The Design and Implementation of a Blockchain-Based Logistics Platform for International Trade

Yen-Yu Chen†, Hsu-Chao Lai†, Jiun-Long Huang†, and Ming-Jiu Hwang‡
†*Department of Computer Science*,
National Yang Ming Chiao Tung University, Hsinchu, Taiwan
‡*Department of Transportation and Logistics Management*,
National Yang Ming Chiao Tung University, Hsinchu, Taiwan
E-mail: timmychen1996@gmail.com, hsuchao.cs05g@nctu.edu.tw,
jlhuang@cs.nctu.edu.tw, mjhwang@mail.nctu.edu.tw

***Abstract*—The document transfer and goods delivery process in the traditional international trade process require a lot of manpower and is inefficient. Also, the accidents like damage of goods and tampering of documents may cause the difficulties in attribution of liability. In recent years, there have been related studies analyzing the characteristics of blockchain and further exploring its applications in the logistics industry. But its applications in international trade are seldom discussed. Thus, we develop a blockchain-based logistic platform for international trade on top of blockchain to simplify the process, thereby showing the potential application of the blockchain technology in the logistics industry.**

***Index Terms*—International Trade, Smart Contract, Blockchain**

I. Introduction

With the advance of the information technology, business operations have been transformed from manual delivery operations into information services. As more and more services are built online through the information technology, the adoption of digitalization happens in the supply chain as well. Due to the unique features such as decentralization, traceability and immutability, Blockchain has been used in different fields such as agriculture and pharmaceutical supply chains. In [1], Xie et al. designed a secured data storage scheme for agricultural product tracking. Through the unchangeable features of the data stored in blockchain, the agricultural products can be tracked, which ensures better food safety. In [2], Hasan et al. proposed the system for managing the logistic products from the distribution side to the receiver side. Since the product is sensitive to temperature, the system utilizes the smart contract for event notification to the sender and receiver once the temperature is out of specifications. Also, its decentralized feature provides trust among different roles and it is well suited to be applied in the logistic field as each participant can serve as one node of the blockchain. [3] mentioned that 10% of Bill of Ladings (B/Ls) have the incorrect data that may cause the shipment disputes and the validation of transaction. In order to address the above issue, blockchain can play an important role through the creation of smart contract. On the basis of the above successful applications, in this paper, we design and implement a prototype platform to speedup international trade process by utilizing the blockchain technology.

The reminder of this paper is organized as follows. In Section II, we give an overview of the proposed platform and the developed smart contracts. The operation costs of these smart contracts are given in Section III. Finally, we draw a conclusion and our future work in Section IV.

II. Proposed Blockchain based Logistic Platform

A. International Trade Process

According to [4], the traditional international trade process is mainly composed of five major participants: *importer*, *exporter*, *carrier*, *issuing bank* and *advising bank*.

As shown in Figure 1, the importer is the party who starts the transaction. After the importer and the exporter agree with the trade requirements, the exporter will start to register the product information by the proposed logistics platform.

1) Steps 1-3: The exporter (seller) will negotiate with the importer (buyer) about the details of the goods such as quantity, price, raw material and so on. After the importer and the exporter agree with the trade requirements, the exporter will register the product information.
2) Steps 3.1-5: After the trade is established, the exporter needs to confirm whether the importer is capable of paying the goods since the shipment process is overseas and needs a couple of days to deliver. *Letter of Credit* (L/C) is the document that shows that the importer can offer the payment. To do this, the importer will inquiry its bank (issuing bank) for transferring the L/C to the exporter's bank (advising bank) after they have made the trade agreement.
3) (Steps 6-6.1) After booking the shipping with the carrier, the exporter will send the shipment notification message to the importer.
4) (Steps 7-9.1) After the exporter has loaded the goods on the ship, the carrier will sign the *Bill of Lading* (B/L)

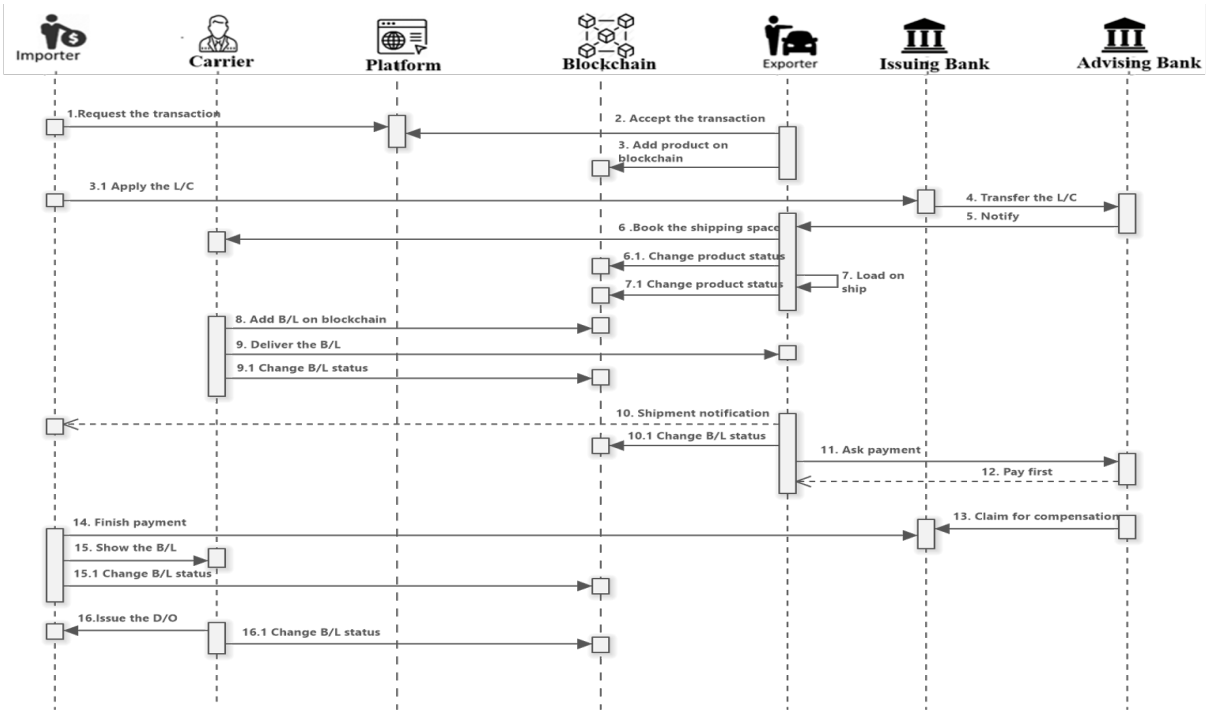

Fig. 1. Sequence Diagram of Blockchain-based Logistic Platform

and deliver the signed B/L to the exporter. At the same time, the importer will get the copy of the signed B/L.

5) (Steps 11-14) When the goods are shipped successfully to the destination port, the exporter will ask its advising bank for getting the payment from the importer's issuing bank. The advising bank sends the claim for compensation to the issuing bank. The importer finishes the payment via its issuing bank and the advising bank will get the payment from issuing bank.
6) (Steps 15-16.1) The importer delivers the copy of the B/L to the carrier after the goods have arrived. Then, the carrier will issue the *Delivery Order* (D/O) to the importer. Finally, the importer can go to the destination port to get the goods via the D/O.

B. Platform Functionalities

In this paper, we consider the following interactions among the exporter, the importer and the carrier, and leave the interactions with the issuing and the advising banks in the future work.

- User Registration and Login:
  During registration, each participant needs to set up its password, account information and his/her role (importer, exporter or carrier) in our platform. After the registration, our platform will issue the user his/her private key, and the user will use his/her password to log into our platform and use his/her private key to add his/her digital signature on the documents.
- Product Registration and Modification: The product registration process is performed by the exporter whose role is seller in the trading process. The modification process is also carried out by the exporter when the product's information (e.g., such as receive time, receive amount, dispatch time, dispatch amount) needs change.
- Product's Current/Historical Status Search:
  The product's status search can be performed by the exporter, importer and shipper.
- B/L Registration and Modification:
  The B/L registration process is performed by the carrier. After the exporter has finished the goods packaging, the carrier can register the B/L. As for the modification part, it is done by the carrier when the information is wrong due to human errors.
- B/L's Digital Signature Display:
  The B/L's digital signature display can be performed by the exporter, the importer and the carrier.
- B/L's Current/Historical Status Search:
  The search of the B/L's status can be performed by the exporter, the importer and the carrier. After the carrier adds the B/L, there are several steps before the product arrives at the destination port. Among these steps, the carrier or the exporter will change the B/L's status information.

C. Smart Contract Design

We implement the following libraries by Solidity to store the information of user (i.e., importer, the exporter and the carrier), product and B/L.

- User Library: The structure `User` is defined in this library to store the information (e.g., account, hashed

password, role and so on) of each user. The getter and setter functions are also defined in User Library.

- Product Library: This library defines the structure `Product` to record the information (e.g., product id, product name and so on) of each product, and implements the corresponding getter and setter functions of each product.
- B/L Library: The structure `BoL` is defined in this library to store the information (e.g., exporter, importer, port of loading, port of discharge, time, date and so on) of each B/L. Besides, it also stores the information of the corresponding D/O. The getter and setter functions are also implemented in B/L Library.

To realize the functionalities mentioned in Section II-B, we implement the following smart contracts by Solidity based on User, Product and B/L Libraries.

1) User Contract: The user registration process and user login process are performed by this contract. User Contract will invoke the `registerUserInfo()` function during the user registration process. `registerUserInfo()` will call the setter functions implemented in User Library to store the user's information into the structure `User`. When a user would like to log into our platform, the platform will check the existence of the user account with the `checkloginUser()` function. When the user's account exists, `checkloginUser()` will call the getter functions of User Library to get the user's information.
2) Product Contract: When a user adds a product into the platform, the function `setProduct()` will be invoked to store the information of the product into the structure `Product`. Besides, the platform will call the `setProductTransferNumber()` function implemented in Product Contract to store the product code and its product state. Then, the `setProductTransferNumber()` function will call the `setProductNumber()` function from Account/Key/Message Contract to store some information such as the product code. Finally, the user will enter his private key so that the product information can be stored into the blockchain. The `getProduct()` function can be used to search the product information, and the `updateProduct()` function can be used to update the product information.
3) Product Transfer Contract: The contract is called when a user would like to alter the product state or look up the current product status. For example, the importer will show the B/L to the carrier and then take the product away. In this scenario, the platform will invoke the `finished()` function to change the state of the product.
4) B/L Contract: The contract is to implement the B/L registration process. When the user adds a new B/L, the `setBoL()` function is invoked to store the information of the B/L into the structure `BoL`. In addition, `setBoL()` will call the `setBoLTransferNumber()` function store the B/L number. Then, the `setBoLTransferNumber()` function will call the `setBoLNumber()` function implemented in Account/Key/Message Contract. The user will enter his/her private key to make the data able to be stored into the blockchain. After the user finishes adding the B/L, the `setBoLDay()` and `setBoLTime()` function will also be invoked to store the data and time of the creation of the B/L. The `getBoL()` and `getBoLDay()` functions are called when the platform would like to query the B/L information.
5) B/L Transfer Contract: The contract implements some functions for the B/L state update process and the B/L status search process. For example, after the exporter loads the products on a ship, the carrier will deliver the B/L to the exporter. In this scenario, it will invoke the `deliverBoL()` function to change the state of the B/L.
6) Account/Key/Message Contract: The contract is designed for storing the digital signature and the corresponding message and recording the product's and B/L's historical transfer states. When adding a product or a B/L, the user needs to input his/her private key and the data will be stored into the blockchain via the `setSignatureandMessage()` function. The structure `BoL_message_list` and `Product_message_list` are defined to trace the historical B/L's and Product's transfer states.

TABLE I
PLATFORM TEST ENVIRONMENT

Description	Version
Back-end Environment	Truffle v5.1.7
Back-end Language	Node.js v10.17.0
Front-end Design	Javascript, JQuery
Smart Contract Design	Solidity
Blockchain Framework	Ganache v2.1.0

III. IMPLEMENTATION

In this paper, we use Solidity to implement the smart contracts and the platform is implemented based on the components shown in Table I. Truffle[1] is a toolset for smart contract development while Ganache[2] is a personal blockchain for the development DApps of Ethereum.

In Ethereum, gas is used as the measurement unit of the operation cost of a transaction. The amount of gas is different between different transactions. We calculate the transaction cost based on the amount of gas and the gas price, which is the price of the gas in terms of Ether. The higher of the gas price, the higher the cost of the transaction. Here, the transaction is related to the deployment of the smart contracts. Table II

[1]Truffle, https://www.trufflesuite.com/truffle
[2]Ganache, https://www.trufflesuite.com/ganache

TABLE II
GAS USED AND OPERATION COST OF SMART CONTRACT DEPLOYMENT

Smart Contract	Gas Used	Gas Cost (Ether)
Product Contract	2594377	0.0518875
Account/Key/Message Contract	2198365	0.0439673
B/L Contract	1927227	0.0385445
B/L Transfer Contract	773101	0.0154620
User Contract	752180	0.0150436
Product Transfer Contract	399624	0.0079925
B/L Library	1383935	0.0276787
Product Library	1370512	0.0274102
User Library	486314	0.0097263

TABLE III
GAS USED AND OPERATION COST OF FUNCTION CALL

Smart Contract Function Call	Gas Used	Gas Cost (Ether)
`addBoL`	758989	0.0151798
`addProduct`	605761	0.0121152
`changeBoLstatus`	178028	0.0035606
`changeProductstatus`	175370	0.0035074
`updateProduct`	174560	0.0034912

shows the costs of the deployment of all smart contracts, while III shows the costs of the calls of some functions implemented in the smart contracts.

IV. CONCLUSION AND FUTURE WORK

In this part, we have designed and implemented a blockchain-based logistics platform for the international trade process. We have implemented several smart contracts so that the processes of user registration, user login, product and B/L registration and modification, product's and B/L's digital signature display, and product's and B/L's current/historical status search can be realized through the collaboration of these smart contracts. In the future, we will extend the platform to support the cash flow. In addition, we will consider more participants such as the shipper and the forwarder in the platform.

ACKNOWLEDGMENT

This work is supported in part by MOST under grants 109-2221-E-009-118-MY3, 110-2222-E-009-005-MY2, and 109-2218-E-009-015.

REFERENCE

[1] C. Xie, Y. Sun, and H. Luo. "Secured Data Storage Scheme Based on Block Chain for Agricultural Products Tracking". In: *Proceedings of the 3rd International Conference on Big Data Computing and Communications (BIGCOM)*. 2017, pp. 45–50.

[2] H. Hasan et al. "Smart Contract-Based Approach for Efficient Shipment Management". In: *Computers and Industrial Engineering* 136 (2019), pp. 149–159.

[3] E. Tijan et al. "Blockchain Technology Implementation in Logistics". In: *Sustainability* (2019).

[4] S. E. Chang, Y.-C. Chen, and T.-C. Wu. "Exploring blockchain technology in international trade: Business process re-engineering for letter of credit". In: *Industrial Management and Data Systems* (2019). DOI: 10.1108/IMDS-12-2018-0568.