

International Trade process : Letter of Credit (LC) Management using Blockchain technology

Md Sadek Ferdous

Brac University
sadek.ferdous@bracu.ac.bd

Asif Idris Tuhin

Shahjalal University of Science and Technology
tuhin123000@gmail.com

Zahid Khandaker Sumon

Shahjalal University of Science and Technology
zksumon2017@gmail.com

October 2021

*This report is prepared and submitted by: **Asif Idris Tuhin - 2017331101***

Abstract

Traditional trade processes suffer from a great number of issues about intermediaries, information latency and trust, which, in turn, hinder overall process efficiency. But still there are lack of proper tools for mitigating this problem. The objective of this report is to represent our endeavor of developing a Letter of Credit (LC) Management project using Blockchain technology. This report also represents the using technique of this Lc management project. We have developed this project on the basis of Hyperledger Fabric 2.2.1 . Importer and exporter can deal and run their business safely as well as easily by using this software.

Keywords

Blockchain; Letter of Credit; Trade Finance; Hyperledger Fabric.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Defining LC	1
1.3	Novelty of this project	1
1.4	Short Overview of the project	2
2	Problem Definition	3
3	Background	4
3.1	How LC works	4
3.2	Important Terms	5
3.2.1	Blockchain	5
3.2.2	Hyperledger Fabric	5
3.2.3	Node.js	5
3.2.4	Utility Server	6
3.2.5	Cloud Database	6
4	Main Methodology	7
4.1	Architectural overview	7
4.2	Chaincode API Access control	8
4.3	NoSql Cloud Database	8
4.4	Hashing Algorithm	8
4.5	System Notification flow	8
5	Features	9
5.1	How to Run it	9
5.2	User Manual	11
5.3	Premade account	12
5.4	Running from different Organization	12
6	Analysis	15

	Date
Starting	19 Aug 2020
Study fundamental Books	19 Aug 2020 - 26 Aug 2020
Time for Selecting Projects	16 Sept 2020 - 4 Oct 2020
First Individual meeting	31 Oct 2020
First Hyperledger Fabric session	6 Nov 2020
Second Hyperledger Fabric session	2 Dec 2020
Second Individual meeting	30 Dec 2020
Subtask 1	30 Dec 2020 - 14 Feb 2021
Third Individual meeting	20 Feb 2021
Subtask 2	20 Feb 2021 - 19 Mar 2021
First Submission	13 Apr 2021
Final Submission	25 Apr 2021
Viva and Presentation	7 May 2021

Table 1: Project Timeline

Chapter 1

Introduction

1.1 Motivation

International trade allows countries to expand their markets and access goods and services that otherwise may not have been available domestically. But, due to the nature of international dealings, including factors such as distance, differing laws in each country and difficulty in knowing each party personally, International trade become risky. So the use if letters of credit(LC) has become a very important aspect of international trade.

1.2 Defining LC

A letter of credit is a letter from a bank guaranteeing that a buyer's payment to a seller will be received on time and for the correct amount. So, it becomes indispensable for international transactions since they ensure that payment will be received. Using documentary letters of credit allows the seller to significantly reduce the risk of non-payment for delivered goods, by replacing the risk of the buyer with that of the banks.

1.3 Novelty of this project

Some analyses and feasibility studies has been conducted to identify and validate the enhancing overall trade performance by blockchain-based international trade process from a perspective of letter of credit [6]. Another work has done regarding a potential paradigm shift in trade finance utilizing blockchain technology [7].

But, still there are lack of such implementation of letter of credit management software using blockchain technology by which importer and exporter can run their business without difficulty. On the other hand, some of the alternative to the Letter of Credit for international trade like Purchase Order Financing has some cons such as While most suppliers would like to be paid by wire transfer, unless your supplier is a fortune 500 company, po finance companies will only pay them using a letter of credit. The do this because a letter of credit ensures that the supplier will only be paid if and when they deliver the goods and more on [8]. So Letter of Credit management using blockchain is very important in this new global Era and We have implemented this project.

1.4 Short Overview of the project

We have used Hyperledger Fabric 2.2.1 for implementing this project [3]. We need Linux operating system(Ubuntu), Docker, Docker Compose, Node js (LTS), cloud database etc. technologies for it. The users who will use this need some of this technologies also.

There are four organizations - org1(Buyers), org2(Sellers), org3(Buyer Bank), org4(Seller Bank), Shown in figure 1.1. Fabric 2.2.1 test network only makes 2/3 orgs. So, we edited the test network scripts for creating 5 orgs. In this project, it is possible to create many seller and buyer member through sign up. But, for the simplicity, we have restricted only one Seller Bank and one Buyer Bank.



Figure 1.1: Four Organizations

Chapter 2

Problem Definition

Letters of credit have become a crucial aspect of international trade , due to the difficulty of knowing each party personally. Using LC allows the seller to significantly reduce the risk of non-payment for delivered goods, by replacing the risk of the buyer with that of the banks.

Blockchain resolves this problem. Using blockchain technology can help streamline the manual processing of import/export documentation, improve security by reducing errors, make companies' working capital more predictable and increase convenience for all parties through mobile interaction. So application of Blockchain to Letter of Credit is very important and it has been done in our project.

Chapter 3

Background

3.1 How LC works

A letter of credit, or “credit letter” is a letter from a bank guaranteeing that a buyer’s payment to a seller will be received on time and for the correct amount. In the event that the buyer is unable to make a payment on the purchase, the bank will be required to cover the full or remaining amount of the purchase.

Figure 3.1 shows the process of LC. Here, *Importer* = *buyer*, *IssuingBank* = *Buyer'sBank*, *Exporter* = *Seller* and *AdvisingBank* = *seller'sBank*.

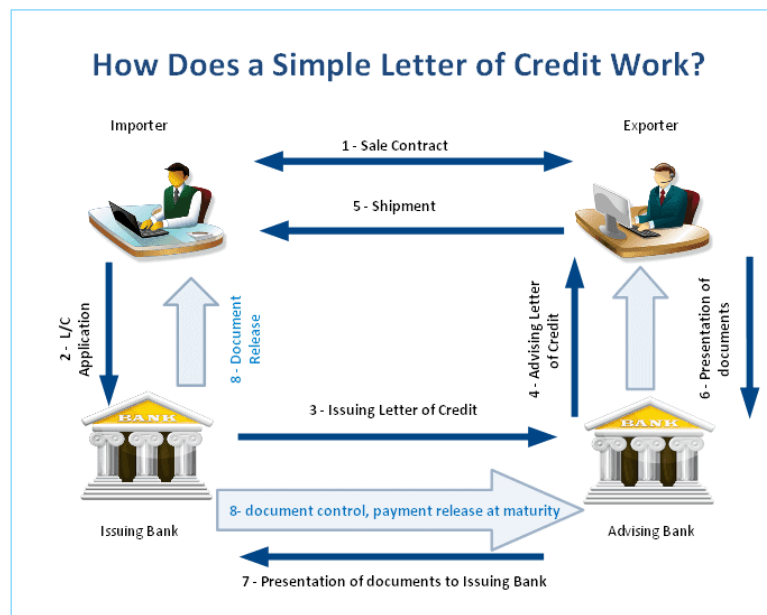


Figure 3.1: LC Process

3.2 Important Terms

Here are some related and important term that we should know before dive into the project.

3.2.1 Blockchain

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). Virtually anything of value can be tracked and traded on a blockchain network, reducing risk and cutting costs for all involved[2].

3.2.2 Hyperledger Fabric

Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture. Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Its modular and versatile design satisfies a broad range of industry use cases. It offers a unique approach to consensus that enables performance at scale while preserving privacy[4].

3.2.3 Node.js

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following, Figure 3.2 “Hello World” example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Figure 3.2: Hello World in node.js

3.2.4 Utility Server

The Utility Server is a virtual machine that can optionally be connected to your RelativityOne instance. It contains additional support tools to help you work with data in your RelativityOne staging area before editing and loading it into your RelativityOne instance. You access your Utility server through a remote desktop connection, using an issued set of credentials and a custom IP address. Once you are connected to your Utility Server, you can perform the following actions:

- **Access a mapped drive for the file share** (TenantUser accounts) - access your uploaded files to edit in the staging area before you add them to your RelativityOne workspaces or save them to a RelativityOne file storage location. You can also access and verify any production sets before you download them locally.
- **Install applications** - if you have TenantAdmin access you can install applications.
- **Manage user administration** - if you have TenantAdmin access and have Terminal Services licensed on the computer you can manage user administration yourself.

3.2.5 Cloud Database

A “cloud database” can be one of two distinct things: a traditional or NoSQL database installed and running on a cloud virtual machine (be it public cloud, private cloud, or hybrid cloud platforms), or a cloud provider’s fully managed database-as-a-service (DBaaS) offering. The former, running your own self-managed database in a cloud environment, is really no different from operating a traditional database. Cloud DBaaS, on the other hand, is the natural database equivalent of software-as-a-service (SaaS): pay as you go, and only for what you use, and let the system handle all the details of provisioning and scaling to meet demand, while maintaining consistently high performance[1].

Chapter 4

Main Methodology

4.1 Architectural overview

As we can see Figure 4.1, the 4 organizations create channel and install chaincode. First two node.js servers can have many users. Though Buyers&Sellers organizations are like a centralized entity, they can never get the users hex secret string which is used for creating private-public key pair. Users' hex secret string never goes through the 4 organizations node.js servers. All users use an utility server for sending all payload(Lc details) and secret hex string and getting public key string and sign.

In real world situations, Users just have to prove that they are the real owner of the public keys mentioned in the ledger and that's very easy.

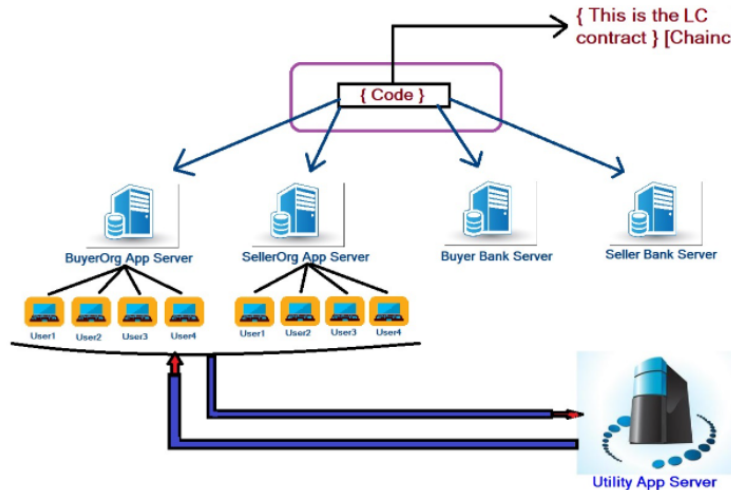


Figure 4.1: Architectural view

4.2 Chaincode API Access control

We made all of the 4 node.js servers. There isn't any function in them which should not be their. Example : seller should not be able to invoke buyers bank verify function. Even if we implement this kind of function in server it will get rejected on the chaincode. Because we are checking the fabric MSP ID in all functions. If there is a contradiction the functions will immediately return without heading further.

4.3 NoSql Cloud Database

All servers (except the utility) are connected to their own MongoDB cloud databases. So, when testing there should be internet connection. We gave all server secrets, email service secrets and MongoDB uri in .env files, So it will work fine.

4.4 Hashing Algorithm

For all the signing and sign verification, we use node js elliptic library. We are using elliptic's EDDSA algorithm[5]. For the curve selection, we have used ed25519[5] which is considered to be a very secure curve. On lc, we are giving unique lc id based on product details string hash's first 8 characters. This hash is done by crypto library's md5 hashing[9]. For password hashing and verifying, we have used bcrypt library.

4.5 System Notification flow

When Buyersbank or Sellersbank does something, it notifies all other 3 servers and the servers save them in cloud database.

When seller makes an LC, it notifies Buyers organization (all saved in cloud).

When buyer approves an LC, it notifies Buyers Bank (all saved in cloud).

Note : All account related information (not secret hex string) like notification list, account, email, password (hashed with bcrypt library) is stored on MongoDB Cloud Database, Our login session is managed by jsonwebtoken and browser cookies (1h expire time).

Chapter 5

Features

5.1 How to Run it

We need to install pm2 module globally in node js as bellow:

```
npm install pm2 -g
```

Figure 5.1: npm install

Now, clone our github repo in the same directory where your fabric-samples folder is located.

```
git clone https://github.com/Sumon2017/hyperledger-fabric-lc-project.git
```

Figure 5.2: git repo

Note: hyperledger-fabric-lc-project and fabric-samples should be on same directory. (see the fig:5.3)

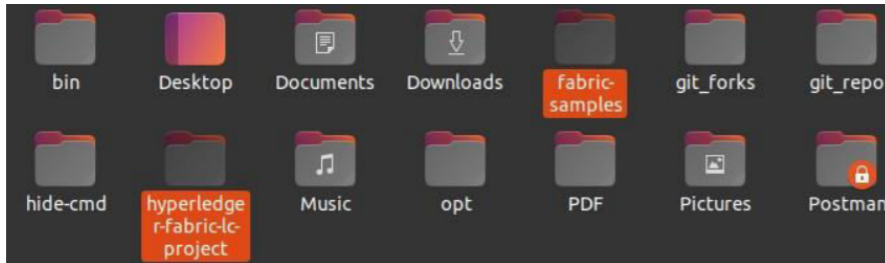


Figure 5.3: Directory

Open up a terminal in **hyperledger-fabric-lc-project/onoff**
Now, from **onoff** directory give execute permission to our scripts.(see in fig 5.4)

```
chmod 777 ./on.sh  
chmod 777 ./off.sh
```

Figure 5.4: Giving Executive permission

Now from **onoff** directory to start the program run :


```
./on.sh
```

Figure 5.5: Starting the program

Cautious: you have to be in **onoff** directory.

This takes between 10-15 minutes depending on your hardware and internet speed. It create the fabric network (creating channel, installing chaincode) and 5 node.js servers (we can see them by [docker ps -a] and [pm2 list]).

To stop it, run :



```
./off.sh
```

Figure 5.6: Stopping the program

Cautious: you have to be in **onoff** directory.

At the end of it, success will be printed.

Now Open browser and make 4 new tabs on the browser and hit the following links:

localhost:3001/index.html

localhost:3002/index.html

localhost:3003/index.html

localhost:3004/index.html

These are BuyersOrgApp, SellersOrgApp, BuyersBank and SellersBank accordingly.

We find some crypto materials in this project but these will get overriten with new valid crypto materials.

5.2 User Manual

- **Step 1:** Sign up in sellersorgapp and buyersorgapp. We get our verification code in your email. If we don't find it, check spam folder. Then verify with it. (if we don't want to do this, we will give two premade accounts for login.)
- **Step 2:** Login for sellersorgapp and buyersorgapp.
- **Step 3:** From sellers organization app create an LC and get pub.key and sign (with hex string secret).
- **Step 4:** create LC from sellers organization app.
- **Step 5:** From buyers organization app check notifications for the LC id and approve with our individual specific sign.
- **Step 6:** From buyersbank check notifications and issue the LC.
- **Step 7:** From sellersbank check notifications and send shipment details. (This should be a string.)

- **Step 8:** From buyersbank check notifications and send transaction money payment details. (This should also be a string). This is the last step and all is done.

Tips : Use dummy strings in shipment details and transaction details for testing purpose. You can also corrupt sign by changing data(after getting pub_key and sign in step 3) and see if buyer/buyers bank takes it.

5.3 Premade account

Table 5.1 showing premade accounts for seller or buyer.

Sellers Org App	Buyers Org App
Username : sumon	Username : tuhin
zksumon2017@disbox.org	tuhin123@disbox.org
Password : 1234	Password : 1234

Table 5.1: Premade Account for seller and buyer

5.4 Running from different Organization

Figures - 5.7, 5.8, 5.9, 5.10, 5.11 is given bellow demonstrating some situations when we run from different organization :

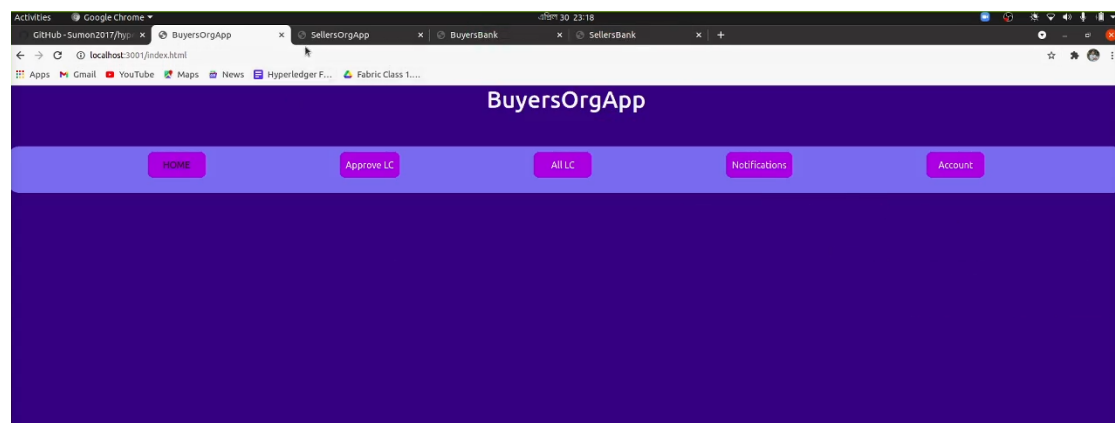


Figure 5.7: Buyer Organization

Sign first on left ... You will automatically get key and sign input filled up on right side. Then press create lc

Your HEX secret string
zxmnxksaklsakji

Buyer's Id(username)
sumon2017

Seller's Id(username)
tuhin2017

Carrier Org Id
besimco

Ammount
50

Product Details
moderna

Expire Time
2

Sign

Your Public Key for Above Sectret
524911d3449947ed1b9ba95a
51666b417e52b048159045a4
a6322d66bb190e1a

Payload Signed
079B0309403EC5C4D2B495F
EE1B9FD8CF61D4BC690FF1
DF1F8AB61456A1D8D9F599
0447E6FE78975DB9F650913
CB7136198FA60728B81F742
07C642FADEC03

Create LC

Figure 5.8: Buyer signing with product,time,seller details

SellersOrgApp

Sign first on left ... You will automatically get key and sign input filled up on right side. Then press create lc

Your HEX secret string
must be HEX (put an easy)

Buyer's Id(username)
username

Seller's Id(username)
username(you)

Carrier Org Id
put anything for testing

Ammount
in USD \$(Integer)

Product Details
put anything for testing

Expire Time
number of days (Integer)

Sign

Your Public Key for Above Sectret
Public Key

Payload Signed
Sign

Create LC

Figure 5.9: Seller signing

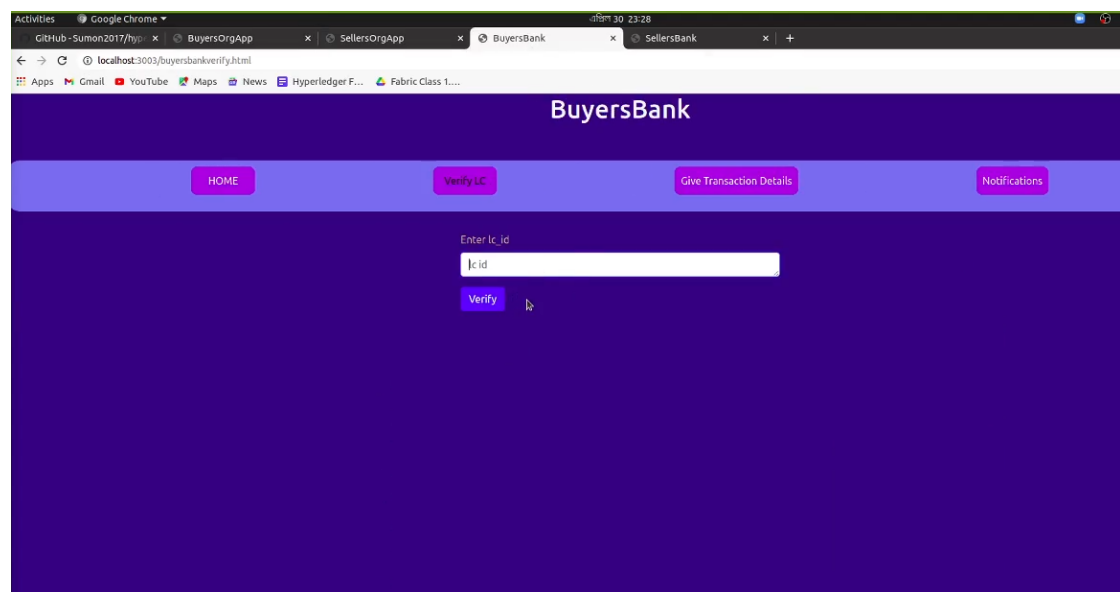


Figure 5.10: Buyers Bank Verifying

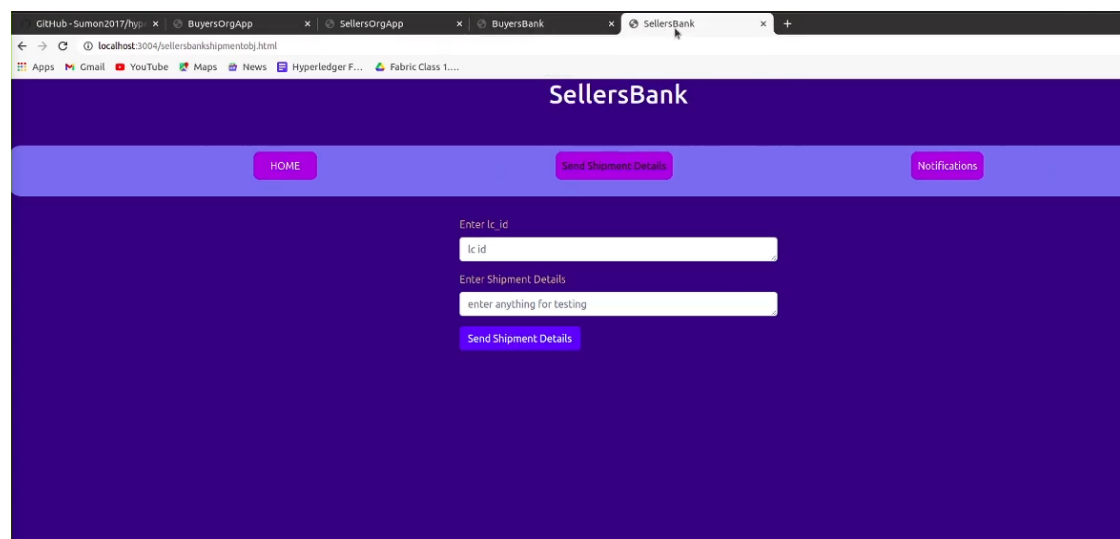


Figure 5.11: Sellers Bank sending shipment details

Chapter 6

Analysis

In our project or system, following activities from different organization are done one by one maintaining these rules :

- At first, Seller initiates the LC.
- Buyer approves LC pointed towards it, if everything is okay with the LC and sellers sign.
- Then Buyers bank issues LC after checking both buyer and seller sign.
- All sign verification is done inside the chaincode (not server).
- Buyers can verify and sign only if the lc_status is initial.
- Buyers bank can verify and issue LC only if the lc_status is approved.
- Sellers bank can send shipment details only if the lc_status is issued.
- Buyers bank can send transaction details only if the lc_status is shipped.
- Everything should be under lc_expire time.

We can say that the main users of project or system are Seller(Exporter), Buyer(Importer), SellerBank(Advising Bank), BuyerBank(Issuing Bank).

Figure 6.1 shows the main activity of our project.

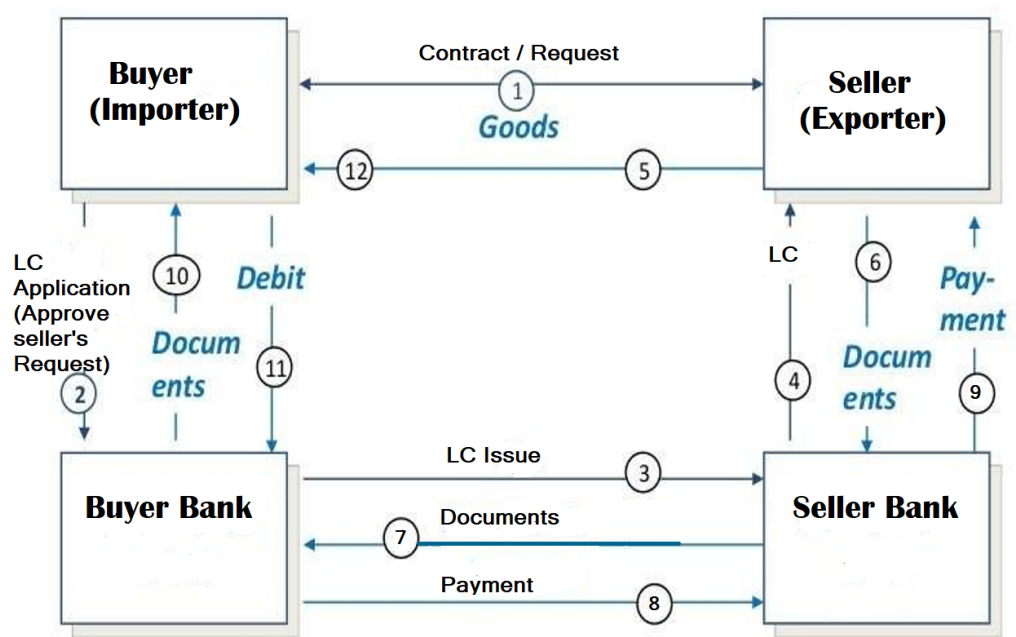


Figure 6.1: Overall Main Activity and its flow

Chapter 7

Conclusion

In this project, The application of blockchain to Letter of Credit can help streamline the manual processing of import/export documentation, improve security by reducing errors, make companies' working capital more predictable and increase convenience for all parties through mobile interaction.

Here are some limitations of our project -

- it is possible to create many seller and buyer member through sign up. But, for the simplicity, we have restricted only one Seller Bank and one Buyer Bank.
- The utility service can also be implemented in complex frontend app or users own managed server. We haven't implemented such kind of utility services.
- We have created four organization app/browser's tab for four organizations. It can be wrapped up within only one app.

The project can be improved by further work on "Invoice & Legal Documents Management" , "Well defined frontend utility service implementation", "Ensuring Account, Stock price accuracy" etc.

References

- [1] Cloud Database, 2018. Accessed May 4, 2021. Available[Online]. <https://www.mongodb.com/cloud-database>.
- [2] What is Blockchain, 2019. Accessed February 10, 2021. Available[Online]. <https://www.ibm.com/topics/what-is-blockchain>.
- [3] Hyperledger Fabric, 2020. Accessed April 3, 2021. Available[Online]. <https://hyperledger-fabric.readthedocs.io/en/release-2.2/index.html>.
- [4] Type: Distributed ledger software, 2020. Accessed April 1, 2021. Available[Online]. <https://www.hyperledger.org/use/fabric>.
- [5] Eddsa, 2021. Accessed April 30, 2021. Available[Online]. <https://en.wikipedia.org/wiki/EdDSA>.
- [6] Shuchih Ernest Chang, Yi-Chian Chen, and Tzu-Ching Wu. Exploring blockchain technology in international trade: Business process re-engineering for letter of credit. *Industrial Management & Data Systems*, 2019.
- [7] Shuchih Ernest Chang, Hueimin Louis Luo, and YiChian Chen. Blockchain-enabled trade finance innovation: A potential paradigm shift on using letter of credit. *Sustainability*, 12(1):188, 2020.
- [8] Qiuping Huang, Xiande Zhao, KwanHo Yeung, Lijun Ma, and Jeff Hoi-yan Yeung. Effects of information-processing mechanisms on internet-based purchase order financing. *Transportation Research Part E: Logistics and Transportation Review*, 146:102222, 2021.
- [9] valdeci. Md5 hashing, 2015. Accessed April 25, 2021. Available[Online]. <https://stackoverflow.com/questions/14733374/how-to-generate-an-md5-file-hash-in-javascript>.