Trading & Letter of Credit

Blockchain Ninja Team

|  |  |
| --- | --- |
| Document Owner: | Abhay Prasad |
| Authorised by: |  |
| Version: | 1.0 (Draft) |
| Date Released: | 2018/12/03 |
| Status: | In Progress |

Table of Contents

[1 Introduction 4](#_Toc531465042)

[1.1 Version Control 4](#_Toc531465043)

[1.2 Distribution List 4](#_Toc531465044)

[2 General Overview 5](#_Toc531465045)

[2.1 Shared assets and data 6](#_Toc531465046)

[2.2 Participants' roles and capabilities 6](#_Toc531465047)

[3 Implementation using Hyperledger Fabric 8](#_Toc531465048)

[3.1 Designing a network 8](#_Toc531465049)

[3.2 Solution Information 9](#_Toc531465050)

[3.3 Implementation 10](#_Toc531465051)

[3.3.1 Get Access Token 10](#_Toc531465052)

[3.3.2 Get Store Details 10](#_Toc531465053)

[3.3.3 Get list of matching stores 10](#_Toc531465054)

[3.3.4 Get CBO version from Database 10](#_Toc531465055)

[3.3.5 Identity Service JSON Parameters 10](#_Toc531465056)

[3.3.6 JSON Draft (example) 11](#_Toc531465057)

[3. WS LAYER 12](#_Toc531465058)

[4.1 Similar Store Web App🡨>Tesco Identity Service 12](#_Toc531465060)

[4.1.1 WS (Tesco Identity Service) 12](#_Toc531465061)

[4.2 Similar Store Web App 🡨>Tesco Location Service 14](#_Toc531465062)

[4.2.1 WS (Tesco Location Service) 14](#_Toc531465063)

[4.3 Similar Store Web App 🡨> Tesco Location Service 15](#_Toc531465064)

[4.3.1 WS (Tesco Location Service) 15](#_Toc531465065)

[4.4 Set PUT/GET/POST/DELTE Permissions in the Environment 15](#_Toc531465066)

[5 Implementation using R3 Corda 18](#_Toc531465067)

[6 DB Schema&Definition 19](#_Toc531465068)

[6.1 Definition 19](#_Toc531465069)

[6.2 Schema 19](#_Toc531465070)

[6.3 Sql query 19](#_Toc531465071)

[5. Integration Testing 20](#_Toc531465072)

# Introduction

## Version Control

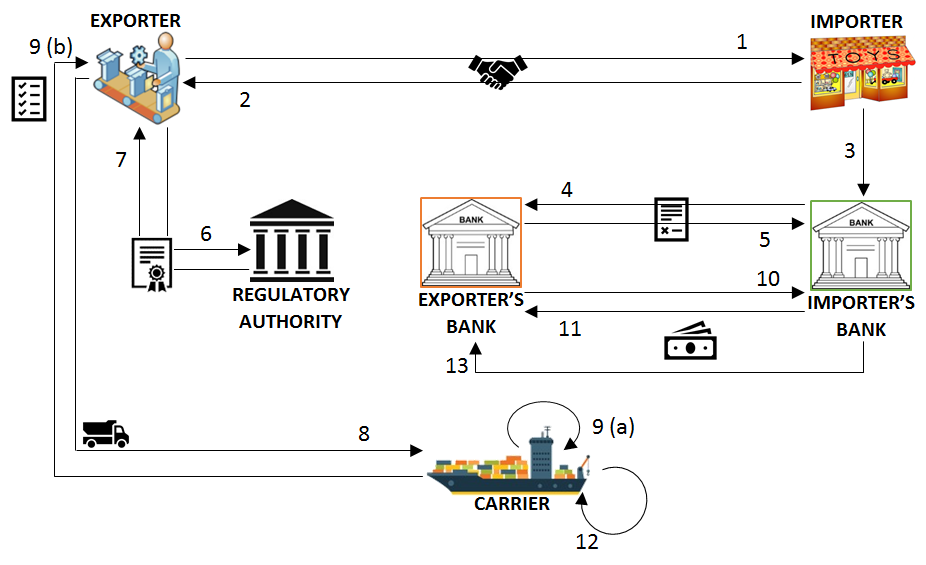
|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Changes |
| 1.0 | 03/12/2018 | Abhay Prasad | First Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Distribution List

|  |  |  |
| --- | --- | --- |
| Name | Email Address | Title |
| Satya |  | Senior Technology Architect |
| Mahendran |  | Senior Technology Architect |
| Rohini |  | Delivery Manager |
|  |  |  |

# General Overview

It is to be developed a system able to communicate among each other to facilitate a transaction: the sale of goods from one party to another.



The transactions in our workflow are as follows:

1. Importer requests goods from the exporter in exchange of money
2. Exporter accepts the trade deal
3. Importer asks its bank for an L/C in favor of the exporter
4. The importer's bank supplies an L/C in favor of the exporter, and payable to the latter's bank
5. The exporter's bank accepts the L/C on behalf of the exporter
6. Exporter applies for an E/L from the regulatory authority
7. Regulatory authority supplies an E/L to the exporter
8. Exporter prepares a shipment and hands it off to the carrier
9. The carrier accepts the goods after validating the E/L, and then supplies a B/L to the exporter
10. The exporter's bank claims half the payment from the importer's bank
11. The importer's bank transfers half the amount to the exporter's bank
12. The carrier ships the goods to the destination
13. The importer's bank pays the remaining amount to the exporter's bank

## Shared assets and data

The participants in the previous workflow must have some information in common that gives them a view into the trade arrangement and its progress at any given moment.

The following is a table of the assets owned by the participants, which are shared with each other to drive the process from one stage to the next. This includes documentary and monetary assets:

|  |  |
| --- | --- |
| **Asset type** | **Asset attributes** |
| Letter of credit | ID, issue date, expiration date, issuer, beneficiary, amount, and a list of documents |
| Bill of lading | ID, shipper (exporter), consignee (importer), party to notify (importer's bank), places of receipt and delivery, description of goods, and freight amount |
| Export license | ID, issue date, expiration date, beneficiary, license holder, and description of goods |
| Payment | Amount in standard currency units |

The following are the data elements that circumscribe the options available to participants in each stage:

|  |  |
| --- | --- |
| **Data type** | **Data attributes** |
| Trade agreement | Requested by importer and accepted by exporter |
| Letter of credit | Requested by importer, issued by importer's bank, and accepted by exporter's bank |
| Export license | Requested by exporter and issued by regulatory authority |
| Shipment | Prepared by exporter, accepted by carrier, and current position or location |

## Participants' roles and capabilities

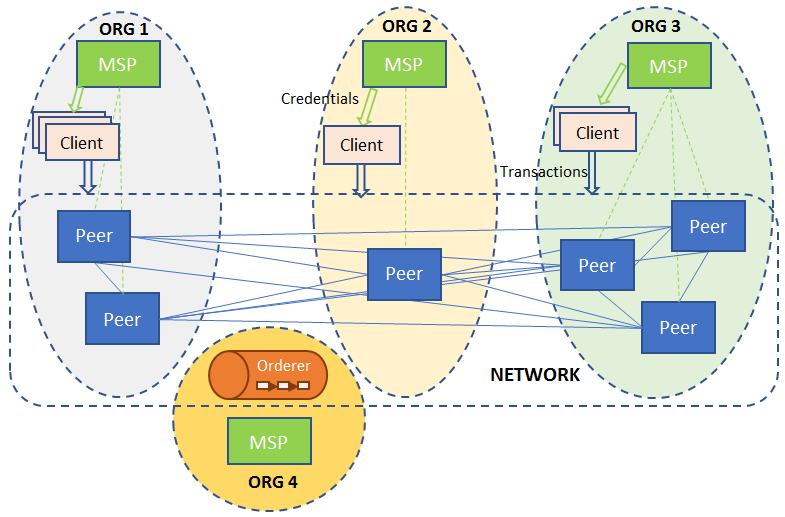
There are six categories of participants in our scenario: which are exporter, importer, exporter's bank, importer's bank, carrier, and regulatory authority. The terms in this set refer to the roles an entity can assume in a trade deal; for example, a company exporting goods in one instance may be an importer in another. The capabilities and restrictions of each role are also detailed in the following list:

* Only an importer may apply for an L/C
* Only an importer's bank may supply an L/C
* Only an exporter's bank may accept an L/C
* Only an exporter may request an E/L
* Only a regulatory authority may supply an E/L
* Only an exporter may prepare a shipment
* Only a carrier may supply a B/L
* Only a carrier may update a shipment location
* Only an importer's bank may send money, and only an exporter's bank may receive money

# Implementation using Hyperledger Fabric

## Designing a network

The following diagram illustrates a typical peer network structure with clients, MSPs, and logical organization groupings.



The components are:

* **Similar Stores Web App**:System responsible for interacting with Identity, Location service and Frontoff Database.
* **Tesco Identity Service**: responsible of receiving client id, username, password & scope and provides the access token information.
* **Tesco LocationService**: offers the services needed for getting branch related information.
* **Database**: storage system composed of main DB and stores all branch related information.

The Organizations with their MSPs, peers, and clients of our trading network are illustrated in the following diagram:

## https://www.safaribooksonline.com/library/view/hands-on-blockchain-with/9781788994521/assets/a00229d0-2a2f-44fa-a721-97619dad4d59.png

## Solution Information

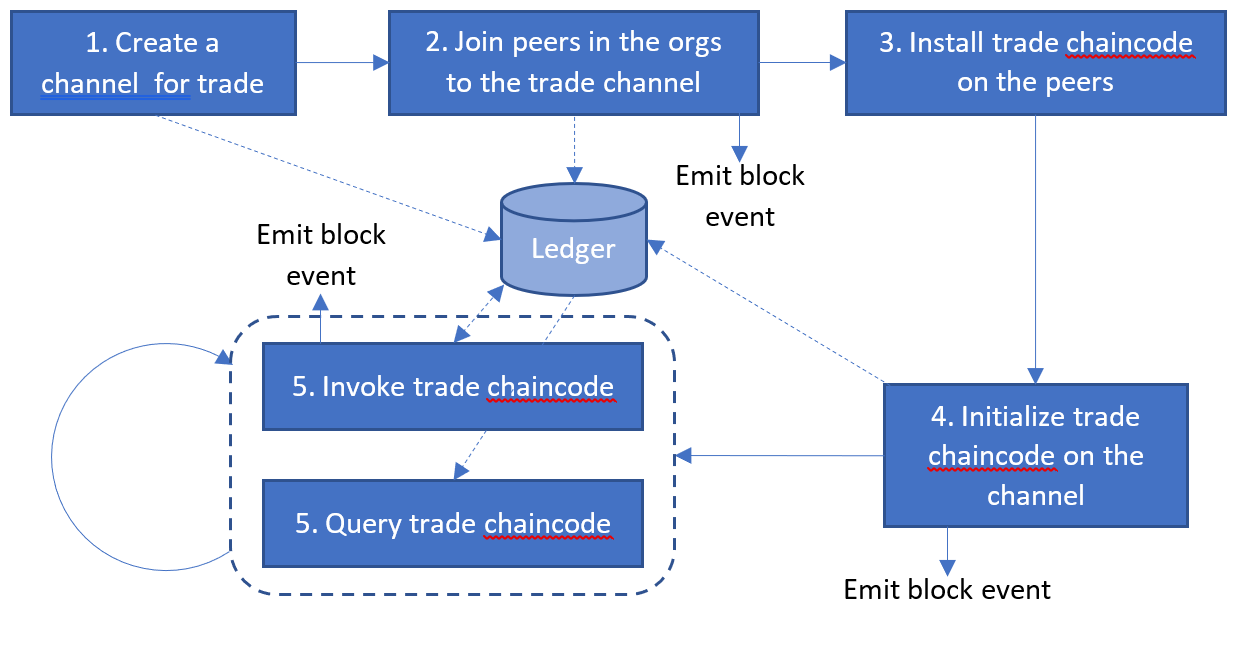
The implementation of this solution will be resolved in an efficient and balanced way so that users will get the list of similar stores matching the CBO version, countryand store type of a given store.

It is divided in following different developments components:

* Get the identity token from the Tesco Identity Service
* Pass the identity token and branch number to the Tesco Location Service to get the store details.
* Pass the country and store type fetched from above service call as a filter attribute to Tesco Location Service to get all matching stores.
* Connect to each matching store SQL Server FrontOff DB to get CBO Version. Stores having the same CBO version as that of original input store will be selected as Similar Stores.

## Implementation

The components development is thought in the following way.



### Get Access Token

Create anIdentity Service callto<https://identity.api.tesco.com/>which contains a method accepting client id, grant type, username and password. This method will then return the access token.

### Get Store Details

In order to get the single location / store information from the location service (<https://api.tesco.com/tescolocation/v3/locations>) we need to pass branch number to the*search*methodwhich will return the given store detail.

### Get list of matching stores

After getting the given store details, parse the JSON result to fetch the countryandstore type. Pass these attributes as a filter criteria to the location service (<https://api.tesco.com/tescolocation/v3/locations>)*search* method which will return the list of matching stores.

### Get CBO version from Database

The *ExecuteReader* method will fetch the CBO version fromFrontOff database with the latest version information.

### Identity Service JSON Parameters

* **Client\_id:**
* **Grant\_type:**
* **scope:**
* **username:**
* **password**

### JSON Draft (example)

{

"client\_id":"trn:tesco:cid:254deb72-b04d-4dd3-afdb-5c4223885b57:3499d5ce-1538-4dac-9719-38abdc470604",

"grant\_type":"password",

"scope":"oob",

"username":"PLMS",

"password":"@2gn9G+Medgyw7YS"

}

# 3. WS LAYER



There are three different WS layers to be used for this project:

1. Similar Store Web App🡨>Tesco Identity Service
2. Similar Store Web App🡨>Tesco Location Service (Get single store detail)
3. Similar Store Web App🡨> Tesco Location Service (Get similar stores list)

## Similar Store Web App🡨>Tesco Identity Service

|  |  |
| --- | --- |
| WS (Tesco Identity Service) | |
| **Desciption** | Obtain access token | |
| **URL** | <https://identity.api.tesco.com/v3/api/auth/oauth/v2/token> | |
| **HTTP Method** | POST | |
| **Mandatory URL Params** |  | |
| **Optional URL Params** |  | |
| **Headers** | Content-Type : application/json | |
| **Sample request** | <https://identity.api.tesco.com/v3/api/auth/oauth/v2/token> | |
| **Body** | {  "client\_id":"trn:tesco:cid:254deb72-b04d-4dd3-afdb-5c4223885b57:3499d5ce-1538-4dac-9719-38abdc470604",  "grant\_type":"password",  "scope":"oob",  "username":"PLMS",  "password":"@2gn9G+Medgyw7YS"  } | |
| **Sample success response** | {  "access\_token": "acee431c-eed7-4862-8292-369c36c6fb0e",  "token\_type": "Bearer",  "expires\_in": 3600,  "refresh\_token": "c1033fd1-e8fc-4f04-a000-d31253b4c5ff",  "scope": "oob",  "Claims": [  {  "claimType": "http://schemas.tesco.com/ws/2011/12/identity/claims/clientid",  "value": "trn:tesco:cid:254deb72-b04d-4dd3-afdb-5c4223885b57"  },  {  "claimType": "http://schemas.tesco.com/ws/2011/12/identity/claims/scope",  "value": "oob"  },  {  "claimType": "http://schemas.tesco.com/ws/2011/12/identity/claims/userkey",  "value": "trn:tesco:uid:uuid:f40c7c67-490f-4c30-90af-cfa0a644038a"  },  {  "claimType": "http://schemas.tesco.com/ws/2011/12/identity/claims/confidencelevel",  "value": "12"  },  {  "claimType": "http://schemas.microsoft.com/ws/2008/06/identity/claims/expiration",  "value": "1528116457"  }  ]  } | |
| **Sample error response** | NULL | |
| **Error Codes** | N.A | |
| **Comments** |  | |

## Similar Store Web App 🡨>Tesco Location Service

|  |  |
| --- | --- |
| WS (Tesco Location Service) | |
| **Title** | WS Get single Store Details |
| **Desciption** | Get store details based on branch number |
| **URL** | <https://api.tesco.com/tescolocation/v3/locations/> |
| **HTTP Method** | GET |
| **Mandatory URL Params** | filter (search criteria) |
| **Optional URL Params** |  |
| **Headers** | Authorization : bearer a239fab3-ab50-403a-aa8a-21f076df3d6a |
| **Sample request** | <https://api.tesco.com/tescolocation/v3/locations/search?filter=branchNumber:2654> |
| **Sample success response** | <https://jsoneditoronline.org/?id=19a13eb738aa40269c29a91a0c8cb1ef> |
| **Sample error response** | NULL |
| **Error Codes** |  |
| **Comments** |  |

## Similar Store Web App 🡨> Tesco Location Service

|  |  |
| --- | --- |
| WS (Tesco Location Service) | |
| **Title** | WS Get similar store list |
| **Desciption** | Get similar store list based on a given store number |
| **URL** | <https://api.tesco.com/tescolocation/v3/locations/> |
| **HTTP Method** | GET |
| **Mandatory URL Params** | filter (search criteria) |
| **Optional URL Params** |  |
| **Headers** | Authorization : bearer a239fab3-ab50-403a-aa8a-21f076df3d6a |
| **Sample request** | <https://api.tesco.com/tescolocation/v3/locations/search?filter=isoCountryCode:GB+AND+type:Extra> |
| **Sample success response** | <https://jsoneditoronline.org/?id=39d9e31efe2048a2bfcab54cfebb3873> |
| **Sample error response** | NULL |
| **Error Codes** |  |
| **Comments** |  |

## Set PUT/GET/POST/DELTE Permissions in the Environment

Select *Edit Permissions* for the *Default Web Site:*

Then select *Security* tab and click on *Edit* button:

Then select the *Users* group and grant them *Full control* rights:

Then restart the server to reflect the changes:

User permissions will be inherited by all the subfolders, granting write access to web application

# Implementation using R3 Corda

# DB Schema&Definition

## Definition

## Schema

See Schema below:

USE [FrontOff]

GO

/\*\*\*\*\*\* Object:  Table [dbo].[VERSION]    Script Date: 05/24/2018 13:13:31 \*\*\*\*\*\*/

SETANSI\_NULLSON

GO

SETQUOTED\_IDENTIFIERON

GO

SETANSI\_PADDINGON

GO

CREATETABLE [dbo].[VERSION](

      [VER] [varchar](25)NOTNULL,

      [NOTE] [varchar](255)NULL,

      [VER\_DATE] [datetime] NULL,

      [PACKAGE\_VAR] [varchar](50)NULL,

      [APP\_PATH] [varchar](1024)NULL,

CONSTRAINT [PK\_VERSION] PRIMARYKEYCLUSTERED

(

      [VER] ASC

)WITH(PAD\_INDEX  =OFF,STATISTICS\_NORECOMPUTE  =OFF,IGNORE\_DUP\_KEY=OFF,ALLOW\_ROW\_LOCKS  =ON,ALLOW\_PAGE\_LOCKS  =ON)ON [PRIMARY]

)ON [PRIMARY]

GO

SETANSI\_PADDINGOFF

GO

## Sql query

SELECT top 1 PACKAGE\_VAR From VERSION order by VER\_DATE desc

# 5. Integration Testing