web server SDK that interacts with the data layer (the Hyperledger Fabric). The webserver SDK abstracts the complexity of all endorsement flows. The SDK is responsible for selecting the identity from the wallet, connecting to the gateway, accessing the network channel, selecting the smart contract, submitting the transaction, and processing the response or notification [4]. The final layer, or the data layer, includes the Hyperledger Fabric. Within the Hyperledger Fabric, two channels are created – C1 has the manufacturer and seller as the participants, while C2 creates a gateway to allow consumers to query from the ledger, connecting the seller and consumer to C2. The consensus implementation takes place in this layer.

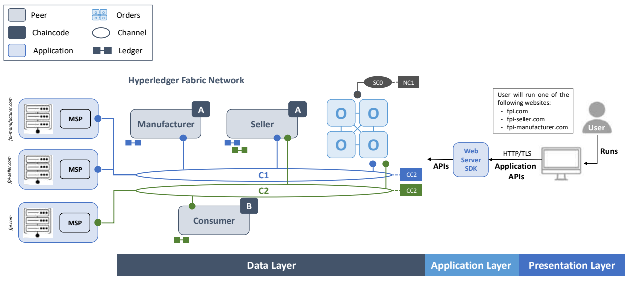


Fig. 5: Network architecture

D. *Data Management:* cloud-based storage will be used to save the database and ledger. A database will be used, using MySQL, and is responsible for saving all users' registration and log-in credentials. The ledger will be split into two parts:

1) *World state:* mutable data that can be deleted – the consumer will view this data.
2) *Blockchain:* blocks of hashed linked lists with immutable data that cannot be deleted and only accessible by the network participants.

E. *Smart Contracts and Chaincode:*

1) *addProduct Transaction:* a new product is added to the blockchain network by a manufacturer.
Smart Contract: once a product has been approved for manufacture, each will receive a unique serial number; once the serial number has been issued, the manufacturer must add the product to the system.

Algorithm 1 ADDPRODUCT

Input: productName, productSN, sellerCode, productBrand, productColor, productPrice, productSize

```
function addProduct (productName, productSN, sellerCode, productBrand, productColor, productPrice, productSize)
    this.productName ← productName;
    this.productSN ← productSN;
    this.sellerCode ← sellerCode;
    this.productBrand ← productBrand;
    this.productColor ← productColor;
    this.productPrice ← productPrice;
    this.productSize ← productSize;
    this.productStatus ← "MANUFACTURE";
    return product;
```

2) *addSeller Transaction:* a new seller is added to the blockchain network by a manufacturer.
Smart Contract: once a new point of sale has been established, the manufacturer will add the seller's information to their database to start sending products to be sold.

Algorithm 2 ADDSELLER

Input: sellerName, sellerBrand, sellerCode, sellerNum, sellerManager, sellerAddress

```
function addSeller (sellerName, sellerBrand, sellerCode, sellerNum, sellerManager, sellerAddress)
    this.sellerName ← sellerName;
    this.sellerBrand ← sellerBrand;
    this.sellerCode ← sellerCode;
    this.sellerNum ← sellerNum;
    this.sellerManager ← sellerManager;
    this.sellerAddress ← sellerAddress;
    return seller;
```

3) *querySellers Transaction:* view all or selected sellers and their information; performed by manufacturer.

Algorithm 3 QUERYSELLERS

```
Input:
function querySellers()
seller[] += get (sellerCode);
return sellers;

Input: sellerCode
function querySeller(sellerCode)
seller ← get (sellerCode);
return seller;
```

4) *queryConsumers Transaction:* view all or selected consumers and what they bought; performed by manufacturer or seller.

Algorithm 4 QUERYSCONSUMERS

```
Input:
function queryConsumers()
get (sellerCode);
return consumers;

Input: sellerCode
function queryConsumers (sellerCode)
get (sellerCode);
return consumers;
```

5) *productForSale Transaction:* a product is now in the hands of a seller, performed by the manufacturer.
<u>Smart Contract:</u> once the manufacturer has completed the production of a product, they must choose which seller shall receive the product and change the status to "for sale" with the seller's information.

Algorithm 5 PRODUCTFORSALE

```
Input: productSN
function productForSale (productSN)
get(productSN);
product.productStatus ← "FOR SALE";
put(product);
return product;
```

6) *queryProducts Transaction:* view all products in possession of a seller for sale; performed by the seller.
<u>Smart Contract</u>: a seller can query the products in their inventory that are available for sale from the manufacturer.

Algorithm 6 QUERYPRODUCTS

```
Input: sellerCode
function queryProducts(sellerCode)
products [] += get(SellerCode);
return products;
```

7) *productSold Transaction:* a product has been sold and is now owned by a consumer; performed by the seller.
<u>Smart Contract:</u> once a seller completes a sale, they must change the product status to "sold" and add the consumer's information.

Algorithm 7 PRODUCTSOLD

```
Input: productSN, consumer
function productForSale (productSN, consumer)
get(productSN);
product.productStatus ← "FOR SALE";
put(product);
return product:
```

8) *productAuthenticate Transaction:* a serial number is used to check if a product is real; performed by the consumer.
<u>Smart Contract:</u> a consumer can insert the serial number of a product to view whether it is authentic and the status (bought – available).

Algorithm 8 PRODUCTAUTHENTICATE

```
Input: productSN
function productAuthneticate (productSN)
get(productSN);
if (productSN == found)
return true;
else
return false;
```

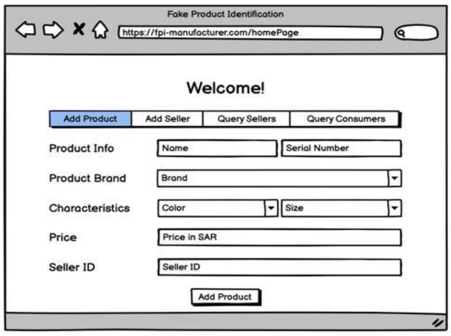
9) *purchaseHistory Transaction:* a serial number is used to check if a product is real; performed by the consumer.
<u>Smart Contract:</u> a consumer can view a person's purchase history using their phone number; this is useful in case of secondhand selling.

Algorithm 9 PURCHASEHISTORY

```
Input: phoneNum
function purchaseHistory (phoneNum)
get(phoneNum);
return product;
```

F. *Client Application:* The application will be implemented on a website. Each participant will have a different website, as well as a different website for the user. Each website will have different functionalities that can be implemented, creating individual websites makes it easier to add more functionalities in the future without affecting other groups of participants:

A. *Manufacturer:* as soon as a manufacturer is logged in, they can add new sellers by entering the information needed such as (name – brand – location – phone number – email). As well as adding products that will be manufactured including (name – color – serial number – brand – size (if applicable)) – as well as the seller's information that will be receiving the product (Fig. 6). Finally, the manufacturer can query two different lists: sellers and consumers (Fig. 7).



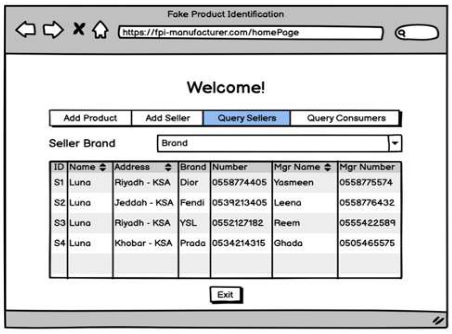
Fake Product Identification

https://fpi-manufacturer.com/homePage

Welcome!

Product Info	Name	Serial Number
Product Brand	Brand	
Characteristics	Color	Size
Price	Price in SAR	
Seller ID	Seller ID	

Fig. 6: Manufacturer Application 1



Fake Product Identification

https://fpi-manufacturer.com/homePage

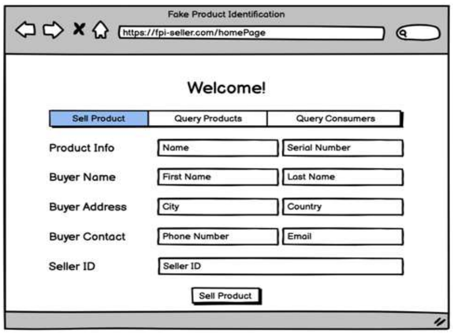
Welcome!

Seller Brand: Brand

ID	Name	Address	Brand	Number	Mgr Name	Mgr Number
S1	Luna	Riyadh - KSA	Dior	0558774405	Yasmeen	0558775574
S2	Luna	Jeddah - KSA	Fendi	0539213405	Leena	0558776432
S3	Luna	Riyadh - KSA	YSL	0552127182	Reem	0555422589
S4	Luna	Khobar - KSA	Prada	0534214315	Ghada	0505465575

Fig. 7: Manufacturer Application 2

B. *Seller:* once logged in, the list of products available for sale will show up. Once a product is sold, the seller will update the status to "sold" and enter the consumer's information (Fig. 8). The seller can also query the list of consumers (Fig. 9).



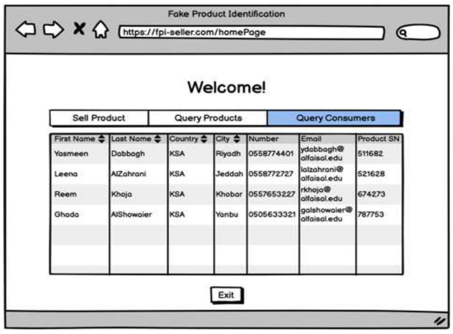
Fake Product Identification

https://fpi-seller.com/homePage

Welcome!

Product Info	Name	Serial Number
Buyer Name	First Name	Last Name
Buyer Address	City	Country
Buyer Contact	Phone Number	Email
Seller ID	Seller ID	

Fig. 8: Seller Application 1



Fake Product Identification

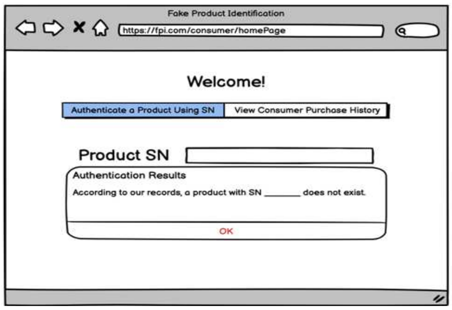
https://fpi-seller.com/homePage

Welcome!

First Name	Last Name	Country	City	Number	Email	Product SN
Yasmeen	Dabbagh	KSA	Riyadh	0558774401	ydabbagh@alfaisal.edu	511682
Leena	AlZahrani	KSA	Jeddah	0558772727	lalzahrani@alfaisal.edu	521628
Reem	Khoja	KSA	Khobar	0557653227	rkhoja@alfaisal.edu	674273
Ghada	AlShowaier	KSA	Yanbu	0505633321	galshowaier@alfaisal.edu	787753

Fig. 9: Seller Application 2

C. *Consumer:* after logging in, the consumer will have two options to pick from: authenticate an item from a serial number (Fig. 10) or view purchase history from a phone number (Fig. 11).



Fake Product Identification

https://fpi.com/consumer/homePage

Welcome!

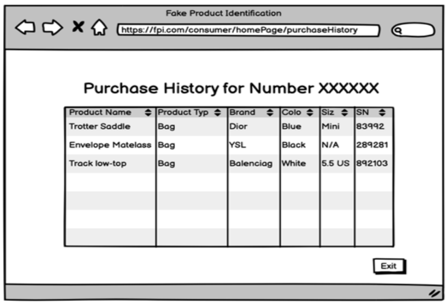
Authenticate a Product Using SN
View Consumer Purchase History

Product SN

Authentication Results

According to our records, a product with SN _______ does not exist.

Fig. 10: Consumer Application 1



Fake Product Identification

https://fpi.com/consumer/homePage/purchaseHistory

Purchase History for Number XXXXXX

Product Name	Product Typ	Brand	Colo	Siz	SN
Trotter Saddle	Bag	Dior	Blue	Mini	83992
Envelope Matelass	Bag	YSL	Black	N/A	289281
Track low-top	Bag	Balenciag	White	5.5 US	892103

Fig. 11: Consumer Application 2

V. Conclusion

This paper uses blockchain to combat the sales of counterfeit products. We have two main participants that dictate the blockchain transactions: the manufacturer and seller. The manufacturer adds the sellers and products that are authentic and real. As the seller sells products, the products are updated to sold with the buyer's information. This ensures that a consumer owns an authentic product in case he or she wants to resell the product. In the future, the application will incorporate consumers by allowing them to sign, using their digital signature, the purchase of a product to enhance the security further. Furthermore, the logic of our application can be used for different industries, such as pharmaceuticals or cosmetics, to further combat counterfeit products.

Acknowledgment

The authors would like to thank Dr. Nidal Nasser for his continuous support throughout the project. The authors would also like to thank Alfaisal University for funding and supporting this work through Eng. Mohammed Alagil Scientific Conference Participation Support Award 2022.

References

[1] OECD. (2019, March 18). *Trade in fake goods is now 3.3% of world trade and rising.* Retrieved from OECD: https://www.oecd.org/newsroom/trade-in-fake-goods-is-now-33-of-world-trade-and-rising.html

[2] Segran, E. (2021, May 17). *'The volume of the problem is astonishing': Amazon's battle against fakes may be too little, too late.* Retrieved from Fast Company: https://www.fastcompany.com/90636859/the-volume-of-the-problem-is-astonishing-amazons-battle-against-fakes-may-be-too-little-too-late

[3] TFL. (2021, June 14). *Nearly 1 in 10 EU Consumers Have Mistakenly Purchased a Counterfeit Product Over the Past Year, Per Report.* Retrieved from TFL: https://www.thefashionlaw.com/nearly-1-in-10-eu-consumers-have-mistakenly-bought-a-counterfeit-product-over-the-past-year-per-report/

[4] IBM Corporation. (2018-2019). *IBM Training*. IBM.

[5] Viriyasitavat, W., & Hoonsopon, D. (2019). Blockchain characteristics and consensus in modern business processes. Journal of Industrial Information Integration, 32-39.