Blockchain Engineering

Technical report

Blockchain Engineering

Authors	*TUD*
J.M. van der Boon	4078128
B.I.Y.L. Ho	4320867
B.T.J. van Schaick	4489357
J.I.. Mulder	4492021

April 9, 2021

1 Introduction

This report is an overview of the software product that was developed during the MSc course Blockchain Engineering (CS4160) at TU Delft. This specific topic was guided by Blocklab, a blockchain fieldlab initiated by the port of Rotterdam and the municipality of Rotterdam [2].

Academic guidance was provided by Dr. Zeki Erkin, Miray Ayşen and Tianyu Li. The goal of the project is to gain understanding of blockchain technology, and to design and engineer a blockchain based solution that satisfies the requirements of our client Blocklab.

1.1 Problem introduction

Inventory Financing (IF) refers to the process of acquiring a line of credit by an Importer wishing to buy goods from a Manufacturer. They can't immediately pay the required amount, and thus seeks financing from Financiers, who will pay the required amount for them in advance for a small fee. The bought items are stored at a warehouse, and used as collateral for the loan, until the Importer finds buyers for their goods. This is done so that the cash flow of the Importer remains constant and predictable.

A Logistics Service Provider (LSP) stores inventory on behalf of their clients. While they do not become owner of the goods themselves, they can keep control over the inventory until their client has paid all associated charges with the inventory (right of retention). Furthermore, the LSP possesses valuable information about the goods and their condition.

The right of retention and the information that the LSP possesses, makes them a valuable partner for financiers. Research[4] shows that involving the LSP in the IF process can have a positive effect on all parties involved. However, as of 2021, the IF process still is very much a paper process, relying on documents such ass the bill-of-lading and letter-of-credit. These documents have the tendency to get lost and can be easily forged.

A system needs to be built which improves the efficiency and reliability of this process. The required system can be decomposed into 2 main functionalities, namely: Proof of Existence, and Proof of Ownership. Proof of Existence refers to the connection made to the Warehouse Management System (WMS) of the LSP. When a shipment arrives at an LSP, they enter it into their WMS, which "proves" its existence. This existence then needs to be verifiable by the Supplier and Importer. Proof of Ownership refers to the use of a non-fungible token which is tied to a specific shipment. Ownership of the shipment can then be transferred by transferring ownership of the token. The token should then be usable as collateral to obtain financing from Financiers.

1.2 Blockchain as solution

Blockchain is a data model where transactions are grouped together into a block. Each block is linked to a previous block on the chain since each block stores a hash of the previous block (the merkle root). Inside each block there is a merkle tree which stores transactions in a specific way where every leaf node is a hash of transaction data, and every non leaf node is a hash of its children nodes so from this bottom-up approach you end up with a single hash at the top which is also called the merkle root. It is clear that changing one transaction inside a merkle tree would invalidate the block since each non leaf node in the tree is dependent on its children, so the merkle root will obtain a different hash. This entails that blockchain allows for efficient verification of the integrity of the data, and data cannot be changed and is immutable once it is put on the blockchain.

A core feature of using blockchain is decentralized trust. It can be used in a decentralized way that eliminates the risk of storing all data in a central place or all trust in a single party. And because of the immutability property of the blockchain, it guarantees that information stored on the blockchain keeps its integrity. Decentralization also enables transparency because every person can access the data on the blockchain. By using blockchain technology it is possible to improve the efficiency and reliability of inventory financing by using Proof of Ownership and Proof of Existence. An importer and a financier often don't know each other, and by using blockchain they can reduce the level of required trust needed for trading by using these proofs and realize their business goals more efficiently.

2 Blockchain Solution

In this section, the relevance of the proofs is explained, followed by the technological considerations and technological decision. The technological considerations explains the frameworks that we have considered to use for the product in order to solve the problem. The technological decision explains our reasoning for choosing a specific framework for the product.

2.1 Proof of Existence

The Proof of Existence assures financiers that the shipment exists. The Proof of Existence in our solution refers to a shipment that arrives at a Logistics Service Provider, and that information is placed in the warehouse management system. This information, e.g. a Bill of Lading can be hashed, in conjunction with the public key of an Importer. A zero knowledge proof can then be generated, verifying that the LSP knows the preimage to the hash, without revealing the details of it publicly. The resulting hash and proof can then be pushed to an on-chain merkle tree, by which the LSP commits to the existence of the shipment and owner.

2.2 Proof of Ownership

A financier accepts an asset as security to secure a loan. And when an importer defaults on this loan, the financier can seize the shipment that was used as collateral and sell it to recover from a loss. A proof of Ownership can be realized by creating a a non fungible token on the blockchain which is linked to a specific shipment. The ownership of this shipment can be transferred between actors, and the token can be used as collateral to obtain financing from financiers.

2.3 Technological considerations

For LSPs to participate in IF a connection needs to be made to the WMS of the LSP. This WMS tracks the in- and outflows of the warehouse and thus knows when certain shipments arrive from Suppliers. An Importer buying goods from a Supplier does not know/trust this entity, and thus waits for arrival and confirmation of goods by the LSP, only then the payment is made with the LSP as the trusted third party.

The integration and standardization of the workflow between these various entities will be done by using a public distributed ledger, as it provides a common frame of reference, minimizing the required level of trust necessary. The main requirement is that proof of existence and ownership can be provided through tokenization of inventory. We have looked at both the Baseline Protocol [6] in conjunction with TradeTrust [8], as well as Hyperledger Fabric [1] as possible solutions for this problem.

2.3.1 Hyperleder Fabric

Hyperledger Fabric provides a modular and open-source system with which permissioned blockchains can be deployed, aimed at integrating enterprise use cases. It is permissioned as each Fabric blockchain requires all participating nodes to have a verified identity, which are provided by Membership Service Providers (MSPs) [1]. This means that all nodes in a network know each other, and have all agreed to work together in a certain context. This Fabric is then private and only participants in it can see the recorded transactions in it.

2.3.2 Baseline Protocol

The Baseline Protocol aims to provide a framework which enables confidential and complex collaboration between businesses, using a public ledger as a common frame of reference, without moving the sensitive data out of the traditional systems of record in use by each respective business. For now, the consensus in the community is that it should be built upon Ethereum [9], which is a permissionless blockchain. zk-SNARKs are utilized [7] on top of it so that workflows can be private and enforced according to pre-determined business logic. The benefit of using Ethereum is that it requires no trusted third party, and participation is open.

2.3.3 TradeTrust

TradeTrust is another framework built upon Ethereum which allows for the digital verification of official documents. It also allows for the creation of Non Fungible Tokens (NFTs), which are linked to specific documents, so that ownership of this document can be moved by transferring the token using digital signatures.

2.4 Technological decision

Hyperleder Fabric was rejected mainly due to the fact that Hyperledger Fabric is permissioned, which generates organizational overhead in terms of setting up the Fabrics for specific sets of Suppliers, Importers and LSPs. Furthermore, there is no native currency available in order to buy and sell tokenized goods, so that would have to be developed in addition.

A permissionless blockchain solution was preferred over a permissioned one so that any business party can easily participate and collaborate with each other. Also, Baseline uses the Ethereum network so we can utilize the native currency of ethers for token trading. Besides that, Ethereum provides clear advantages over Fabric in terms of tokenization as it provides the ERC-721 standard for NFTs.

TradeTrust was chosen because we were looking for a solution to solve the problem of tokenization and transferring ownership between parties. TradeTrust can do this by using the Token Registry and Title Escrow contracts which are deployed on the Ethereum blockchain. Given these considerations, we chose to utilize Baseline and TradeTrust to build the required system.

3 Our solution

In this section the product solution is explained. First the development process is discussed, followed by the software development platform and tooling that was used for the product. Next, the scope and requirements are shown that has approval from our client Blocklab. And finally the most important components for our solution are described, including the cyber security aspect of the product.

3.1 Development process

The solution was build over a period of 7 weeks; from 15th of February untill the 9th of April 2021. Weekly update meetings were held with Aljosja Beije and Hamza Suwae of Blocklab. Furthermore an internal meeting was planned every Thursday which resulted in a weekly email to all stakeholders regarding the current state of the project. Work was divided among the group members per block of functionality on a weekly basis, as circumstances made it impossible to meet physically. This was deemed the most efficient use of the time available.

After understanding the supply chain financing process flow as mentioned by Blocklab, we created a design document which outlined the scope of the project and the requirements for the Minimal Viable Product, see the appendix 5. After approval, development was started quickly as preparations were already underway.

3.2 Software development platform and tooling

GitHub[1] was chosen as a project management platform, as it allows for fast cooperation through code sharing and issue tracking. Several repositories were created on this over the duration of the course. Earlier repositories were used to test reference baseline implementation and get acquainted with the protocol, while our final repository is used to develop our own implementation of the protocol.

Due to all project members having different development environments, causing problems while integrating software, a CI server was introduced early on in the development. CircleCI[2] allowed us to run our builds and tests against a clean environment and was the CI of choice due to experience with it in the past.

JavaScript, provided us with numerous useful libraries with off the shelf client implementations for interactions with existing blockchain, distributed messaging and database solutions. TypeScript's type annotations proved their worth when implementing interfaces exposed by the work of other team members, while Yarn was used to manage package dependencies.

A special mention for the Truffle Suite[3], a set of tools to simplify blockchain/ethereum development. Truffle is used throughout the project for smart contract creation and deployment. Ganache, the private blockchain included in the suite, improved feedback loop in the blockchain development cycle by a landslide. allowed us to test and execute our blockchain transactions and contract implementations in a controlled environment.

3.3 Scope and requirements

Actors

A list of the parties that are involved in the process of international trading.

- **LSP**
  Owns the platform which is used to interact with the system.
- **Importer**
  Uses the platform to liquidize the goods bought from the manufacturer as long as they are in storage at the LSP.
- **Financiers**
  Provide liquidity to the importer with the goods as collateral.

[1] https://github.com/

[2] https://circleci.com/

[3] https://www.trufflesuite.com/

- **Manufacturer**
  Produces the goods.
- **Buyer**
  Buys the goods at the end of the process.

With regards to this platform and the product we will be building, only the first three actors are users of the platform. For each of these use cases will be created to identify the features required in the platform.

Requirements

The requirements for the final product are divided in two categories. Namely the functional requirements and the non-functional requirements. The functional requirements are defined by defining use cases for the actors interacting with the system. The non-functional requirements are regarding the environment of the system and aspects that should be taken into consideration while designing the system. This ensures that the final product will be usable.

Functional requirements

User	Use case	Functionality	Information
LSP	Coupling with WMS	Ingest	Automatic coupling with WMS of LSP. When goods arrive at the warehouse automatically tokens are generated and provided to the importer as the new owner.
	List of goods in warehouse	User interface	Overview of items in the warehouse that have been tokenized.
	Overview of release requests	User interface	Overview of the release requests that currently are pending for approval.
	Approve release requests	Release	Approve the release of goods after the correct tokens have been provided.
Importer	Overview of items	User interface	Overview of goods that have been bought and are currently stored at LSP. Shows the status of the goods as currently tokenized or not.
	Request for release	Release	Use of ownership tokens to request the release of goods from the LSP.
	Tokenize goods	Ingest	Request the LSP to tokenize a batch of goods and provide the importer with the ownership tokens.
	Wallet for tokens	Ownership	Place to store the tokens
	Buy tokens	Ownership	Buy back tokens from financiers to be able to retrieve the goods from the LSP and send the goods to a buyer.
	Sell tokens	Ownership	Liquidize goods by requesting the LSP to tokenize them.
Financiers	Overview of financing deals	User interface	Overview of the deals the financier is or has been involved in.
	List of tokens	User interface	Overview of the ownership tokens linked to the goods. Proof of colleteral.
	Wallet for tokens	Ownership	Place to store the tokens
	Buy tokens	Ownership	Liquidize goods by buying tokens from the importer.
	Sell tokens	Ownership	Sell tokens to importer so the goods can be released.

Non-functional requirements

#	Name	Description	Applicable for MVP
1	*Security*	As this system will be used for international trade security is an important part on multiple fronts. Foremost the amount of capital that is controlled with this system is substantial, attacks from organized crime are to be expected. Confidentiality of the data is also important due to business competition.	Yes
2	*Scalability*	The system has to be able to handle the volume of the international trade. Delays are costly within logistics and will prevent adoption of this system.	No
3	*Regulations*	In the international context of the trade sector regulations change often. Any implementation of legal requirements will need to be implemented in such a way that it does not create dependencies, therefore allowing them to be changed rapidly. Another important matter to address is the issue of legal ownership. As of now regulation does not exist that recognizes tokens as a legal indicator of ownership.	No
4	*Auditability*	As customs officials are likely interested in the data of this platform, an appropriate way of sharing the data has to be devised.	No
5	*Maintainability*	In the same regard as scalability, the system has a need for being maintainable. This reduces downtime and allows adoption of the system. By making the system maintainable it will be possible to change aspects without creating issues. This is also important regarding the security demands, if a cryptographic method proves to be less secure it needs to be changed.	No

The decision on it being included for the minimal viable product was made with regards to time costs and estimated projected effort. A use case diagram and a process flow diagram were also created, both can be found in the design document in the appendix, see 5.

3.4 Solution components

3.4.1 Baseline

The Baseline Protocol aims to provide a specification/design pattern for implementing secure private business to business workflows, using a public blockchain. It is still in development, although some reference implementations and examples exist [5]. In our opinion, it can best be seen as a form of middleware for businesses on which to transact privately, while being public and open. This decentralizes the burden of coordination, as it is no longer necessary to assign a centralized entity to do this.

The current protocol specification and reference implementations suggest the following division of components:
Organization Manager This component is responsible for managing organizations and workgroups. This is done by utilizing an on chain registry, where organizations and workgroups can be added. An organization consists of the following properties:

- **Organization name**
- **Address**: in our case it refers to an Ethereum Address.
- **Messenger URL**: public endpoint for secure point to point messaging.
- **Messenger public key**: public key used to authenticate messages.
- **Zero knowledge public key**: a specific public key to use for verifying identities in zero knowledge proofs.

A workgroup consists of the following properties:

- **Workgroup name**
- **Shield Address**: On chain merkle tree contract, linked to verifier.
- **Verifier Address**: Verifier contract address.

The organization manager is capable of tracking any on-chain registry. The owner of the registry can add organizations and workgroups. It is also responsible for deploying the respective shield contract for each workgroup.
ZKP Manager The Zero Knowledge Proof (ZKP) Manager manages local zero knowledge proof circuits. It is capable of compiling them, deploying them as a smart contract to the blockchain, as well as generating proofs for them. The verifying keys are automatically baked into the smart contract, so these do not have to be exchanged. The manager keeps track of each compiled circuit and deployed verifier.
Commit Manager The commit manager is responsible for tracking the on-chain shield contract, containing a merkle tree. This on-chain tree is sparse, as to save on transaction fees. Thus a full version of the tree is stored locally, while verifying it is still in the same state as the on chain tree. This can be done by only storing the frontier of nodes necessary to calculate the merkle root of the tree on-chain. The commit manager also provides functionality to verify that certain commitments are indeed in the tree.
Blockchain Manager The blockchain manager is responsible for all blockchain related interactions. It can deploy smart contracts, get their interfaces, check if they are still deployed, as well as compile them during runtime. It also provides a HD Wallet instance, which can be used to generate ethereum accounts using a single private seed of 24 words.

Specifically for our use case, this implementation of the baseline protocol allowed us to do the following:

- Maintain an on-chain registry with Importers and Financiers, as well as their respective workgroups.
- Allow an LSP to entangle local WMS records with Importers, by hashing the bill and their public key as a commitment to a merkle tree. This commitment can then be used as input to mint the NFTs as discussed in 3.4.2.
- Allow Importers and Financiers to track and verify these commitments, thereby keeping them up to date, while keeping the confidential information private.
- Removes the burden of coordination from the LSP, as they only need to invite people to the workgroup, from then on it is up to them to set up their local baseline stack accordingly.

3.4.2 Tokens

Non fungible tokens (NFT) or unique tokens (ERC721) are deployed on the Ethereum blockchain, by using the TradeTrust[4] framework. The framework requires the existence of two contracts on the chain chain, the Token Registry and a Title Escrow contract. The Token Registry is a smart contract that keeps track of the current owner of a NFT, while the Title Escrow contract stores information about a joint ownership, a holder and a beneficiary of a token. When a LSP verifies the arrival of goods in the warehouse, the Title Escrow is immediately deployed by the LSP and a token is constructed from the hash of an Electronic Bill of Lading and minted to a Title Escrow contract. The holder and the beneficiary of the contract is the importer and is set by the LSP.

Next, the LSP looks for financiers and agrees on a price to transfer and sell the token from importer to financier, and a price for an importer to buy this token back from the financier. The LSP will set this deal, including the public address of the financier into the Title Escrow for the financier so that the financier can buy this token. Once this deal is set on the contract, the prices and the public address that will become the new holder becomes fixed and cannot be changed anymore.

When the financier transfers the correct amount of ethers to the Title Escrow contract, the financier automatically becomes the new holder of the title escrow, and the importer will receive this payment from the Title Escrow. At a later point in time, the importer buys this token back from the financier, by sending the correct buy back price to the Title Escrow in order to become the holder again.

When the importer becomes the holder of the token, the importer can request a release of the goods from the LSP by issuing a release. The token will be transferred from the Title Escrow to the Token Registry contract. The LSP can use function calls on the Token Registry to verify that the token has returned and can proceed with releasing the goods to the importer and burn the token.

[4] `https://www.tradetrust.io/`

3.4.3 API and front-end

The front-end is developed using the React framework. It interacts with the back-end API to provide an interface of the users to the system. Initially MetaMask integration was planned for the project, this had to be abandoned due to unforeseen time constraints.

An overview of the entire deployed stack can be seen in Figure 1. It consists of the baseline stack, which exposes the components discussed in 3.4.1 as gRPC services. The backend API consists of a REST API using Express, which is used by the front end.

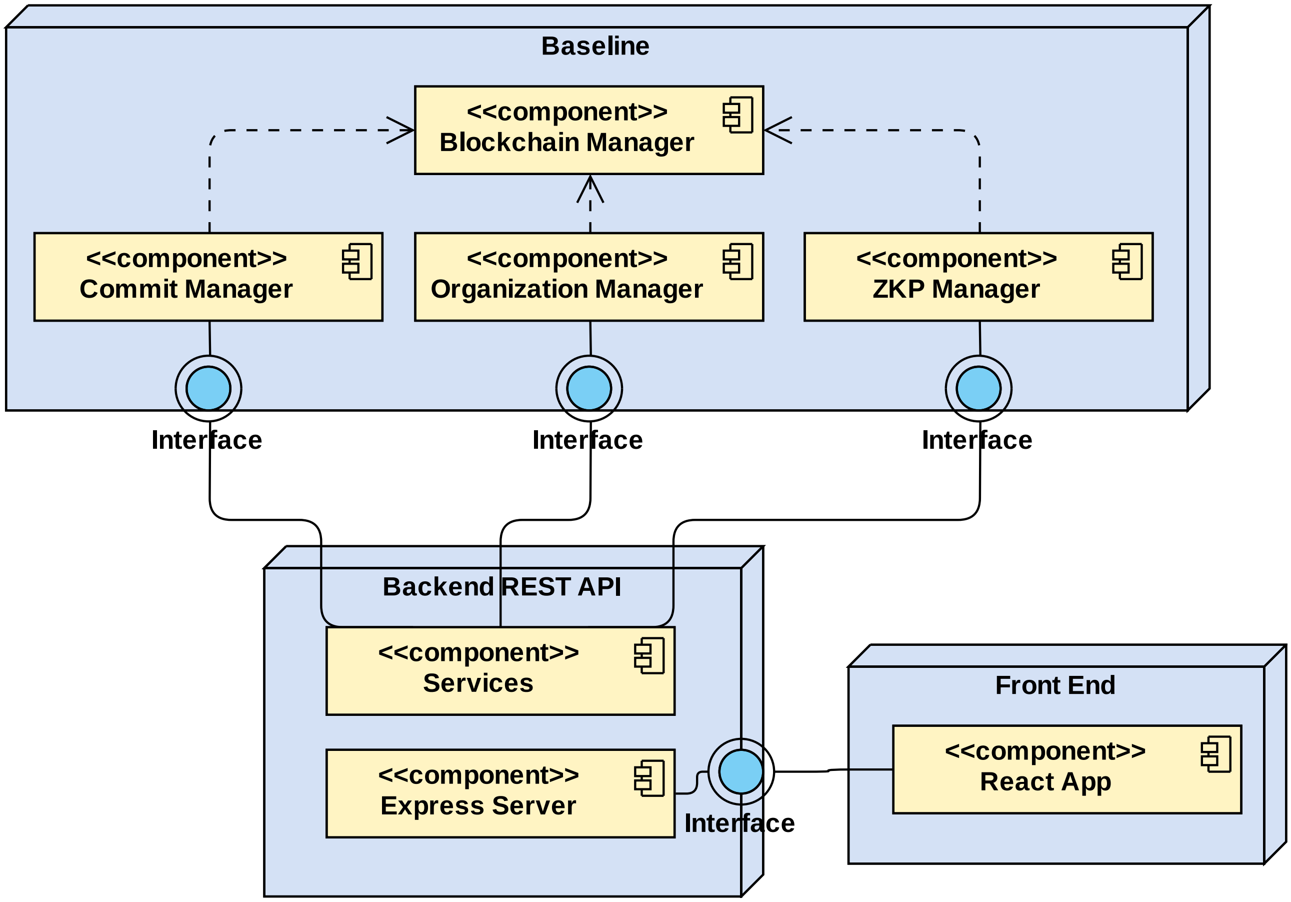

Figure 1: UML Deployment Diagram of stack.

3.5 Cyber security

The shipments involved in IF transfer economic value, and as such should be treated accordingly. It is estimated that 40% of all maritime fraud involves documentary fraud [3]. This documentary fraud includes the forgery of bill of ladings, to claim delivery ahead of the rightful owner. For our use case, this specifically means that the generated tokens should be handled with care. More specifically, it should not be possible for any malicious party to tokenize a bill of lading of someone else. When a party requests shipment release to the LSP, they have to be sure that they are the rightful owner.

This is guaranteed in our solution, as the LSP controls the token registry which issues the tokens. This deployed smart contract can only be operated by its owner, which is an ethereum address. As long as the LSP manages their private key for this address with care, no token can be minted by any malicious party.

4 Limitation and future work

The current product is a proof of concept and has limitations. There is no account management system like MetaMask integrated into the product that allows users to manage their wallets and keys. The second limitation is that only one token for each shipment is made. This could be extended into supporting multiple tokens per shipment, and even products within the shipment. And by having more tokens, it is possible to obtain financing from multiple financiers. The third limitation is that there is no negotiation mechanism in place between the importer and the financier regarding the price for a token, this would be a nice addition for future work.

Due to their complexity, the zero knowledge proofs are not used to their full extent. For now, they are only used to entangle a bill of lading with a importer in the form of a commitment. Nothing more is done after this. It would be possible to use this commitment as input into further proofs, so that the specific Importers could spend these commitments. This allows for a progression of state, making the entire process actually look like a workflow with worksteps.

5 Conclusion

We have created a product that satisfies the requirement that were given by the client by using blockchain technology for the supply chain finance environment. The frameworks that we used for the product are Baseline, TradeTrust and the React framework.

Each shipment that arrives at a LSP is entered into the WMS and a NFT token (ERC-721) is automatically created. This token is linked to the WMS record by making use of on-chain commitments using zero knowledge proofs. This commitment allows for the verification of existence, and the token for proof of ownership. The actors involved can use these proofs to verify that the goods exists and ownership of the goods can be proven by being the owner of the token. This token can then be used as collateral to obtain financing by trading this token for an agreed price. The goods can be released by the importer once the importer is the owner of the token. The goods can then be sold to a customer.

As for the cyber security aspect of the product, as long as the private key of the LSP is not in hands of malicious parties, no token can be created. So ownership of the goods cannot be claimed by adversaries.

References

[1] Elli Androulaki et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains". In: *Proceedings of the thirteenth EuroSys conference*. 2018, pp. 1–15.

[2] BlockLab. *About Us*. Blocklab. Sept. 11, 2017. URL: `https://www.blocklab.nl/about/` (visited on 04/09/2021).

[3] cognizant. *Blockchain for Trade Finance: Trade Asset Tokenization (Part 3)*. Feb. 22, 2018. URL: `https://www.cognizant.com/whitepapers/blockchain-for-trade-finance-trade-asset-tokenization-part-3-codex3337.pdf`.

[4] Erik Hofmann. "Inventory financing in supply chains: A logistics service provider-approach". In: *International Journal of Physical Distribution & Logistics Management* (2009).

[5] Oasis. *Baseline Github Repository*. Jan. 1, 2020. URL: `https://github.com/ethereum-oasis/baseline` (visited on 04/09/2021).

[6] Oasis. *The Baseline Protocol an Oasis Open Project*. Dec. 2020. URL: `https://www.baseline-protocol.org/`.

[7] Alexandre Miranda Pinto. "An Introduction to the Use of zk-SNARKs in Blockchains". In: *Mathematical Research for Blockchain Economy*. Springer, 2020, pp. 233–249.

[8] TradeTrust. *An easy way to check and verify your documents*. 2020. URL: `https://www.tradetrust.io/`.

[9] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

Appendices

Approved design document

This document was created during the conversations and discussions with Blocklab. In the first weeks this was a living document that was changed often to reflect the scope and goal of the project to ensure all stakeholders were in agreement of the endgoal.

Delft University of Technology

Blockchain Engineering

CS4160

Tokenized IF: Design Document

April 9, 2021

TU Delft Delft University of Technology

Introduction

Overview

Inventory Financing (IF) refers to the process of acquiring a line of credit by an Importer wishing to buy goods from a Manufacturer. They can't immediately pay the required amount, and thus seek financing from Financiers, who will pay the required amount for them in advance for a small fee. The bought items are stored at a warehouse, and used as collateral for the loan, until the Importer finds buyers for their goods. This is done so that the cash flow of the Importer remains constant and predictable.

A system needs to be built which improves the efficiency and reliability of this process by utilising blockchain technology. The required system can be decomposed into 2 main functionalities, namely: Proof of Existence, and Proof of Ownership. Proof of Existence refers to the connection made to the Warehouse Management System (WMS) at the Logistics Service Provider (LSP). When a shipment arrives at an LSP, they enter it into their WMS, which "proves" its existence. This existence then needs to be verifiable by the Supplier and Importer. Proof of Ownership refers to the use of a non-fungible token which is tied to a specific shipment. Ownership of the shipment can then be transferred by transferring ownership of the token. The token should then be usable as collateral to obtain financing from Financiers.

Definitions

- **IF**: Inventory Financing
- **WMS**: Warehouse Management System
- **MVP**: Minimal Viable Product

Assumptions

In order to properly demarcate the scope of the required system, assumptions need to be made about various entities involved with the system. The following assumptions are made for this product.

- Users know how to send/receive Ether on the Ethereum network.
- Users know how to use Metamask, a wallet capable of interacting with the Ethereum network as a browser plugin.
- Suppliers are responsible for the generation of the token.
- No disputes will arise when trying to buy back the tokens from the financiers.

Deadlines

Product delivery:	29th of April.
Presentation:	Week after delivery. Week 18.

Scoping

Actors

A list of the parties that are involved in the process of international trading. These are the same actors as can be found in the process diagram.

- **LSP**
  Owns the platform which is used to interact with the system.
- **Importer**
  Uses the platform to liquidize the goods bought from the manufacturer as long as they are in storage at the LSP.
- **Financiers**
  Provide liquidity to the importer with the goods as collateral.
- **Manufacturer**
  Produces the goods.
- **Buyer**
  Buys the goods at the end of the process.

With regards to this platform and the product we will be building, only the first three actors are users of the platform. For each of these use cases will be created to identify the features required in the platform.

Requirements

The requirements for the final product are divided in two categories. Namely the functional requirements and the non-functional requirements. The functional requirements are defined by defining use cases for the actors interacting with the system. These can be found later in this document. The non-functional requirements are regarding the environment of the system and aspects that should be taken into consideration while designing the system. This ensures that the final product will be usable.

Non-functional requirements

Use cases

We have created use cases that are applicable for the minimal viable product, and from that we have created the use case diagram which you can see below in figure 1. This list of use cases can expand if time and resources permit.

We have also created a table that contains more details about the use cases in table 2 . Other use cases can be added later on if time permits.

#	Name	Description	Applicable for MVP
1	*Security*	As this system will be used for international trade security is an important part on multiple fronts. Foremost the amount of capital that is controlled with this system is substantial, attacks from organized crime are to be expected. Confidentiality of the data is also important due to business competition.	Yes
2	*Scalability*	The system has to be able to handle the volume of the international trade. Delays are costly within logistics and will prevent adoption of this system.	No
3	*Regulations*	In the international context of the trade sector regulations change often. Any implementation of legal requirements will need to be implemented in such a way that it does not create dependencies, therefore allowing them to be changed rapidly. Another important matter to address is the issue of legal ownership. As of now regulation does not exist that recognizes tokens as a legal indicator of ownership.	No
4	*Auditability*	As customs officials are likely interested in the data of this platform, an appropriate way of sharing the data has to be devised.	No
5	*Maintainability*	In the same regard as scalability, the system has a need for being maintainable. This reduces downtime and allows adoption of the system. By making the system maintainable it will be possible to change aspects without creating issues. This is also important regarding the security demands, if a cryptographic method proves to be less secure it needs to be changed.	No

Table 1: Non-functional requirements

User	Use case	Functionality	Information
LSP	Coupling with WMS	Ingest	Automatic coupling with WMS of LSP. When goods arrive at the warehouse automatically tokens are generated and provided to the importer as the new owner.
	List of goods in warehouse	User interface	Overview of items in the warehouse that have been tokenized.
	Overview of release requests	User interface	Overview of the release requests that currently are pending for approval.
	Approve release requests	Release	Approve the release of goods after the correct tokens have been provided.
Importer	Overview of items	User interface	Overview of goods that have been bought and are currently stored at LSP. Shows the status of the goods as currently tokenized or not.
	Request for release	Release	Use of ownership tokens to request the release of goods from the LSP.
	Tokenize goods	Ingest	Request the LSP to tokenize a batch of goods and provide the importer with the ownership tokens.
	Wallet for tokens	Ownership	Place to store the tokens
	Buy tokens	Ownership	Buy back tokens from financiers to be able to retrieve the goods from the LSP and send the goods to a buyer.
	Sell tokens	Ownership	Liquidize goods by requesting the LSP to tokenize them.
Financiers	Overview of financing deals	User interface	Overview of the deals the financier is or has been involved in.
	List of tokens	User interface	Overview of the ownership tokens linked to the goods. Proof of colleteral.
	Wallet for tokens	Ownership	Place to store the tokens
	Buy tokens	Ownership	Liquidize goods by buying tokens from the importer.
	Sell tokens	Ownership	Sell tokens to importer so the goods can be released.

Table 2: List of use cases required for the minimal viable product

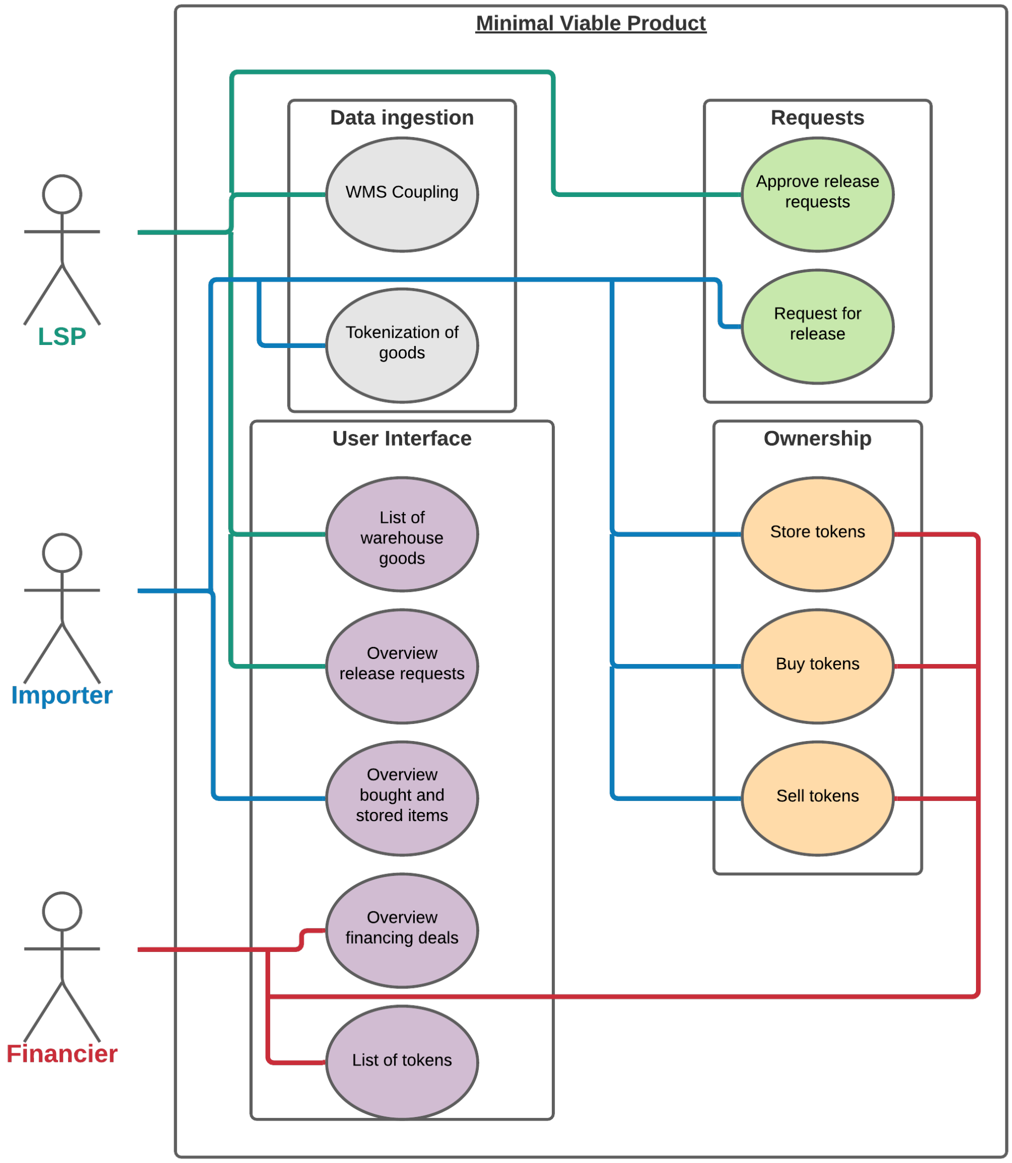

Figure 1: Use case diagram of the minimal viable product

Process flow diagram

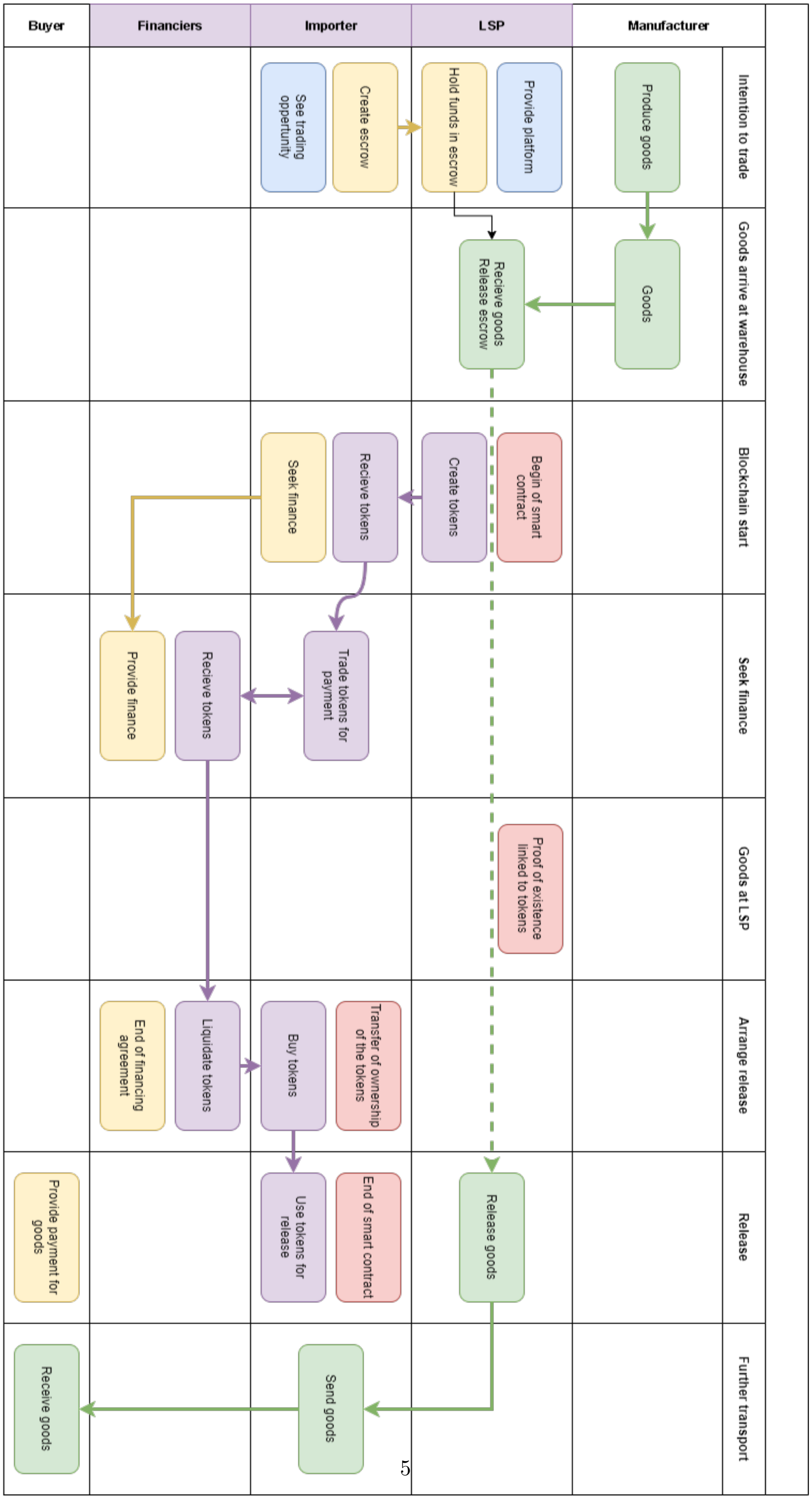

Figure 2: Flow diagram of the process.

Technological Considerations

For LSPs to participate in IF a connection needs to be made to the WMS of the LSP. This WMS tracks the in- and outflows of the warehouse and thus knows when certain shipments arrive from Suppliers. An Importer buying goods from a Supplier does not know/trust this entity, and thus waits for arrival and confirmation of goods by the LSP, only then the payment is made with the LSP as the trusted third party.

The integration and standardization of the workflow between these various entities will be done by using a public distributed ledger, as it provides a common frame of reference, minimizing the required level of trust necessary. The main requirement is that proof of existence and ownership can be provided through tokenization of inventory. We have looked at both the Baseline Protocol Oasis (2020) in conjunction with TradeTrust TradeTrust (2020), as well as Hyperledger Fabric Androulaki et al. (2018) as possible solutions for this problem.

Hyperledger Fabric: Hyperledger Fabric provides a modular and open-source system with which permissioned blockchains can be deployed, aimed at integrating enterprise use cases. It is permissioned as each Fabric blockchain requires all participating nodes to have a verified identity, which are provided by Membership Service Providers (MSPs) Androulaki et al. (2018). This means that all nodes in a network know each other, and have all agreed to work together in a certain context. This Fabric is then private and only participants in it can see the recorded transactions in it.

Baseline Protocol: The Baseline Protocol aims to provide a framework which enables confidential and complex collaboration between businesses, using a public ledger as a common frame of reference, without moving the sensitive data out of the traditional systems of record in use by each respective business. For now, the consensus in the community is that it should be built upon Ethereum Wood et al. (2014), which is a permissionless blockchain. zk-SNARKs are utilized Pinto (2020) on top of it so that workflows can be private and enforced according to pre-determined business logic. The benefit of using Ethereum is that it requires no trusted third party, and participation is open.

TradeTrust: TradeTrust is another framework built upon Ethereum which allows for the digital verification of official documents. It also allows for the creation of Non Fungible Tokens (NFTs), which are linked to specific documents, so that ownership of this document can be moved by transferring the token using digital signatures.

Given these considerations, we chose to utilize Baseline and TradeTrust to build the required system. Mainly due to the fact that Hyperledger Fabric is permissioned, which generates organizational overhead in terms of setting up the Fabrics for specific sets of Suppliers, Importers and LSPs. Besides that, Ethereum provides clear advantages over Fabric in terms of tokenization as it provides the ERC-721 standard for NFTs.

Architectural Overview

Baseline Protocol

The Baseline Protocol provides a reference implementation adhering to the Baseline Protocol specification. This implementation leverages the Provide stack of which an overview is given in Figure 3. It would make sense to use this reference implementation as a starting point from which to develop the required system. Any entity which participates in the Baseline Workflow will need to run at least this stack locally. Of course depending on the entity more functionality needs to be built around it by us.

TradeTrust

TradeTrust is built using the OpenAttestation framework, for which SDKs are available written in Javascript. An overview of these components is given in the TradeTrust Documentation TradeTrust (2021). Using these components all required actions necessary (issuing transferable records, verifying issuance, transferring ownership) will be possible. An overview is given in Figure 4.

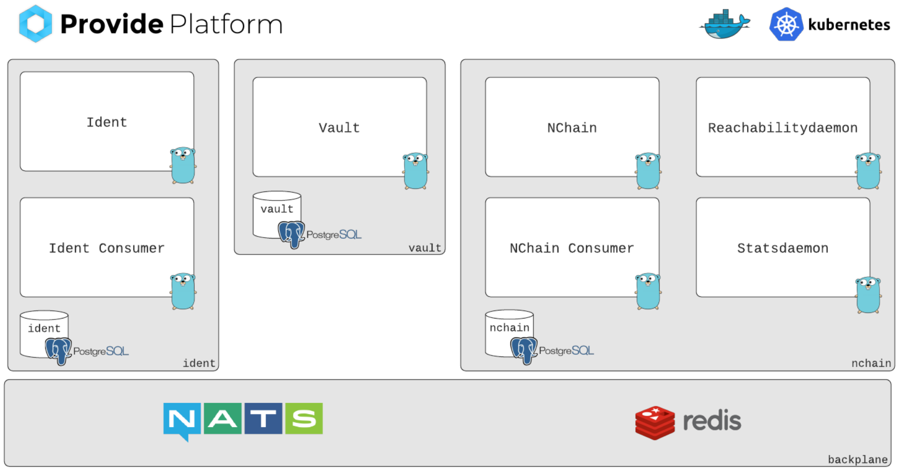

Figure 3: Overview of the Provide stack.

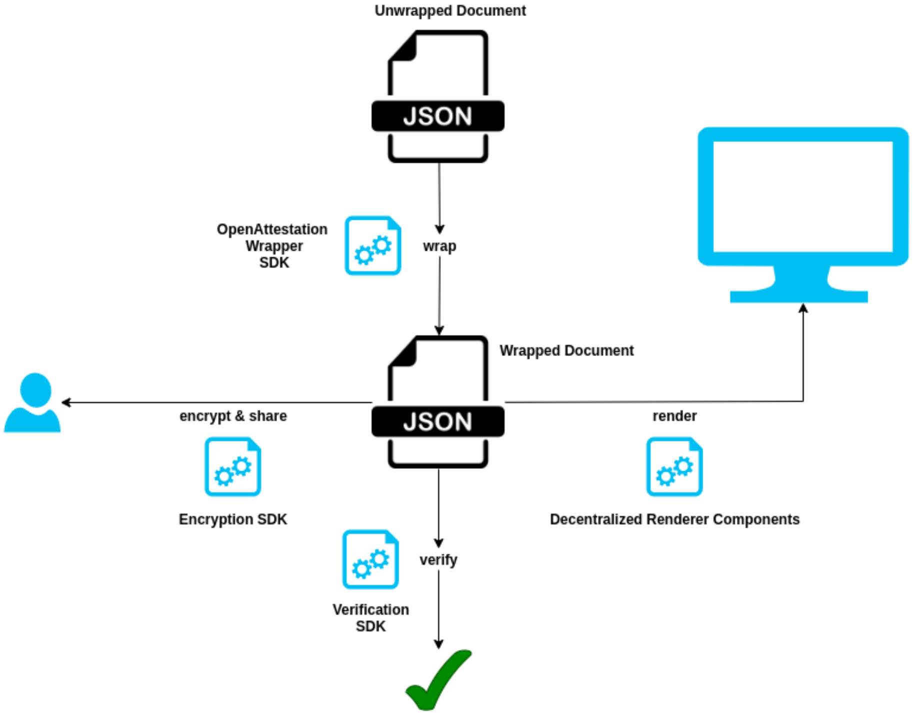

Figure 4: Overview of TradeTrust SDK components.

Process flow in Baseline and TradeTrust

Figure 5 depicts a global view on the required actions that will be necessary in Baseline and TradeTrust to achieve our MVP.

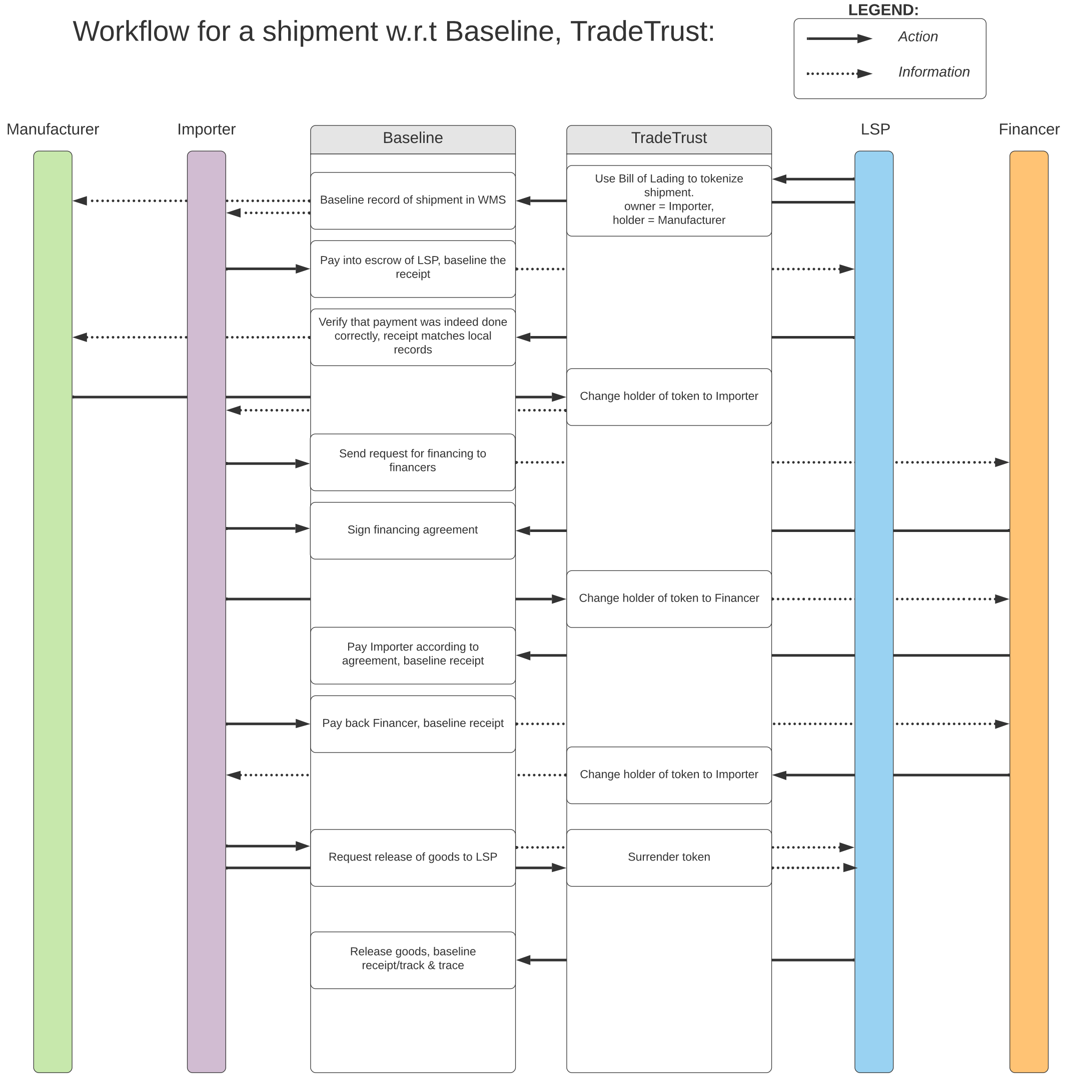
Workflow for a shipment w.r.t Baseline, TradeTrust:
LEGEND:
Action
Information
Manufacturer
Importer
Baseline
TradeTrust
LSP
Financer
Use Bill of Lading to tokenize shipment. owner = Importer, holder = Manufacturer
Baseline record of shipment in WMS
Pay into escrow of LSP, baseline the receipt
Verify that payment was indeed done correctly, receipt matches local records
Change holder of token to Importer
Send request for financing to financers
Sign financing agreement
Change holder of token to Financer
Pay Importer according to agreement, baseline receipt
Pay back Financer, baseline receipt
Change holder of token to Importer
Request release of goods to LSP
Surrender token
Release goods, baseline receipt/track &amp; trace

Figure 5: Process flow for Baseline and TradeTrust for particular shipment.

References

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y. et al. (2018), Hyperledger fabric: a distributed operating system for permissioned blockchains, *in* 'Proceedings of the thirteenth EuroSys conference', pp. 1–15.

Oasis (2020), 'The baseline protocol an oasis open project'.
URL: *https://www.baseline-protocol.org/*

Pinto, A. M. (2020), An introduction to the use of zk-snarks in blockchains, *in* 'Mathematical Research for Blockchain Economy', Springer, pp. 233–249.

TradeTrust (2020), 'An easy way to check and verify your documents'.
URL: *https://www.tradetrust.io/*

TradeTrust (2021), 'Components overview: Tradetrust developer hub'.
URL: *https://docs.tradetrust.io/docs/component/overview*

Wood, G. et al. (2014), 'Ethereum: A secure decentralised generalised transaction ledger', *Ethereum project yellow paper* **151**(2014), 1–32.