Research Report

Interplanetary Textile Passport

30-05-2022

Technical Advisor: Marcio Fuckner - m.fuckner@hva.nl
Coach: Pieter Leek - p.d.leek@hva.nl
Client: Troy Nachtigall - t.r.nachtigall@hva.nl

Students:
Anthony de Jong - 500847075 - anthony.de.jong@hva.nl
Andreh Almoussa - 500858870- andreh.almoussa@hva.nl
Shakira Hadjih - 500833588 - shakira.hadjih@hva.nl
Angelique Roelofsma - 500802837 - angelique.roelofsma@hva.nl
Marco Arana Barrantes - 500801368 - marco.arana.barrantes@hva.nl

Preface

This project was part of the minor Blockchain at the Amsterdam University of Applied Sciences. It was commissioned by Troy Nachtigall. The project had a span of 12 weeks, it started on the 8th of march and ended on the 31st of may.

Our project team consists of Andreh Almoussa (Software Engineer student), Anthony de Jong (Cyber Security student), Shakira Hadijh (Communication and Multimedia Design student), Marco Arana Barrantes (Game Development student) and Angelique Roelofsma (Business IT & Management student).

Index

Our Project

The mission of our project is giving clothing an opportunity to show their own story. With this opportunity, people will be able to trace back the story of their clothing piece with the use of the Interplanetary Textile Passport (ITP).

How will it work?

The passport will be stored on the blockchain; it consists of multiple NFT's that are connected to each other by an NFT with the ERC 988. It will be viewable through an ID-token that will be placed in the clothing piece. After filling in the ID-token on the web application, the application will view a summary of the passport, here users will see where it's made, who designed it and more!
This project will affect groups of people. These stakeholders are fashion designers, fashion brands, clothing manufacturing and fashion enthusiasts.

In the case of the clothing manufacturing, fashion designers and fashion brands, they will be credited for their work in a different way then they do now. Nowadays the credits are a brand logo on the clothes and the label inside the clothing. Our way of crediting will be through the clothing page. There will be much more space to have a story about that piece of clothing and about the brands and designers.

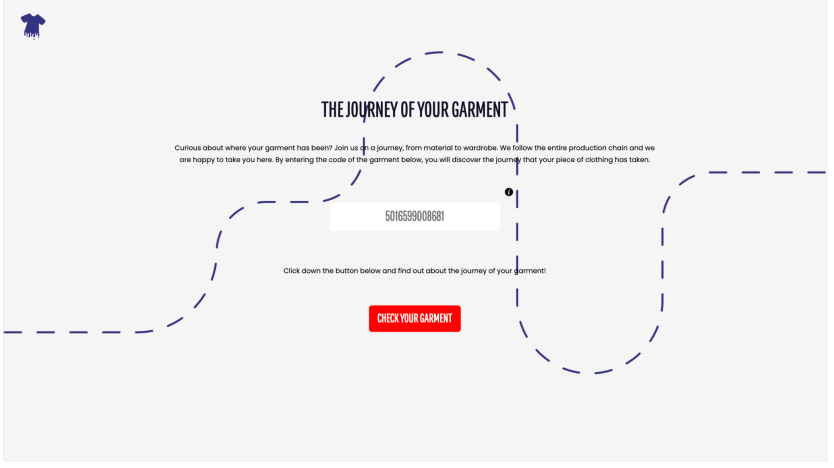
Fashion enthusiasts are also stakeholders, they will have way more information about their clothing. It will be more exciting to own clothing for them with the ITP added.

How will it look?

The passport will be viewable on a web application. The interface starts with a screen where users have to type in the token ID, which is found on the tag of the piece of clothing.

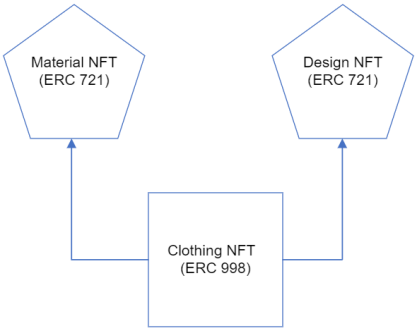
The team worked out the path of a normal day to day user, who will not want really specific information like weight of cotton etc, but it is info that is put into the NFT and can be accessible. This will be described in the chapter "future possibilities".

So, the user will fill in the code and will be led to an overview of the product. Here the product is shown and from here on they can zoom into the product; design and material. So they can track where the clothing has been, zoomed in on the specific NFT.



Link to design: https://xd.adobe.com/view/2e86b976-d230-43b2-baa1-4201979d6b3d-7bd5/

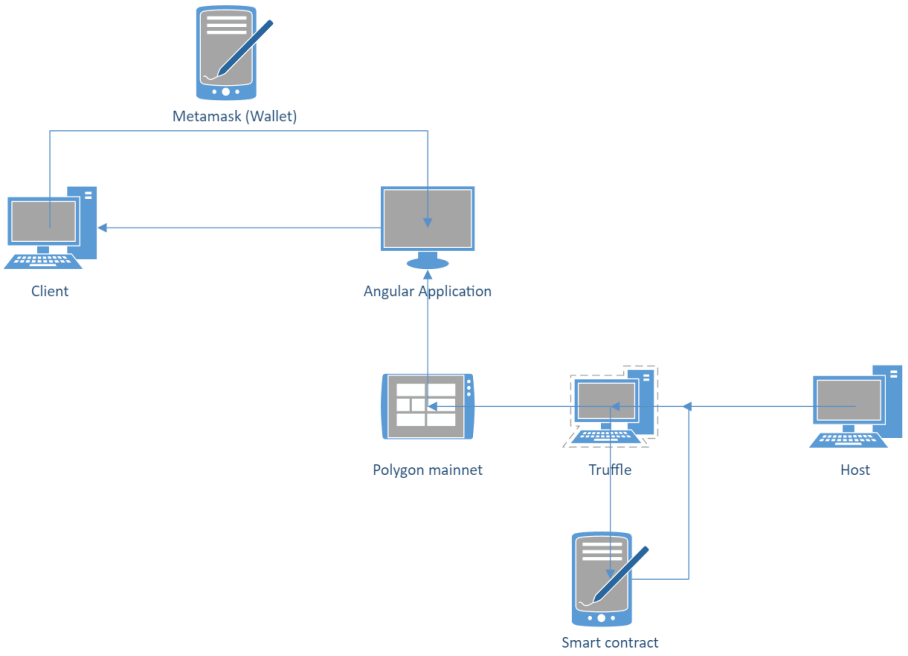
Interplanetary Textile Passport (ITP)



The passport is created using 3 NFT's : the Design(Brand) NFT, the Material NFT and the Clothing NFT. The structure of these NFTs together can be viewed on the right. The NFT's will all be a NFT with the ERC 721 except for the Clothing NFT, that NFT will be with the ERC 998.

There will be information stored in these NFTs that will be visible on the clothing page of the web application. For the material NFT that information is the description of the NFT, the type of material, if the material is sustainable, the harvest information (time of harvest and country), the companies that worked on the materials and extra information about the materials. In the design NFT there is the description of the NFT, an image of the design (like a blueprint), the brand and the designers. The clothing NFT is composed of the material NFT and the Design NFT.

The image below describes how the infrastructure is set up in order to deploy the application. It starts off with the host using Truffle to deploy the composable NFT and the other NFTs onto the Polygon mainnet. The decision why the application is deployed on the Polygon mainnet compared to for example the Ethereum mainnet is primarily down to several reasons. The major reason to go for Polygon was because



Polygon has one of the lowest token fees and gas fees, which makes the project cheap and sustainable over a long term. Polygon also provides quick transactions and is Layer-2 Ethereum, which means that the NFT transactions will be fast and it's close to Ethereum so it's easier to develop with because it's more familiar. When the client tries to connect with the Angular application, the client's Metamask wallet will ask to connect to the Angular application. When the client is connected with his/her wallet, is when the application will show the ITP of the client in all its glory.

The code of the ITP was fully written by our project team. It is based on the ERC721 and ERC 998. On the right there is a piece of our code, in this piece there is a function of the material NFT contract. It solves a problem of material quantity amortisation when the function 'safeTransferFrom' is executed in ERC 721. That was not possible as the number of parameters in safeTransfer cannot be changed. This is due to the function being overwritten from the ERC721, which means that it cannot be modified, as it is an essential function that many other functions and contracts depend on. A parameter in the function 'safeTranferFrom' called ' _data' of the type bytes opens the option to insert multiple inputs into it in the form of bytes, and here comes the job of the function 'extractMaterialAmountAndTokenId()'. It is fed with data in which there are multiple inputs, and then it splits the data in two, the first is the Token ID required, and the other is the 'AMOUNT'. It returns them both as Solidity allows two types of inputs. The function efficiency is pretty good, ''O(N) N' is the length of 'data bytes'. The array splits the data apart in a loop.

```
function extractMaterialAmountAndTokenId(bytes memory data)public pure returns(bytes memory,  bytes memory){

  require(data.length%2==0);

  uint datalen = data.length;
  uint stopCondition = data.length/2;

  bytes memory tokenIdArray = new bytes(data.length/2);
  bytes memory materialAmountArray = new bytes(data.length/2);
  uint count = 0;
  uint countdown = data.length/2;
  while(datalen != stopCondition){
    tokenIdArray[count] = data[count];
    count++;
    materialAmountArray[countdown-1] = data[datalen-1];
    datalen--;
    countdown--;
  }
  return(tokenIdArray,materialAmountArray);
```

This is a small piece of the whole code, the whole code can be viewed on Github. That can be viewed following this link: https://github.com/Adj2k22/Amfi.git

Approach

With this project, our team had breakthroughs. This chapter will list what the team has learned from this project and what our team made as a project group. The chapter will go through our successes based on when they happened on the timeline.

Dividing the Project into phases

Our group decided to split the project into phases in accordance with the number of sprints that were available. The phases would be; Defining, Design, Realisation, Testing, Delivery & Reflection.

Defining entails finding the main problem, defining it, and then setting up an objective to solve the main problem.

In the Design phase, the team came up with concepts to solve the main problem and present them to the client. The client will evaluate these concepts and subsequently approve or suggest improvements to these concepts.

The Realisation phase aims to realise a concept that the client has approved. This phase will take the longest out of all the phases to complete and requires extensive development time to work out a functional solution that meets the requirements set forth from the concept. This is also the phase where the team finds out the possible risks involved with the project in practice.

The testing phase aims to find out whether the product meets the functional standards imposed from the original concept. The team will attempt to answer questions like; Can all quality features be found in the product? Are the functional requirements met?

Finally, the Delivery & Reflection phase is where the team wraps up the project and formally delivers the final product along with its documentation to the client. This phase will also give the opportunity to reflect upon the team members and our performances.

Our timeframe for these phases were;

Sprint 1 would contain the Defining and Design phase.

Sprint 2 and 3 were set for the Realisation phase.

Sprint 4 contains the Testing and Delivery & Reflection phase.

Change of plans

As written above, the planning that the team intended would have been the optimal path that they wanted to take, however it changed as per our client's wishes and developments were made throughout the project.

Our Design phase took longer than planned due to our client's wishes and expressed features he would have liked to see. This extended the duration to sprint 3 before the project could actually manage to enter the Realisation phase.

During the transition from Design to the Realisation phase, the team was advised to use ERC998 by our Technical Advisor. This standard in theory could offer us the solutions to the objectives that the team was trying to achieve. But, there were many technical issues involved with this standard that took some time to figure out in order to make the project functional. More on this in the chapter Complications.

Complications

Every project has their setbacks, our team kept track of all of our failures, problems and changes of plan. This chapter will take through all of the complications this project had. Here it will go through each problem that the project has faced by the next points: 'What went wrong?', 'How was it supposed to work?' & 'Was it fixed or discontinued?'.

Outdated standards

Our first major hurdle was the ERC-998 standard itself. This standard was outdated and stuck in the draft process for 4 years with no improvements made sense. As a result, when the team got their hands on it, the contract was non-functional for the current version of solidity. Here it was also found out that if the contract were to work, it would require a tremendous amount of gas fees to deploy on the ETH network, a good 6.3 million Gwei. The team took this calculation into consideration with a considerable extra amount of gas fees and attempted to deploy the contract on remix, but each attempt failed.

Furthermore, the team figured out that the contract was unable to be deployed due to the fact it exceeded the maximum amount of 24 KB a contract can have to be deployed. The contract had to have a rework to reduce the size to below 24 KB by removing some obsolete functions and code. While not fully functional, the team was able to deploy the contract on our test network. Our concern, however, is that it's not sustainable due to the fact that this contract is over the 24 KB limit. However,

the team managed to fix it by removing unused code that contained ERC20 references or code.

Contract complications

Our contract was outdated, it originated from 2018 and was written for solidity version 4.x, the team on the other hand were working in solidity 8.x. In our attempt to migrate the contract to the current version of solidity by updating the version it was discovered that certain functions were no longer working. Our solution was to instead fix every error till there were none left in the current version of solidity. This did not work unfortunately due to wrong function output. So, it had to revert back to 4.x because of the fact that it only worked there. This contract had to be converted to the current version if the team wished to use it in the project. Because of that fact, a solution was created by using multiple truffle projects.

This works in the following manner; The composable supports the idea of getting called from other contracts, this can partially be used by the frontend. This combination is what enables the solution for our project. The partial usage is due to the fact that the authentication is within solidity. For example, when the method SafeTransfer is being called from a different contract it will fail because the caller in this case isn't the wallet owner anymore. The team attempted to fix this by delegating but this wouldn't work because it would have to reconfigure most of the things within ERC721, risks that come out of this is the potential of the ERC721 contract not being ERC721 anymore. Another risk that comes out of this is the potential of NFTs getting stolen.

This is why the team thought of doing it partially through the frontend as well as partially through the backend. Authentication will come from the frontend and everything else that doesn't require authentication will be in the backend of the contract.

ERC998 caused issues when attempting to use the safeTransferFrom function. This is due to the fact that it borrows this function from ERC721 but uses it in a different way. It would cause an invalid token ID for the ERC721 contract. This is due to the fact that ERC721 splits off the token ID and the amount when using this function. In order to fix this a condition has been put in place for using this function. Now when the caller is the contract, it will use safeTransferFrom from ERC721, else it uses the procedure that has been written to transfer the amount through ERC998.

Frontend configuration

Beside that, An issue kept appearing with using Angular and Web3 where every time the team tried to use the "npm" command, it would leave us with an error which specified that the libraries that the team tried to install were deprecated. This meant that the application could not be deployed on Angular with this issue. Eventually our

group discovered that the issue was caused by the different versions of libraries that the team was trying to use inside Angular. That meant luckily for us that this issue was easily resolved by going into the angular configuration to manually edit the Angular setting to allow older versions of these libraries.

Conclusion

The project goal was to give a clothing piece an interplanetary textile passport. The interplanetary textile passport gives the opportunity for fashion designers, brands and manufacturers to get credited in a whole new way. But it also gives an opportunity to fashion enthusiasts and environmentalists to view the whole process of a clothing piece.

The project goal was a success, but there is still a long way to go and to evolve for the interplanetary textile passport. The team has written the future ideas for this project in the next chapter: Future possibilities.

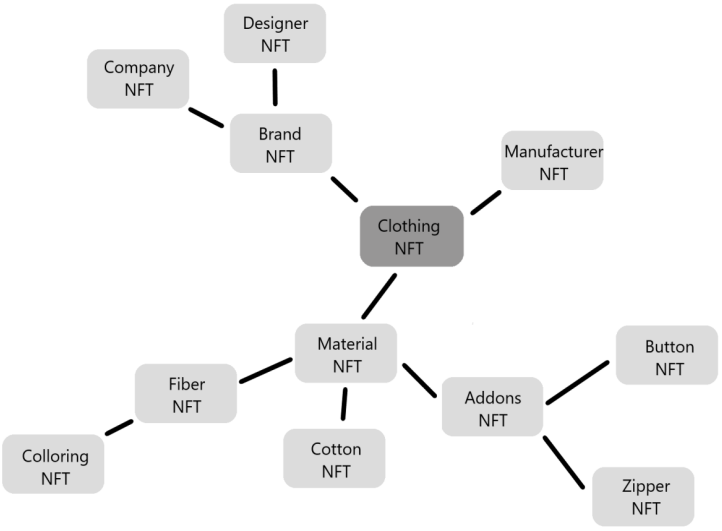
After all, this project gave a new view on an everyday item. It will change the way people can buy clothing pieces and society can make better decisions on the buying and how environmentally safe it is.

Future possibilities

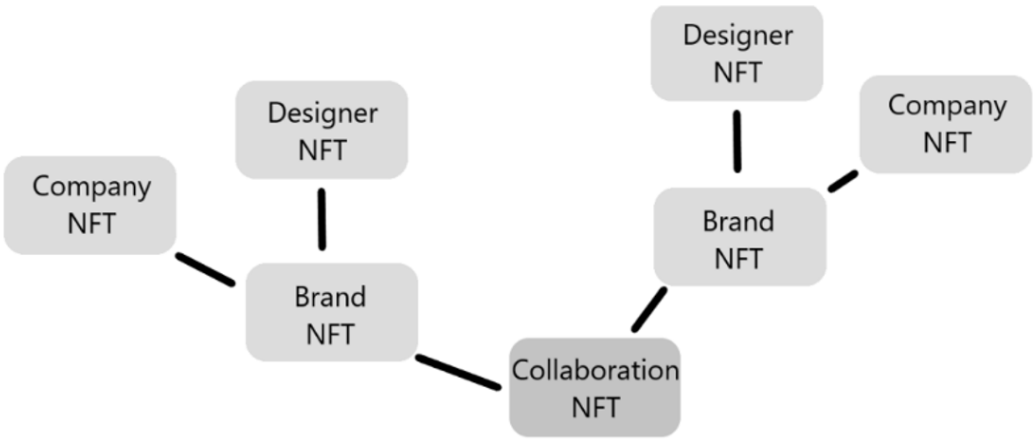
With such a new and promising project there are a lot of ways to use and improve later on as time goes by. This chapter will take you through some of the ideas for the future of this project.

Next up

The second step into this project is going deeper in the making of multiple NFT's. On the right there is how this step would look like. Here there are some new NFT's, this information is now inside the NFTs they are connected to. There is also a new NFT called Addons NFT, in which there can be multiple NFT's. In this example there is a Button and Zipper NFT.



In the future it will also be possible to make the Clothing NFT with multiple designers. In this example here there are two designers from one company. But it is also common that a clothing piece is designed by a collaboration between designers from different brands, like below.



Making every step in the process a NFT

Turning every component of an item into an NFT opens up possibilities of individual tracking and history of said component. However setting this chain up is a complicated process both in terms of coding and linking the physical goods to said NFT. Later on the possibility exists to combine the NFTs with physical goods through a QR code, for example on a cargo manifest for raw materials having the links to an NFT.

Different item passports

The interplanetary textile passport opens up a whole world of possible item passports on the blockchain. For example toys, parents can find out where the toys came from and what is really in it. This way they know what they give to their child.

There are so many opportunities for item passports. It is like a rolling ball, the start is with the interplanetary textile passport. After that success many can and will follow.

Future usability

Once the idea has gone mainstream and the interplanetary textile passport is a thing in daily life for consumers. Consumers will have much more freedom and choice in what they wish to purchase. The possibilities are in a wide range from interest in eco friendly materials in a product to buying vintage second hand products. The system can fit any type of consumer based on what they find important to know in a product to justify a potential purchase without being deceived or potentially having buyer's regret after owning something for a short period of time.

Consumers themselves will also have more power in regards to the products they own. It's the very nature of how the blockchain works, the consumer is effectively capable of keeping a highly accurate ledger up with their product along with the details of what they've potentially done to the product or just simply how they treated it in its life cycle.

One other major point of usability is the fact that goods can be tracked and recycled by companies. This can be an incentive for consumers by getting rewarded in the form of a deposit (like how people have to pay a small deposit for plastic bottles in a store) or some form of credit to buy new products. Aside from this it's good for the environment since ecological waste is reduced and the materials can be processed and reused as raw materials in a new product rather than creating them from scratch again. This can also be endlessly tracked on the blockchain as a

ledger which documents the cycle of a product, its materials and the way it is fabricated.

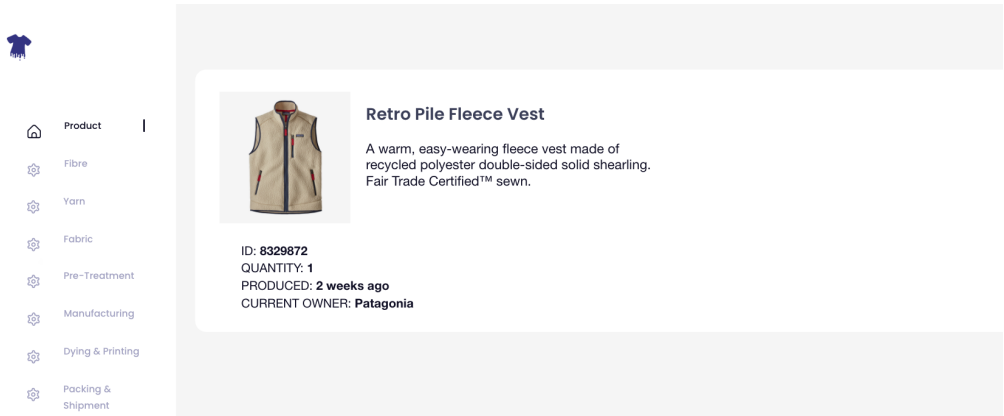
How the UI will look like

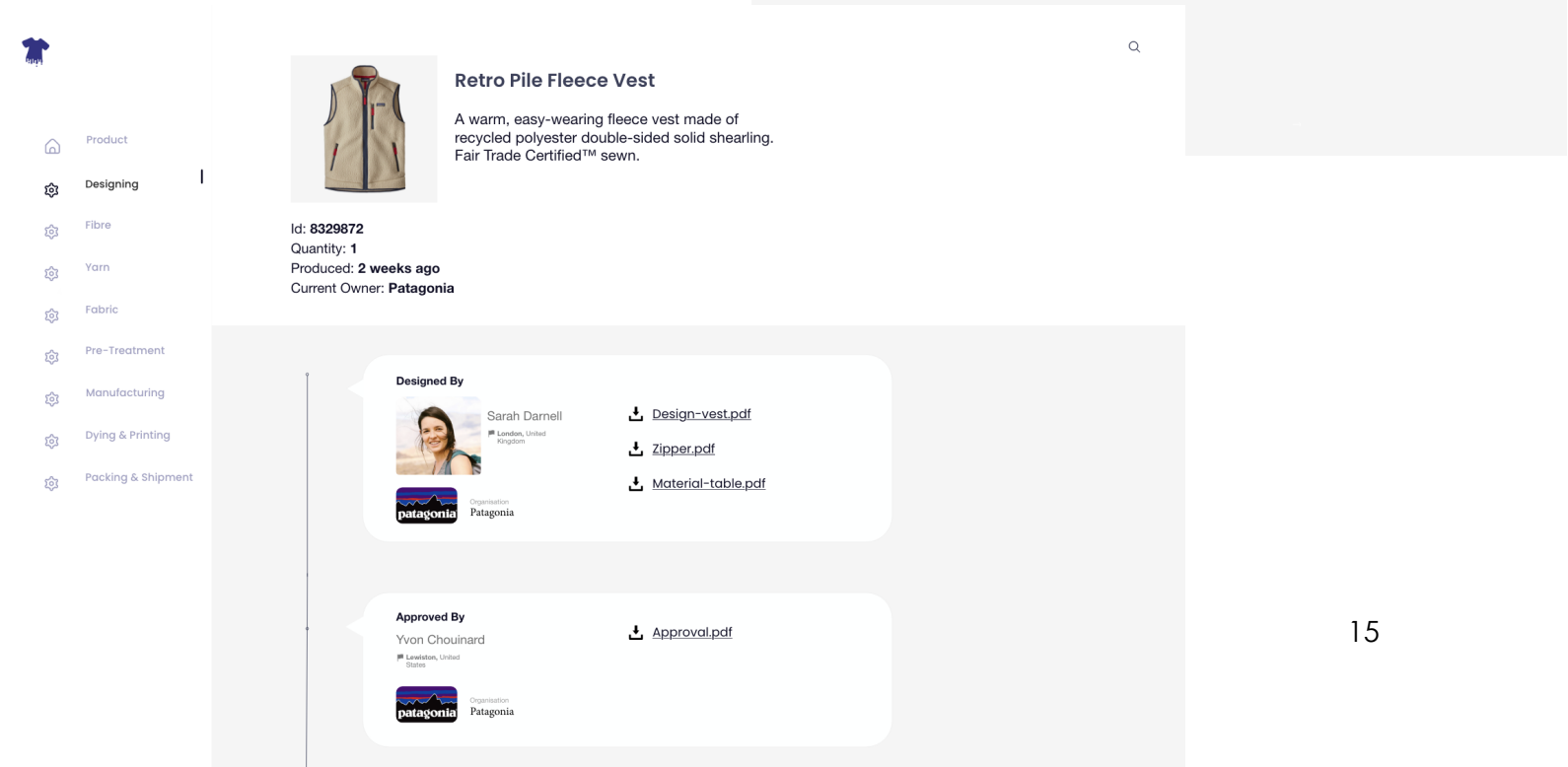
In this product passport the user can read the journey, impact and story behind the product. The goal of this project is to help share reliable information, so that the user as a customer can decide whether a product fits the user's values.

Since the process of a garment involves a lot of people/companies from all over the world, this can also benefit these people that work on the product, as everything is being tracked and the information is reliable. This could be of great value for a manufacturer, or designer or whomever needs reliable and trustworthy information about the product that they are working on.

For this project there was an UI made specifically for the customer, a normal day to day person, that does not need to know the same information as a manufacturer may want. So, the fashion brand can also build an interface where every little detail is being tracked and displayed.

Below there is a first design of how that will look like.





Proof Points

By means of quality marks and certification, a company can prove what their impact is on people and the planet. The future owner could also use these stamps in the UX, so the user can easily see if a piece of clothing fits their need (organic cotton, cruelty free, vegan, recycled etc.).

In a future design people can use these stamps; for instance, when the distributed ledger detects that there are no animal products used, the NFT gets a "vegan certification stamp".

Verified admits that an independent organisation has confirmed its accuracy. This is saved as a public, permanent report. If a user does not see verified, this may mean that the brand may have already provided proof for the quality mark, but this has not yet been confirmed by an independent organisation.

Attachments

Attachment 1 References

Roberts, J. J. (2021b, juni 9). 5 Ways Businesses Are Already Using Blockchains. Fortune. From: https://fortune.com/2017/08/22/blockchain-walmart-maersk-banking/

Uludag, M. (2022, 7 maart). Wat is minting? Crypto Benelux. From: https://cryptobenelux.com/2021/12/20/wat-is-minting/

Vito Eikholt. (2021, 19 november). NFT minting – Binnen 5 minuten jouw eigen NFT - Simpele uitleg. NFT-blog. From: https://nft-blog.nl/nft-minting/

Ooijen, G. (2022, 29 maart). Hoe maak je een NFT? - complete uitleg voor beginners. AllesOverCrypto. From: https://allesovercrypto.nl/blog/maak-nft-complete-uitleg-beginners-0#Stap+4:+Cre%C3%ABer+de+NFT

Geroni, D. (2022, 5 mei). Flow Blockchain vs. Ethereum. Which Is Better for NFT Development? 101 Blockchains. From: https://101blockchains.com/flow-blockchain-vs-ethereum/

Sharma, T. K. (2020, 10 juni). What is Tezos? Pros and Cons. Blockchain Council. From: https://www.blockchain-council.org/blockchain/what-is-tezos-pros-and-cons/

About the author: Marco Filocamo. (2021, 30 november). Blockchain tech for Fashion and Luxury: a future revolution? Fashion Technology Accelerator. From: https://www.ftaccelerator.it/blog/blockchain-technology-future-fashion/

Roberts, J. J. (2021b, juni 9). 5 Ways Businesses Are Already Using Blockchains. Fortune. From: https://fortune.com/2017/08/22/blockchain-walmart-maersk-banking/

Who Made My Clothes Archives. (z.d.). Fashion Revolution. From: https://www.fashionrevolution.org/tag/who-made-my-clothes/

Fta, T. (2021, 30 november). NFT fever: what's the heat like in fashion? Fashion Technology Accelerator. From: https://www.ftaccelerator.it/blog/nft-fever-whats-the-heat-like-in-fashion/

Mcshane, G. (2022, 13 januari). What Are Ethereum Gas Fees? Coindesk. From: https://www.coindesk.com/learn/what-are-ethereum-gas-fees/

Ethereum. (z.d.). Non-fungible tokens (NFT). Ethereum.Org. From: https://ethereum.org/en/nft/#nft-use-cases

Attachment 2 Research points

What does a NFT look like?

An NFT is a non-fungible token. Unlike the cryptocurrency it can't be exchanged with other NFT. Every NFT is unique and has its own value. It has so much in common with cryptocurrency that it is a non-fungible token. It uses blockchain technology in the background to trace the transactions of NFTS exactly like cryptocurrency. NFT can be easily made with Solidity on the Ethereum network, because it has the mechanism of contracts. The NFT contract should follow one or more of those standards:

· ERC-721 – non-fungible tokens

· ERC-223 – much like ERC-20 but with a feature that ensures the tokens are only sent to compatible addresses. This prevents loss of access to the tokens since they cannot be retrieved from incompatible addresses.

· ERC-827 – allows the approval of fungible token transfers so the tokens can be spent by an on-chain third party.

· ERC-777 – an improvement on ERC-20. Users can send tokens on behalf of different addresses.

· ERC-1155 – a smart contract that allows users to manage Ethereum tokens of many types. It can contain ERC-20 or ERC-721 tokens and it works for all types of assets: fungible and non-fungible.

· ERC-1137 – a token standard designed for recurring payments. It works well for subscriptions requiring payments at certain intervals.

· ERC-998 – a smart contract that allows users to merge several NFTs into one NFT.

· ERC-875 – a smart contract that allows users to transfer several NFTs in a single transaction.

· ERC-865 – a smart contract that allows users to pay for a transaction with tokens instead of gas.

What is NFT minting?

NFT minting is the process of converting a digital file into a crypto collectible on the Ethereum blockchain. The digital file is stored inside the decentralised database. When this process is completed, the NFT will be created. It would be impossible to edit, modify or delete this crypto collectible.

To mint an NFT, there are a few steps needed to take before being able to do that. NFTs are primarily minted through different platforms like OpenSea and Rarible. These NFT marketplaces can be compared with eBay or Marktplaats. It can be used for cryptocurrency to mint a digital file collection into an NFT on these platforms. But first, there needs to be some cryptocurrency. Luckily, cryptocurrency can be bought from several platforms like Coinbase and crypto.com. There needs to be a crypto wallet that is compatible with the currency used on a NFT marketplace. When there is a cryptocurrency and a crypto wallet, the minting of a digital file into an NFT will be possible.

What are the possible platforms for NFT's?

The Nfts can be distributed on several different blockchains. Our group made a list to compare the possible options that can be chosen:

Ethereum

Ethereum is still the most widely used chain when it comes to NFTs. However, for beginners, there is a downside to Ethereum. Ethereum is relatively expensive. Of course, there are also many advantages to Ethereum. Ethereum certainly has the biggest name when it comes to NFTs and there are a lot of potential buyers. If people want to use Ethereum, they can, for example, choose larger marketplaces like OpenSea and Rarible.

Solana

Another popular chain for NFTs is Solana. One advantage of Solana for beginners is that Solana is very cheap. On Solana the pay only is $0.00025 per transaction, whereas on Ethereum this will be between $10 and $150, depending on how busy the network is.

Polygon

Polygon can also be used. Polygon is a layer 2 on Ethereum. However, the costs are much lower with Polygon than with Ethereum, where the pay is on average about $1. An advantage is that it can use the same marketplaces as with Ethereum since it's so close to Ethereum.

Tezos

Tezos utilises the Proof-of-Stake consensus mechanism in which participants provide only the necessary computational resources to keep the network working. This is inexpensive as compared to other blockchains that use PoW and other PoS processes. Tezos also allows any stakeholder to participate in the consensus mechanism and rewards for contributing to the security and stability of the network. The primary pitfall associated with Tezos is a delay in token issuance, causing several users to miss out on their rewards without being informed.

Flow

It is a fast, decentralised blockchain platform that can foster the growth of complete ecosystems of applications, especially NFTs. The most critical factor in favour of Flow blockchain is its multi-node architecture. The new blockchain platform could support the creation of apps with security and composability, which could appeal to billions of users. Flow is also the most eco-friendly blockchain. Flow leverages an environment-friendly design via Proof-of-Stake that consumes thousands of times less energy than Proof-of-Work blockchains like the current Ethereum network

What has already been done with blockchain in retail?

Check quality and authenticity of the product

The link below shares the same concept as in Scenario I. Where the wine can be followed using a QR-code.

- *https://scotchwhisky.com/magazine/latest-news/16251/adelphi-adopts-bitcoin-tech-to-battle-fakes/*
- *https://www.youtube.com/watch?v=f6oen4xEO7M&t=41s*

Provenance.org has the exact same concept as described in Scenario I as well. Here the user can see what the journey of the product looks like.

Blockchain as a distributed ledger, that can help to give reliable details about ownership.

Blockchain allows quicker and efficient identity verification processes, with no third-party support needed. Each user can closely and securely monitor their personal data. Everledger has placed more than 1.6 million diamonds on blockchain. The qualities of the diamonds listed in the system include the colour, carat, and certificate number.

How to keep gas fees cheap?

Gas price is primarily determined by the base fee. The most effective way as such to lower the gas fees is by making transactions at a point when nobody is using the blockchain. The lower the users the lower the base fees become.

Another option is by using a 'layer 2 scaling solution', this is an extension of the Ethereum network but probably exists on other networks too in some form. This tool is primarily aimed at increasing the speed of transactions and the number of transactions that can be processed per second. This solution is done off-chain which is later verified by the network and subsequently recorded on the chain. This method essentially cuts down on the amount of gas needed to complete a transaction as it only interacts with the network when a transaction has been validated which in turn requires less gas for the miners to process a transaction. Layer 2 solutions will ease up congestion on the network which should also reduce the overall base fee for all the users.

Finally there's also the obvious option of setting a maximum price that the user is willing to pay for gas fees. This is done pre transaction and usually helps the user to not think twice about spending more than they want to. Once a transaction has been completed and it's under the limit the network will then refund the remaining gas fees.

To take it a step further one can also attempt to simulate a transaction to estimate the amount of gas one has to pay and then if needed make changes to the recipe to decrease the amount gas one has to pay eventually. This is done with software such as DeFi Saver.

How do NFT transactions work?

It's essentially a coin (an asset) with extra information stored on it, usually a digital receipt to an art piece. For example on the Ethereum blockchain NFTs are represented as assets, one can choose to invest in those assets and/or use them as goods for exchange. This system is supposed to give creators new powers, they get to choose how much they make/sell them for and decide the scarcity of their assets. As a result it should 'maximise' their profits, one is capable of even charging a royalty fee each time an owner sells off an NFT to a new buyer. Just like one could see in the current markets today, artificial scarcity could bump these pieces up to insane prices. As such, the value of NFTs on the blockchain are subjective and are solely determined by supply/demand, much like a stock exchange.

Attachment 2 Scenarios

Blockchain: The system that allows NFTs to be verified through a record of transactions across several linked computers. What makes the blockchain unique is that records cannot be altered, making it a perfect system for traceability and transparency.

Here it is possible for instance to embed an NFT in every physical product it sells.

Moreover, NFTs could also authenticate high-value goods and prevent counterfeits from going off, as they provide a great way to block things like massive counterfeiting attacks. By adding an NFT to every product the company sells it would be possible to track them all perfectly and be in full control of both the way they are sold and who the buyer is.

Scenario I - complex clothing label with the use of a NFT

In store concept

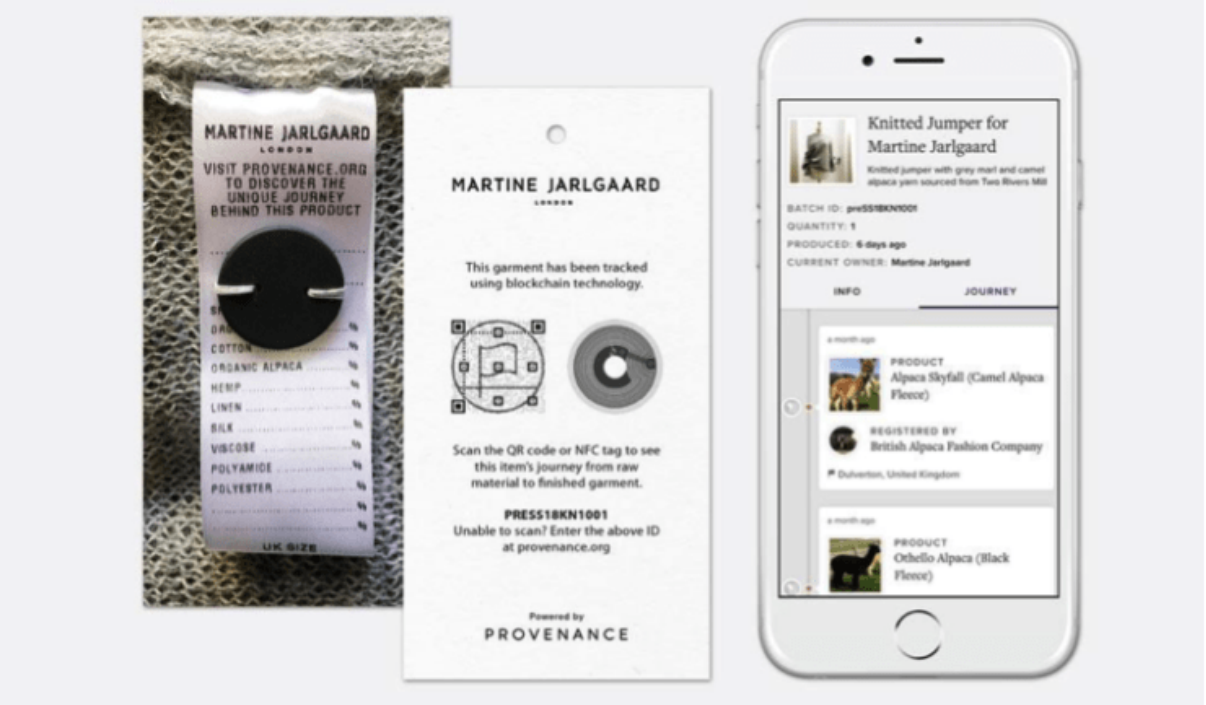
In this scenario the smart contract/ NFT will be used as a clothing label. This NFT will contain all the data from the piece of clothing (manufacturers, designers, etc.). This data is, ofcourse, stored online. To access this NFT, the user will have to scan the QR code which will be printed out on the inside of the clothing piece.

Pains:

- How does the team keep track of the owners of the product? Do they have to create their own block to add it to the blockchain? How does the team keep track of this? Is it necessary to keep track of this?
- How can the team make sure the ID-token doesn't fade after x amount of washes?
- How will reselling work? Who transfers the NFT to the new owner?

Gains:

- User sees the journey of the product
- No waiting time
- Blockchain as a distributed ledger, that can help to give reliable details about ownership.



Provenance. (z.d.). Provenance: Sustainability Communications Software. Van: https://www.provenance.org

Scenario II - creating a NFT after it has been bought online

Online store concept

The user buys a piece of clothing online. Which is linked to a NFT. This NFT will contain all the data from the piece of clothing (manufacturers, designers, etc.). The piece of clothing will be made after it is bought. User receives the piece of clothing.

Gains:

- When buying online, the user creates an account. It is easier to own the NFT in this scenario and link it to a user.

Pains:

- How can the team make the piece of clothing identical to a specific NFT?
- It will take some time to fabricate the piece of clothing.
- How will reselling work? Who transfers the NFT to the new owner?

Attachment 3 User manual

Prerequisites

In order to let the DApp work, the user should install a few prerequisites. These prerequisites are listed below, each with a detailed guide on how to acquire each requirement.

- Node.js - https://nodejs.dev/learn/how-to-install-nodejs
- Truffle - https://trufflesuite.com/docs/truffle/getting-started/installation/
- Metamask - https://metamask.io/download/
- Angular - https://angular.io/cli
- Matic network - https://academy.binance.com/en/articles/how-to-add-polygon-to-metamask
- Matic tokens - https://altcoinreviews.org/guides/how-to-buy-matic-with-metamask/
- Visual studio code - https://code.visualstudio.com/download

Setup tutorial

Listed below is the setup tutorial for the Dapp. These steps are essential to optimise the setup of the product and make sure everything performs as expected.

1. Go to https://github.com/Adj2k22/Amfi.git

2. Clone the Git repository

3. Open the project folder inside Visual studio code

4. Open up the terminal in Visual studio code.

5. After that has opened the terminal in Visual studio code, use the command “cd ./<your project directory>” to head over to the “Frontend” folder inside of the project directory (the folder from the zip).

6. When in the project directory, use the following command: “npm install”.

7. Wait for the installation to finish.

8. Then, search for the “blockChain” directory inside the project folder. The location of the directory is as followed: “./amfi/frontend/src/blockChain/”

9. Open up a second terminal page in Visual studio code by pressing the “+” symbol on the terminal tab

10. Repeat step 8 on this second terminal page

11. When in the “BlockChain” directory. Use the command “cd ./998” on the first terminal, and the command “cd ./nfts” on the other

12. When inside the “998” directory and “nfts” directory at the same time, use the following command: truffle migrate

13. Revert back to the “frontend” directory and use the following command: ng serve

14. Copy the URL from the terminal and paste it into a browser. Press enter and there should appear the DApp on the screen.

Truffle configuration AND Contract deployment

Truffle configuration depends on what network the Dapp is going to be deployed on. The Dapp is configured to automatically detect the address of the contract after deploying through its ABI. If you want to run the app on your own localhost, install Ganache and copy the localhost address and its port to the truffle-config.js by editing the network section in the

“truffle-config.js” file(describing image listed below), paste them into the network JSON object.

truffle-config.js 445 bytes

```
module.exports = {

  networks: {
    development: {
      host: "127.0.0.1",
      port: 7545,            // Standard Ethereum port (default: none)
      network_id: "*",       // Any network (default: none)
    }

  },

  contracts_directory: './src/contracts/',
  contracts_build_directory: './src/abis',

  compilers: {
    solc: {
      version:'^0.8.0',
      optimizer:{
        enabled:'true',
        runs: 200
      }
    }
  },

};
```