LICT Advanced Blockchain Internship

Letter of Credit on Blockchain

Name	**Khandker Aftarul Islam**
ID	**02**
Mentor	**Dr. Sadek Ferdous**

Table of Contents

Background

As per the World Trade Organization, in 2019 $19.48 trillion worth of merchandise exports were transported around the world across sea, air, rail, road, and up to 80% of this global trade requires financing. Trade finance involves complex cross-border trading with multiple parties such as importers, exporters, banks, financiers, insurers, export credit agencies, and service providers. In such a fragmented environment, many organizations might be doing business with each other for the very first time. As there is no central or intermediary party which can administer the whole process, coordination is cumbersome and investment risks are very high. It's quite unfortunate that even though trade financing and letters of credit (LoCs) have been integral to doing business for decades, the supporting documentation and paperwork nonetheless have many process inefficiencies that increase costs, risks, and delays for all participants. In such cases, an LoC is usually used as a method of payment in international trade, mostly to minimize the overall risk to businesses.

A great deal of LoC is concerned with reconciling divergent histories and facts. Inconsistencies are inevitable given the duplication of complex processes. This leads to further costly reconciliation and dispute-resolution, which is itself error-prone and costly. Multiple views of the same transactions are a source of potentially serious risk.

In this case, a shared-ledger which stores the facts and figures of a transaction along with the sign of all the organizations involved which also does not reveal the internal mechanism of any organization can be a revolutionary solution to this problem.

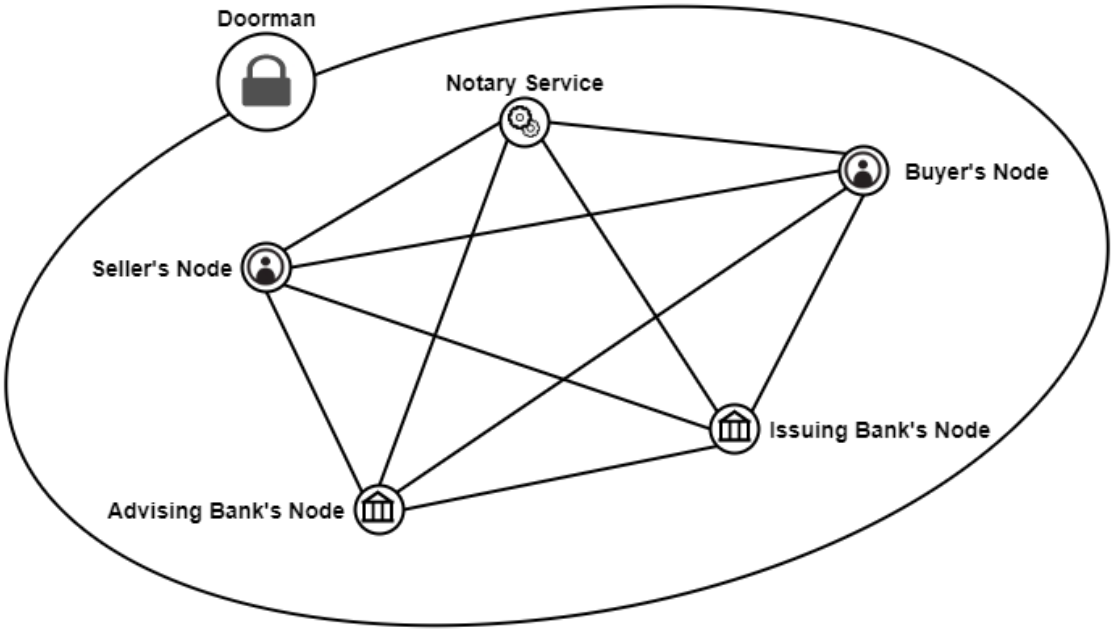
Solution

My proposal is to design a **LoC transaction system on a DLT** which stores all the facts and figures of the contract on a need-to-know basis and also does not reveal the internal business model of any organization.

I have chosen **Corda** as it is the most suitable framework to acquire all the aforementioned targets. In Corda, each organisation maintains a ledger which records the firm's legal agreements and positions with counterparties. Corda provides the network protocol for nodes to exchange messages about possible state transitions and each node verifies for itself if such a state transition is acceptable from a business point of view. Corda does not share data all across the network, only the parties involved and a notary service will get to know about the data. It does not reveal the internal business model of an organization to others. Corda provides non-repudiation, meaning an inarguable history of the shared facts, which is very useful in reconciliation and dispute-resolution.

Design & Architecture

- **The Private Network**

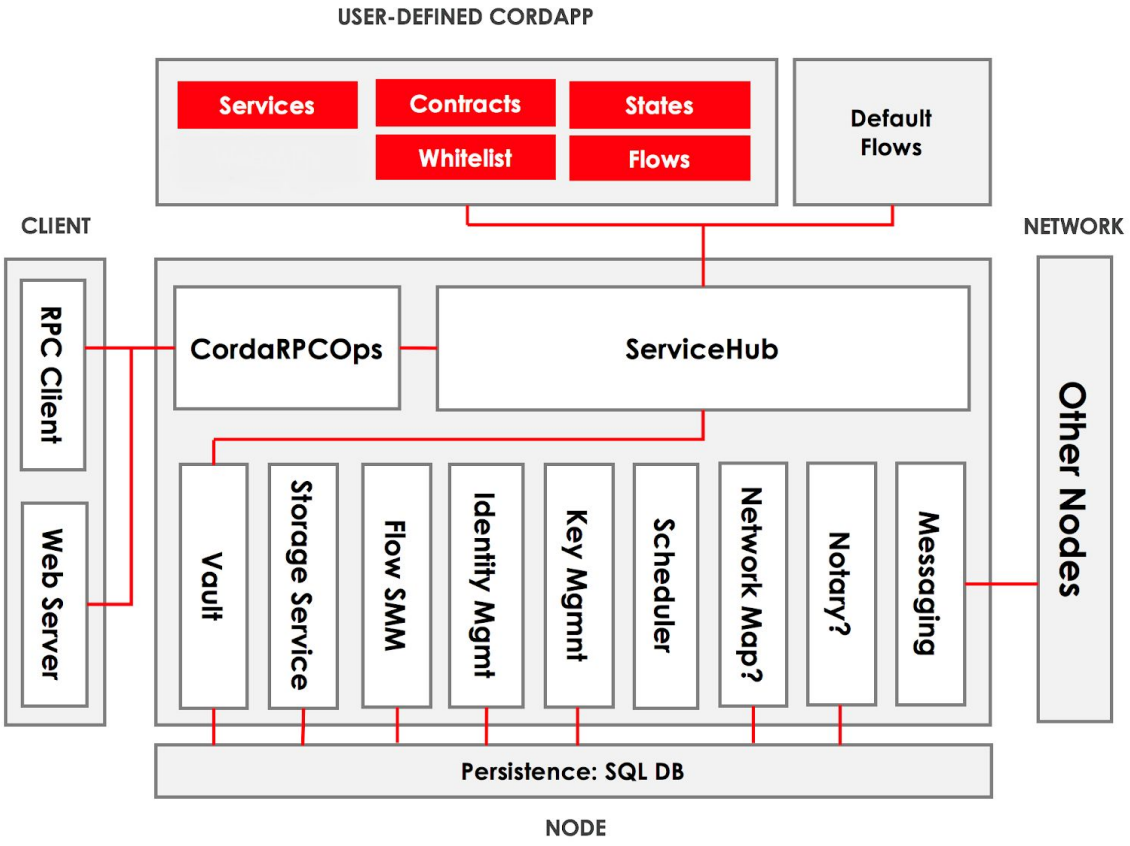


The private network is made up of nodes from all the parties that are required to complete the transaction. All the nodes inside this network can communicate with each other using the AMQP over TLS protocol which is very secure.

The required nodes in this scenario are - Seller's Node, Buyer's Node, Issuing Bank's Node, Advising Bank's Node.

The node that is acting as a notary service provider is necessary for completing a transaction on Corda. It validates the transaction by checking the uniqueness of the input states and also plays an important role to uphold the UTXO model.

- **The Node Architecture**



The core elements of the architecture are:

- A persistence layer for storing data
- A network interface for interacting with other nodes
- An RPC interface for interacting with the node's owner
- A service hub for allowing the node's flows to call upon the node's other services
- A cordapp interface and provider for extending the node by installing CorDapps

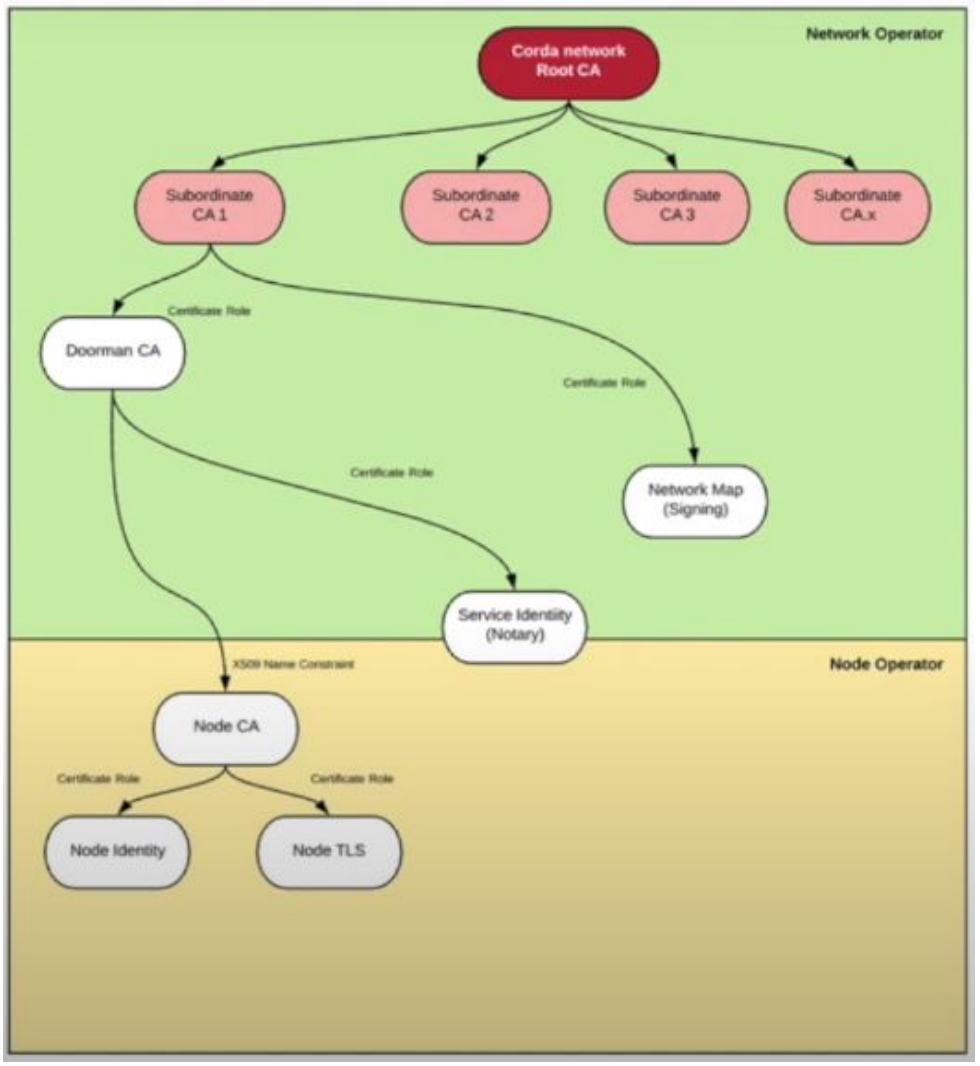


Figure: Corda PKI Hierarchy

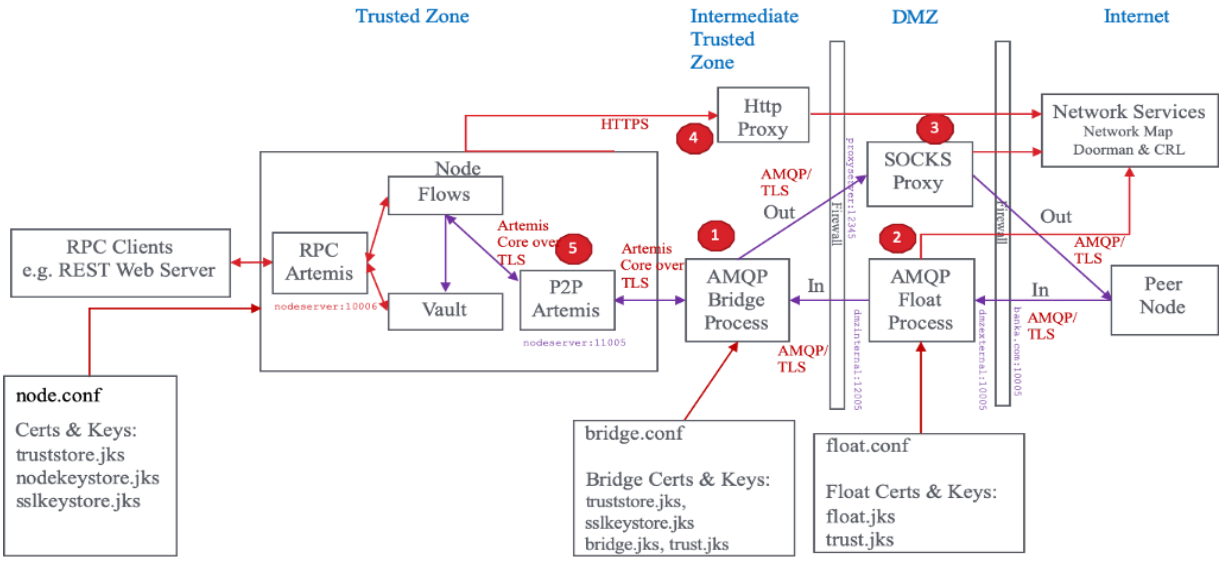
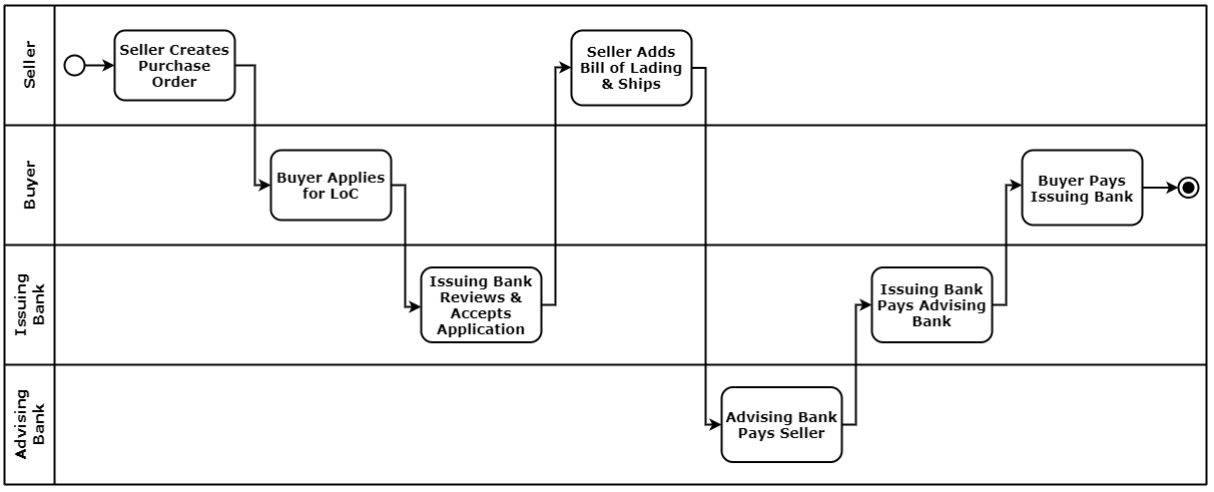


Figure: Node Access Protection Mechanism & Certificates

A Corda network is maintained by the network operator. The network manages the deployment of a node and ensures that a node gets necessary certificates from certificate authorities to access the network. The doorman protects the network from outsiders. A node can only communicate with other nodes in the network if it has required certificates otherwise the doorman will prevent the communication.

The persistent data is stored in the vault of a node. Information regarding other nodes can be accessed via the network map service. The business logic i.e. states, transaction flow is defined and packaged in a corDapp which can be installed on a node to execute a flow. A node can be accessed through a REST API which can communicate with the node using RPC.

- **The Transaction**



There is only one major transaction in this scenario which is the trading based on LoC. Before the transaction, the seller and buyer agree to go on with the trade. They decide the price and the quantity of goods that will be traded as well. Once all of these have been decided, we proceed with the transaction.

Firstly, the seller creates a purchase order with necessary information (Quantity of Goods, Price, Delivery Schedule, Advising Banks Information etc). Then, the buyer applies for LoC to the issuing bank with its credentials and necessary information regarding the trade. Then, the Issuing bank checks the information provided by the buyer and accepts it if the information is valid and they agree with the terms. Then, the seller adds the Bill of Lading to the transaction and ships the goods to the buyer. After that, the Advising bank pays the seller, The Issuing Bank pays the Advising bank. Finally, the buyer pays the Issuing bank.

First of all, we will create a LetterOfCreditApplicationState that all the four primary stakeholders have to sign. It can have the following variables:

- Buyer Organization (Party)
- Issuing Bank (Party)
- Seller Organization (Party)
- Advising Bank (Party)
- expiryDate
- portOfLoading
- portOfDischarge
- descriptionOfGoods
- amount

Then, another LetterOfCreditDeliveryState signifies the status of delivery of goods. This gets updated as the goods are shipped and gets loaded and unloaded port by port until they reach their destination. All stakeholders have to track this for fulfillment of the order and payment. Itcan have the following variables:

- Applicant (Party)
- Beneficiary (Party)
- AdvisingBank (Party)
- IssuingBank (Party)
- status
- paymentAmount
- applicationDate
- expiryDate
- portOfLoading
- portOfDischarge
- locId
- linearId

Finally, we can have a ReceiptOfGoodsState that will be generated by the final port where deliverable goods reach as per contract. The variables it should contain are as follows:

- Applicant (Party)
- Beneficiary (Party)
- AdvisingBank (Party)
- IssuingBank(Party)
- timeOfDelivery
- descriptionOfDeliverables
- locId
- linearId

The flows can be as follows:

- LoCApplicationFlow: LetterOfCreditApplicationState is created
- LoCApprovalFlow: LetterOfCreditApplicationState is approved
- ShipmentFlow: LetterOfCreditDeliveryState is created and updated from time to time till goods reach destination
- PaymentFlow: LetterOfCreditDeliveryState status is updated with payment.

Risks & Challenges

The main challenge of this project is implementing a functional blockchain application in a very limited time period. On completion of the project the application might not be production ready and the chances of having a bug in the application is high.

The notable shortcomings of this project are listed below:

- No Buyer & Seller protection system
- Minimal details of shipment state
- Minimal details of product
- No involvement of ports as parties in the transaction
- The real payment is not automated & handled off-chain

Due to these shortcomings, there are also some risks involved in this project. Though the shared-ledger can be used in reconciliation and dispute-resolution, due to lack of on-chain buyer & seller protection system these problems have to be resolved off-chain and manually. The status of the product and shipment could have been more detailed if we could on-board ports as parties in the network, so that these ports could update the shipment state with necessary information of the shipment. As the payment is done off-chain and later the payment state is updated on-chain, the payment status needs to be checked manually and verify the change of payment state which could cause many serious issues.