National College of Ireland

Analysing the Evolution of Permissioned Blockchain in Financial Sector

MSc Research Project
FinTech

Sean Khye Lee
Student ID: x19116888

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sean Khye Lee

Student ID: X19116888

Programme: MSc in Fintech **Year:** 2020

Module: Research Project

Supervisor: Victor Del Rosal

Submission Due Date: 17th August 2020

Project Title: Analysing the Evolution of Permissioned Blockchain in Financial Sector

Word Count: 6929 **Page Count** 24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: .....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Analysing the Evolution of Permissioned Blockchain in Financial Sector

Sean Khye Lee
X19116888

Abstract

Blockchain has become part of the Fintech development to disrupt finances, increase technology adoption and eventually provide the alternative solution for banking industry. Hyperledger, permissioned blockchain platform uses smart contracts to function as normal application without any intermediaries. The smart contract was extended to include the financial sector such as the bank. Various tasks related to blockchain technology in bank to enhance data security and regular interactions between customers had been implemented. A Multon Dapp for the Hyperledger Fabric has been proposed in financial sector using the Hyperledger Composer tool. Tools use are the Oracle Virtual Box, Sublime Text, Angular, GitHub, command line interface, Hyperledger Composer, Apache JMeter, and Tableau. Security and scalability in the application have been well evaluated. Results show that security is protected by the unique identity for Hyperledger Composer while the transaction in the application is secured by the transaction id and timestamp. Not only that, different experiments are conducted successfully in JMeter to compare the average response time with the different number of users. Finally, all the codes are executed in the design product is finalised in the GitHub repository.

1 Introduction

Fintech has influenced the traditional financial sector. After the financial crisis of 2008, the financial sector's landscape shifted due to the increased financial supervision and development in financial technology. World Economic Forum (WEF) suspected in 2016 that blockchain technology could revolutionise banking services in the world by creating a network that directly links customers and businesses (Yoo, 2017) [1]. Blockchain could bring significant value to a range of financial services activities ranging from commercial finance to payments, settlement of shares and adherence to regulations. So, they could contribute to overcoming certain traditional bank inefficiencies, such as the transfers of foreign exchange (FX), increasing existing business networks and growing discoverability and trust working with bank payment and messaging systems. The application of blockchain technology in commercial banking undermines traditional financial banking.

The blockchain uses 'Smart Contracts' that automatically record and carry out a transaction that can dramatically reduce costs (Ahmed & Kumar, January 10-11, 2019). Blockchain's benefits in automating finance processes are immense and can save costs and time to achieve every transaction. The explanation that blockchain technology fascinates the

[1] https://www.weforum.org/press/2016/08/blockchain-will-become-beating-heart-of-the-global-financial-system/

financial industry is that blockchain features enable people to build confidence and accommodate to new banking technology. If blockchain is widely implemented, banks will need to build a framework to construct and operate on a genuine worldwide basis using frameworks based on this revolutionary invention, and to minimise costs and exceptions rapidly and accurately. Therefore, the use of blockchain in banking would definitely contribute to a better banking system.

The figure 1 shows the adoption rate for FinTech worldwide in 2019. FinTech adoption has gained popularity across several nations such as China, India, Russia and South Africa.

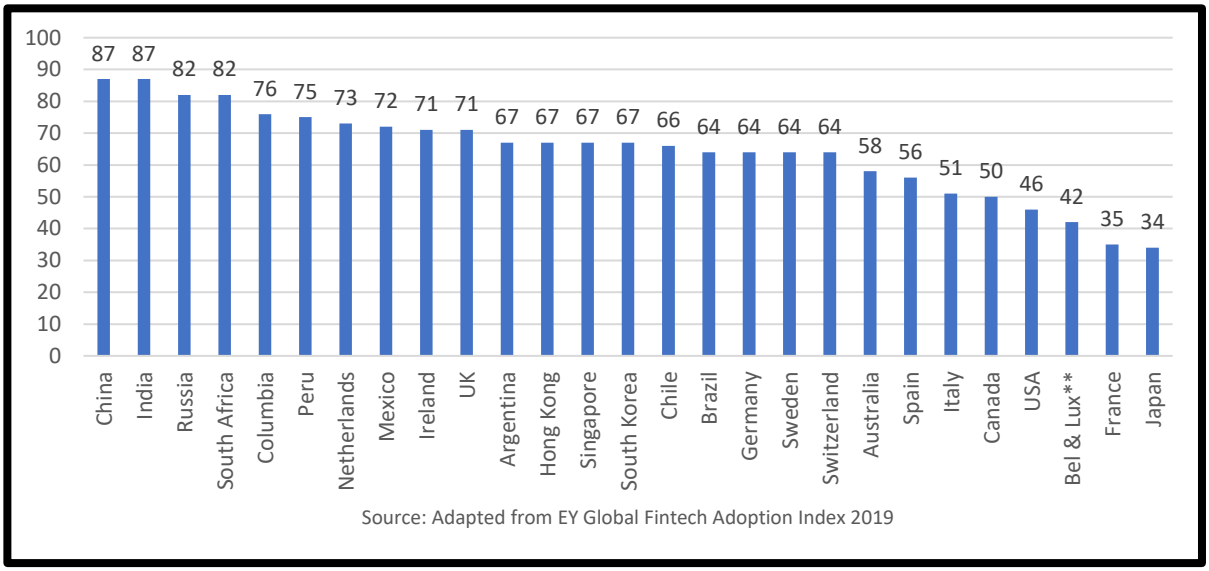

Figure 1: Worldwide Adoption Rate of Fintech (%) [2]

1.1 Motivation

Commercial banks act as the critical value for the countries must comply with the privacy and security regulations such as GDPR. Being a special financial company, the commercial bank faces more financial risks than the traditional banking. Therefore, all bank requires to coordinate data security and track internal space limitations to protect against leakage of information. The geographical distribution of bank infrastructure in the modern world complicates this task: the existence of branches, cash distributors and other services through which the headquarters interacts through information (Popova & Butakova, January 28-31, 2019). Geographically distributed systems should also be secured against internal failures leading to data loss and application disruption.

Additional criteria for business-use applications include scalability, security, privacy, confidentiality and integrity (Ermyas Abebe, December 9-13, 2019). Such conditions led to the development of permissioned blockchain networks. Permissioned blockchains have emerged as an alternative to permissionless blockchains (in which anyone may participate, for example, Bitcoin and Ethereum) to tackle the need for operating blockchain technology among a variety

[2] https://eyfinancialservicesthoughtgallery.ie/wp-content/uploads/2019/09/ey-global-fintech-adoption-index.pdf

of recognised and identifiable users who need special access to the blockchain network (Vukoli´c, April 2, 2017). This idea behind permissioned blockchains is particularly interested in the business applications of blockchain technology and distributed ledgers, in which participants need some ways to recognise each other without actually trusting each other entirely. There are today a broad variety of protocol and application choices for developing approved networks. The Hyperledger Project, a popular Linux Foundation[3] open-source initiative dedicated to introducing blockchain technology to businesses with a strong emphasis on permissioned blockchains.

1.2 Research Question

"How can Hyperledger Fabric for Bank Dapps be implemented in private blockchain to achieve security and scalability for commercial banking applications?"

1.3 Objective

The literature uses to generate this project is: Journal or conference papers along with reports from existing international institutions. A reviewed of the literature allows us to identify contributions to the issue that contributed to the research question being proposed. While there have been abundant of literature discussed the security and scalability of blockchain technology but there has been only minimal work in connection with the blockchain technology in the banking sector. This study aims to show how the Hyperledger Fabric Multon Dapp with Hyperledger Composer is able to access security and scalability efficiency.

This project is organised as follows. Section 2 discusses the related works regarding the blockchain in financial sector and comparison of blockchain platform. Section 3, 4, and 5 show the research methodology, design specification, and implementation of Multon Dapp respectively. Section 6 explains the result in detail about the performance of security and scalability. Last section defines the conclusion which outline the future work of the research.

2 Related Work

2.1 Why Blockchain in Financial Sector?

One of these interviewees in (Chang, et al., 2020) said "The blockchain is decentralised, while the bank is a centralised system. If the underlying technology of the blockchain can be used to make a centralised system, I think the blockchain technology can be used in the banking industry." Wu & Duan (June 21-23, 2019) explored the benefits of blockchain technology by commercial banks on the following aspects such as distributed ledger structure, consensus process, and asymmetric cryptography. The implementation of blockchain in commercial banks can effectively reduce transaction costs and increase banks' operational efficiencies,

[3] https://www.hyperledger.org/about

particularly in billing, cross-border payment and asset securitisation. Lastly, blockchain technology will reduce transaction costs on both sides and improve the commercial bank's operational efficiency and management. Sakho, et al. (December 6-9, 2019) set up a private and secure blockchain to reduce transaction costs, trust problems and theft and falsification of banking details. This system implemented user transactions and then encrypted information for bank transactions on the blockchain. Blockchain will help banks to monitor and evaluate all banking system activities.

Popova & Butakova (January 28-31, 2019) examined the using of blockchain technology without tokens to secure information on banking transactions, i.e. payments, card information, participant names etc. Thus, the blockchain is a tool that will alleviate the processes of maintenance of integrity and unique data on banking transactions when implemented in the banking system without mining and tokens and will allow its implementation through smart contract processes to reduce the number of participants in overseeing such transactions. Singh, et al. (September 28-29, 2018) presented the proposed Secure E-Wallet Architecture based on Digital Ledger Technology (DLT) from various banks and participating institutions that use blockchain for the India financial sector. The architecture of interoperability between e-wallets from different banks or organisations is also implemented by a swarm-based peer to peer network. The proposed solution would reduce the demand on the banks' Core Banking Solution while reducing the burden on the servers at the data centres. Abdulkader, et al. (July 30-31, 2019) analysed the potential security risks and economic benefits of a National Central Bank (NCB), which seamlessly provides all private banks with the alteration of the Digital Fiat Currency (DFC) instead of paper money. Thus, adopting an Intelligent Blockchain-based Management System (IBMSDC) as a methodology for the design of a management system that handles, monitors, and protects digital currencies and transactions. The proposed model combined distributed and centralised functionality intelligently to provide consumers with more security and privacy. Big payments had forced financial institutions to adopt a more effective, reliable and secure Real Time Gross Settlement System (RTGS).

2.2 What is Hyperledger Fabric-Composer?

Hyperledger is one of the private networks developed by Linux Foundation which allowed to host a business network of all kinds. Researchers explained a brief insight into blockchain platforms and an examination of exactly the definition of Hyperledger Fabric and designed steps of the Hyperledger Composer tool (S, et al., May 15-17,2019). Hyperledger Composer is a series of blockchain modelling software (Košt'ál, et al., 2019). It provides a platform to speed up the creation of Hyperledger Fabric apps that run on top of the Hyperledger Fabric. Successfully a study on the Composer web browser has been performed to create a quick transaction in blockchain and the use of Hyperledger Composer certainly simplifies work and takes less time compare to Hyperledger Fabric.

Adoption of a business network requires model file, JavaScript file, and access control rule file to archive as business network archive file and deployed as a Hyperledger Fabric in local or cloud distributed construction (Tanwar, et al., 2020). This JavaScript file on the business network blockchain is ideal for a private business application blockchain (Kim, et al.,

2020). Before deployment in a Hyperledger Fabric, it is compatible with any other alteration but is purely secure and once transactions made are irreversible, cannot be modified or removed.

2.3 Why using Hyperledger Fabric in Consumer Banking Application?

Scalability is a commonly discussed issue for existing blockchain technology. Pongnumkul, et al. (July 31-Aug.3, 2017) performed a performance review of Hyperledger Fabric and Ethereum through private deployment blockchain systems for a variety of transactions. Based on various transactions, the experimental results showed that Hyperledger Fabric constantly exceeds Ethereum across all measurement methods such as time, latency and throughput. Hao, et al. (June 26-30, 2018) proposed a framework for evaluating the efficiency of consensus algorithms on Ethereum and Hyperledger Fabric's private blockchains platforms. Through quantitative latency and throughput analysis, they obtained results from the performance evaluation of consensus algorithms with different transaction numbers. The Practical Byzantine Fault Tolerance (PBFT) of Hyperledger Fabric continuously exceeded the Proof of Work (PoW) of Ethereum in terms of latency and throughput at different workloads.

Kuzlu, et al. (July 14-17, 2019) examined the impact of network workload on the throughput, latency and scalability of the Hyperledger Fabric Architecture. It had been shown that up to 100,000 participants in the selected AWS EC2 instance can be supported by the implementation of the Hyperledger Fabric platform. Wang, et al. (July 2-7, 2018) introduced an end-to-end RTGS prototype based on a Hyperledger Fabric blockchain platform. The prototype supported gross settlement, gridlock settlement and interbank payment reconciliation. This project had been shown to provide more efficient and secure payment service on blockchain networks in businesses. Lee, et al. (2020) presented a time bank system based on the Hyperledger Fabric framework, which is one of the permissioned networks of the blockchain. This research designed and created the time bank system which enables the blockchain to run and record all service-related processes. Developing the blockchain time bank network benefits the time credit transaction, which plays the role of digital currency on the network. In addition, the proposed time bank also maintains a grading system allowing its members to grade each other to reflect their satisfaction with the system's results. The idea to use blockchain to implement grading and reputation had been demonstrated to increase mutual trust among networked members and to enhance the security of the system as a whole.

Table 1 below show the different blockchain platform in metric which differentiate the public blockchains and permissioned blockchains.

Table 1: Comparison between different Blockchain Platform *(Source: Chua, et al., April 2-4, 2019; Ranjan, et al., August 8-10, 2019; Dinh, et al., May 14-19, 2017)*

Blockchain Platform	Hyperledger Fabric	Ethereum	Bitcoin
Network	Permissioned	Public	Public
Smart Contract	Yes (chaincode)	Yes (solidity)	No
Smart Contract Execution	Dockers	EVM	No
Consensus Protocol	PBFT	PoW (ETHASH), PoS	PoW (SHA-256)
Language	Golang, Java	Golang, Python	C++
Scalability	High	Low	Low
Cryptocurrency-based	No	Yes	Yes
Permissioned	Yes	No	No
Immutable	Yes	Yes	Yes
Modularity	Yes	No	No
Access Control	Yes	Yes	No

3 Research Methodology

3.1 Overview for Multon Bank Dapp

Proposed bank decentralised application (Dapp) known as Multon is used in this project to test the scalability and security of this system with the integration of Hyperledger Fabric through the smart contract called chaincode. Operating with Hyperledger Composer creates the blockchain sample network displayed in this project. The Hyperledger Composer is built and then exported into three separate files as an archive file and ready for deployment. The environment contains tutorials, documentation, samples and a forum for community support to promote success in its use. We can also test and expose using Hyperledger Composer API: Applications call for communicating API transactions with the network of the business. Web browser is tested in composer playground(http:/composerplayground.mybluemix.net/) in our PC. Finally, Angular is used to build the front-end from the REST API network.

3.2 Oracle Virtual Box

In the development environment, one virtual machine (VM) is setup in the Oracle Virtual Box version 6.1.8. The VM has 1 CPU Intel PRO/1000 MT Desktop (NAT) and 2GB of RAM. All requirements and dependencies are installed together with the Hyperledger Fabric blockchain framework in VM. Prototype implementation is chosen to launch the Hyperledger Fabric

framework and accelerate the development using the Hyperledger Composer tool as an open development platform and system that will facilitate the development of Hyperledger Fabric blockchain applications. These elements are used on a single host and execute a Docker runtime environment as containers to run the chaincode. After the blockchain network is developed, the Multon business network is built and implemented into the blockchain network.

3.3 Smart Contract for P2P Bank Dapp

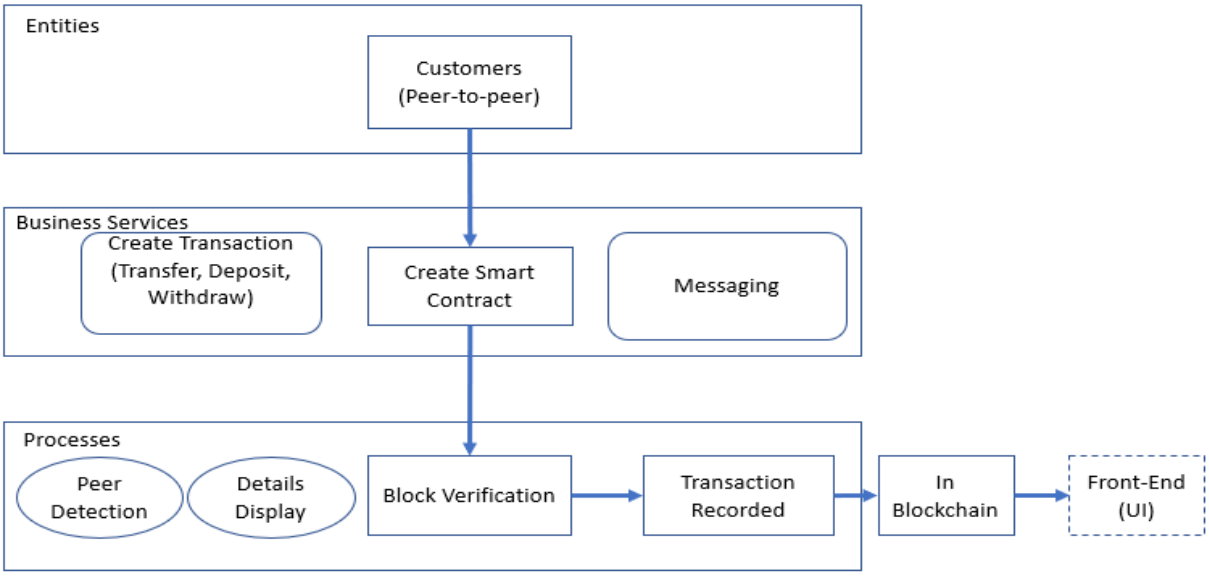

Figure 2: Proposed Multon Bank Dapp

Figure 2 show the component of the proposed Multon Bank Dapp. The front-end decentralised application (Dapp) known as Multon can interact with smart contract with interface develop using Angular. REST API is used as medium to connect frontend with the implementation of Node JS. The analysis in this report develop the bank contract for P2P and the following subsections describe the sharing platform. P2P Bank network is regarded as business network which account as an asset. The business network consists of entities working together to accomplish a certain mission. There should be certain rules regulating their dealings with various people, including how and when transaction should be carried out, who should be involved and how payment should be handled. Through blockchain technology, rules governing a transaction are referred to as smart contract, a digital arrangement between the distributed parties. The design steps are as follows. (1) Register for customers (2) Create the smart contract in blockchain network (3) Managing access control prior block verification (4) Store transaction details in chaincode (5) Integrate the smart contract and Dapp in front-end.

3.3.1 The role of entity in the smart contract network

For the creation of smart contracts, the business service layer is necessary and involves creating transactions and sending messages. The internal process layer includes the network detection by new network members, details display, block verification and construction of the latest

transaction block to be applied to the blockchain network. Below are the terms used when building the smart contract.

Define participants: There are three functions for customers to register their basic platform information, including the ‘customerId’, ‘firstname’, and ‘lastname’.

Define assets: Data owned by customers like account Id, balance, message, and status.

Define transactions: A unique account Id is used to identify different asset in order to track back the customer easily. Users can submit the four types of transaction such as transfer, deposit, withdraw, and message.

Access Control: Limit the access control between owner and customers over the business network. For example, all customers can read resources and submit the transaction while owner has the full access to customer resources and assets. However, once the transaction was submitted, the result cannot be deleted or modified.

Chaincode: Complying access control policies for viewing or modifying key value pairs or other information in the database.

Blockchain: Blockchain is used as an application controller to monitor access control policies and maintain the integrity of the customer data.

Front-end: A decentralised web application is implemented with the REST API which act as a support system.

3.4 Performance Evaluation

The evaluation of system performance in relation to scalability and security in data processing is another significant consideration.

3.4.1 Security

Chaincode is strictly defined and cannot be modified. When the chaincode is installed, the system will provide the computed hash of the chaincode for the name space of the chaincode. Thus, if another pair has different binary code and tries to use it, it won’t be executed by the pair. Furthermore, security of Multon Dapp can also be proved by Unique Id and timestamp for each transaction when customers transfer, deposit, withdraw, and message to other peer.

3.4.2 Scalability

Five experiments are conducted in the result and evaluation section using Apache JMeter[4] on the REST API, where thousands of users accessed the framework for transaction data already in the blockchain database. Before the tests, the transaction data is loaded manually into the blockchain. Apache JMeter is setup to send GET requests to REST Server at http://localhost:3000/explorer/#/AccountTransfer. History of Account transfer is collected as it is the most common activity carried out by participants of the blockchain network.

4 Design Specification

In figure 3, Multon Dapp, blockchain-based decentralised website framework on the Hyperledger Fabric-Composer Network is introduced and used to test the access control mechanism.

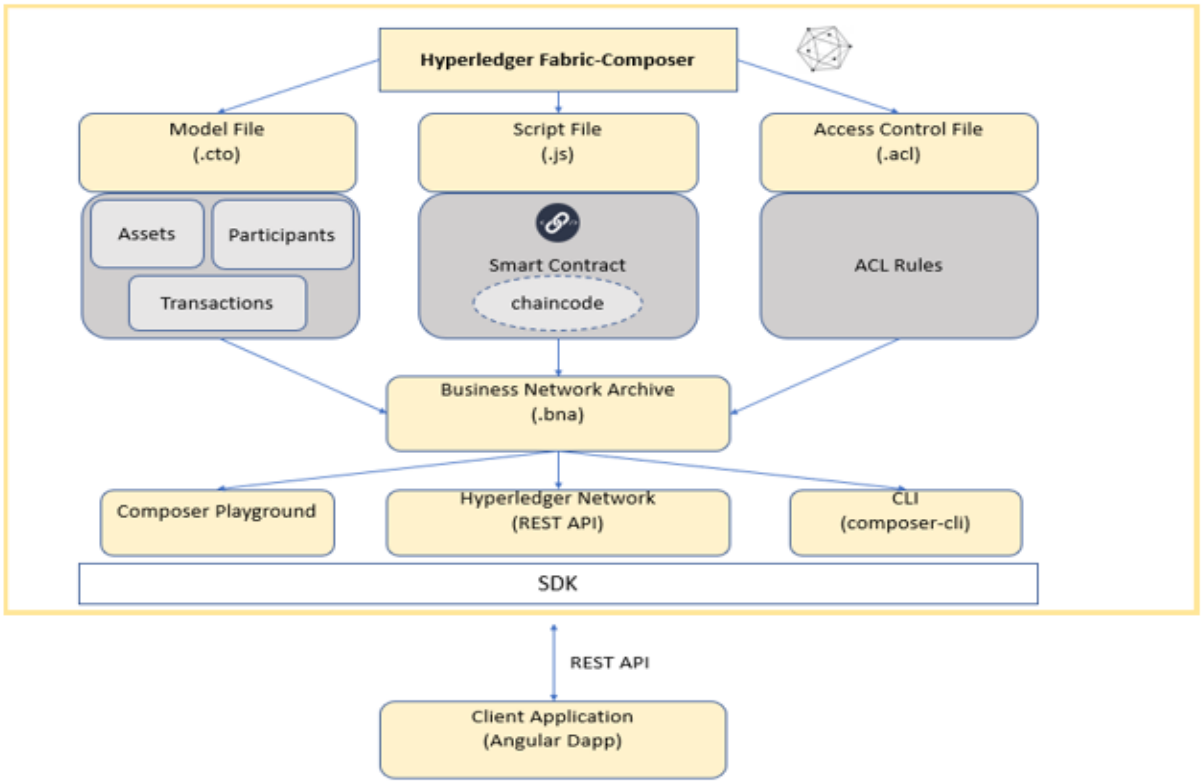

Figure 3: Client application back-end framework

4.1 Business network archive file structure

There are three files for the development of an application blockchain.

• Assets, participants and transactions in Model File(.cto)

[4] https://jmeter.apache.org/

• Transaction logics in JavaScript are written and stored in Script File (.js)

• The Rules for access system are stated in Access Control File (.acl)

All these files are then combined into business network archive file (.bna) and ready for deployment in a Hyperledger Fabric environment.

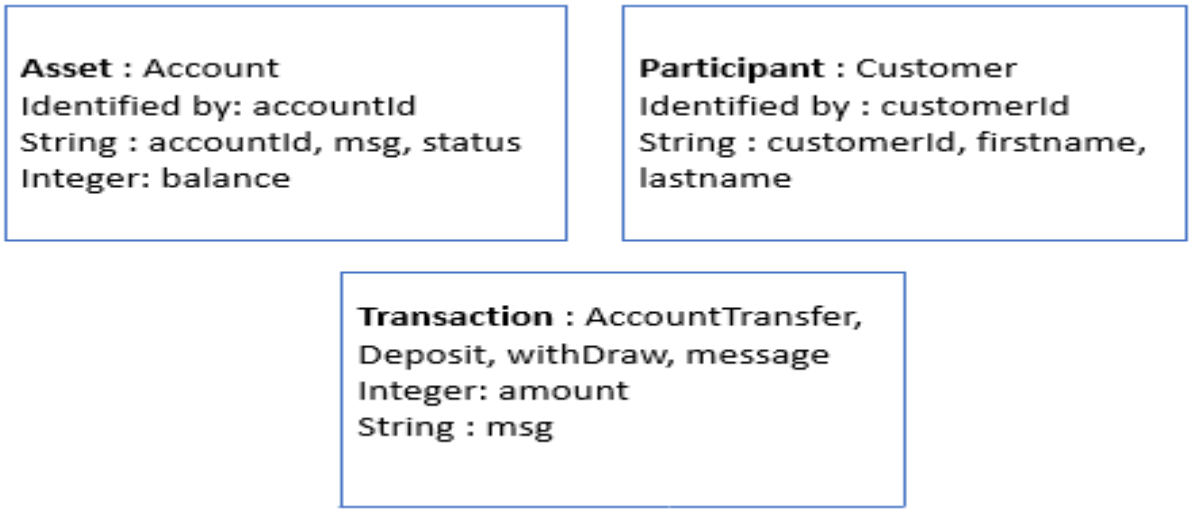

Figure 4: Cto Model

In this model, the asset is account, participants are customers, and transaction is transfer, deposit, or withdraw money for various customers require for the project as shown in figure 4. Initial total amount in which the project participants are free to withdraw, transfer or deposit amounts are made. Each participant has an asset (account). When 'AccountTransaction' is submitted the transaction processor function is called. The function is passed to the participant and the amount of money to be transferred, withdrawn or deposited. Not only that, customers can communicate each other using the message function. If all the constraints are met by the transaction parameters, the transaction will be reported in the registry. This transaction entry also includes a transaction ID with a time stamp upon submission of the transaction. When a transaction is submitted, the amount of money specifies will be withdrawn, transferred or deposited from the total amount. After submission, the registry of assets is updated so that the data is consistent.

5 Implementation

5.1 Hyperledger Composer Playground

Hyperledger Composer Playground offers a user interface to plan, execute and check the business system. The mains of the playground allow customers to deal with business system with numerous blockchain systems of the organisation. It is displayed in the server (http://localhost:8080/login) as illustrated in figure 5. The playground homepage, where you can show each of the business network cards you have for your use.

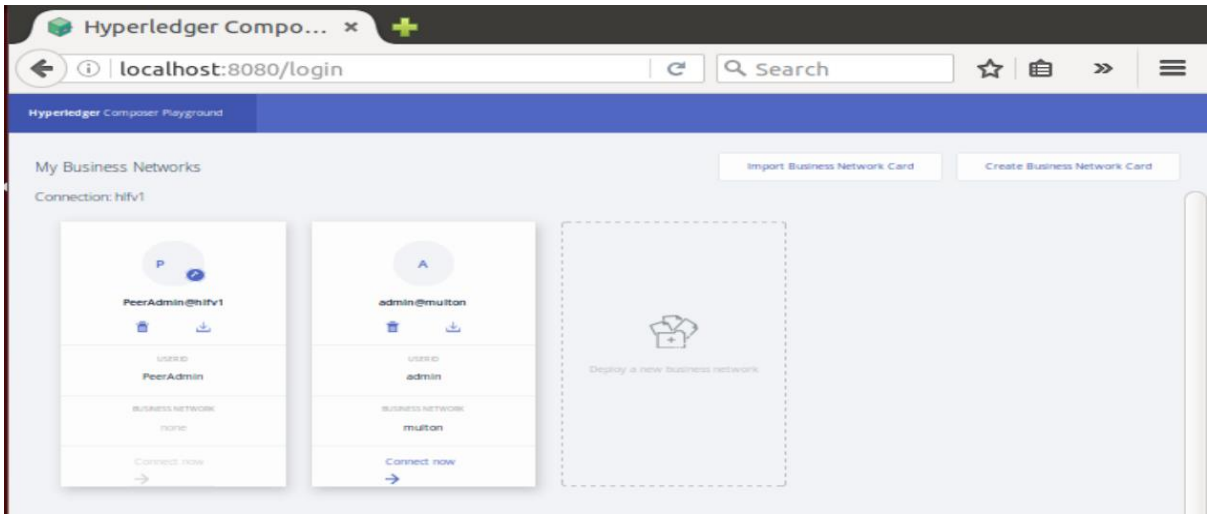

Figure 5: Composer Playground Web Interface

5.1.1 Define Tab

In the define tab, the content of the program can be altered and you can render assets describe in the define section. You can also add or remove the file you like. The export button allows you to save your existing business network into your PC as .bna record. The define tab has the required model file as shown in figure 6.

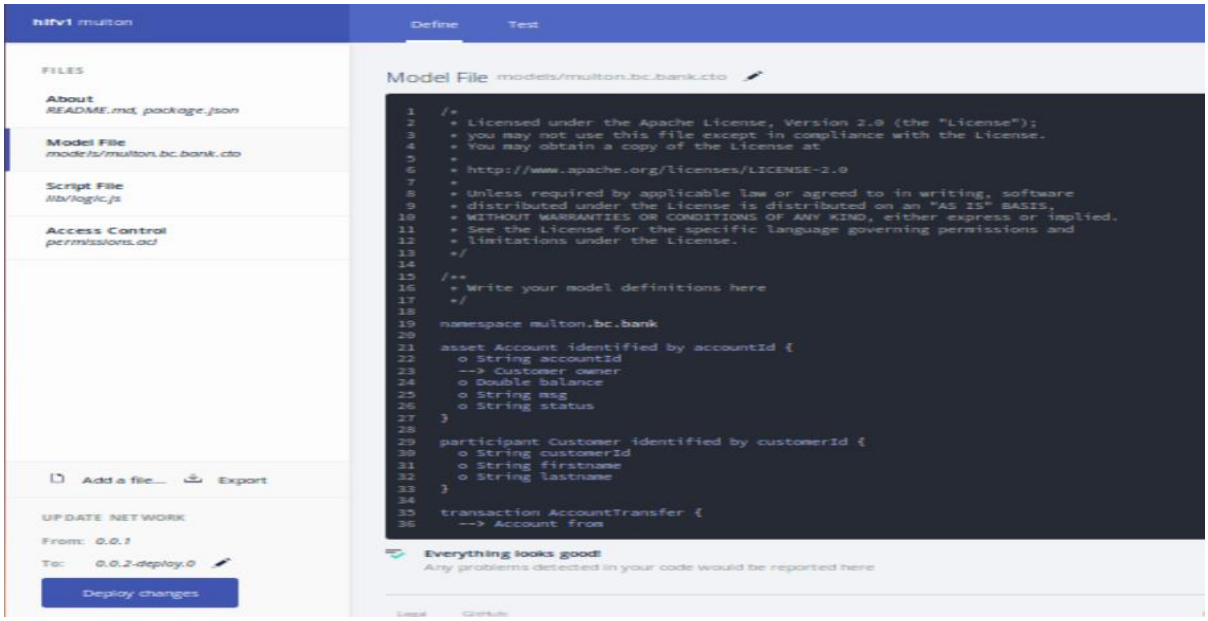

Figure 6: Model File Interface

Here, a simple trade transaction is made. In order to make things simple, assets include attributes such as account Id, balance, messages, and status of the customer for each asset. Participants are identified and the transaction identity is created. Participants will have characteristics such as customer Id, first and last names. Participants are important because only participants are involved in transactions. After all the logics of the transaction are stated, the script file needs to be created as shown in figure 7.

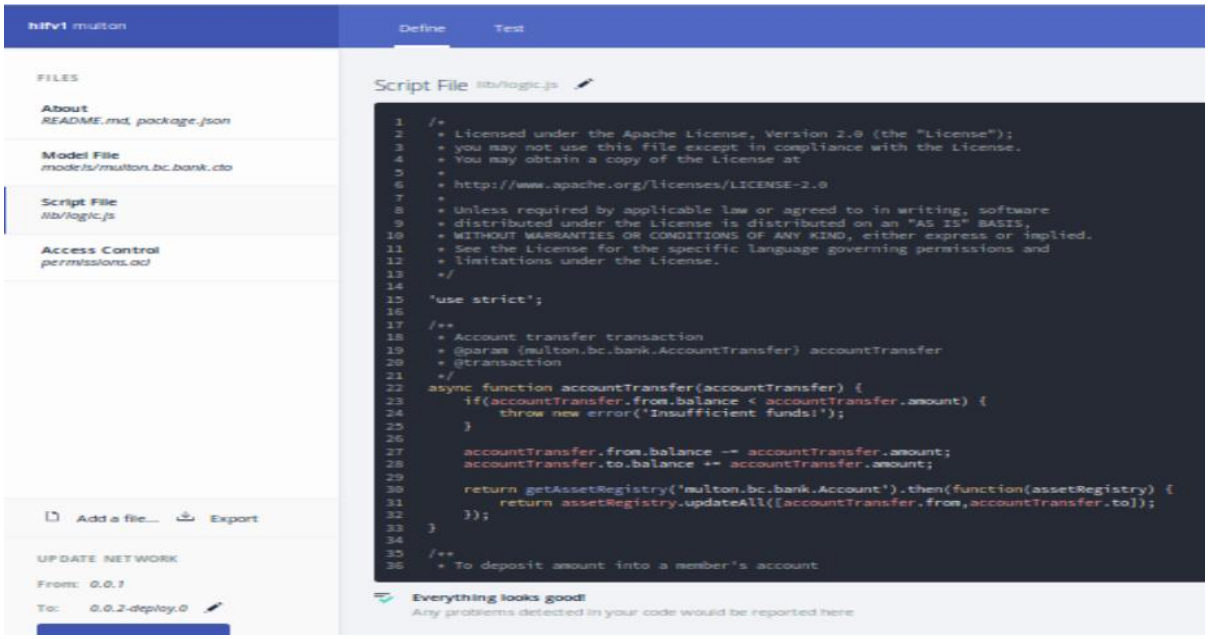

Figure 7: Script File Interface

Transaction involves the attributes like account transfer, deposit, withdraw, and message. In this script file, the Async function is defined to update the new customer immediately after the transaction is processed successfully. After that, the Access Control is defined in figure 8.

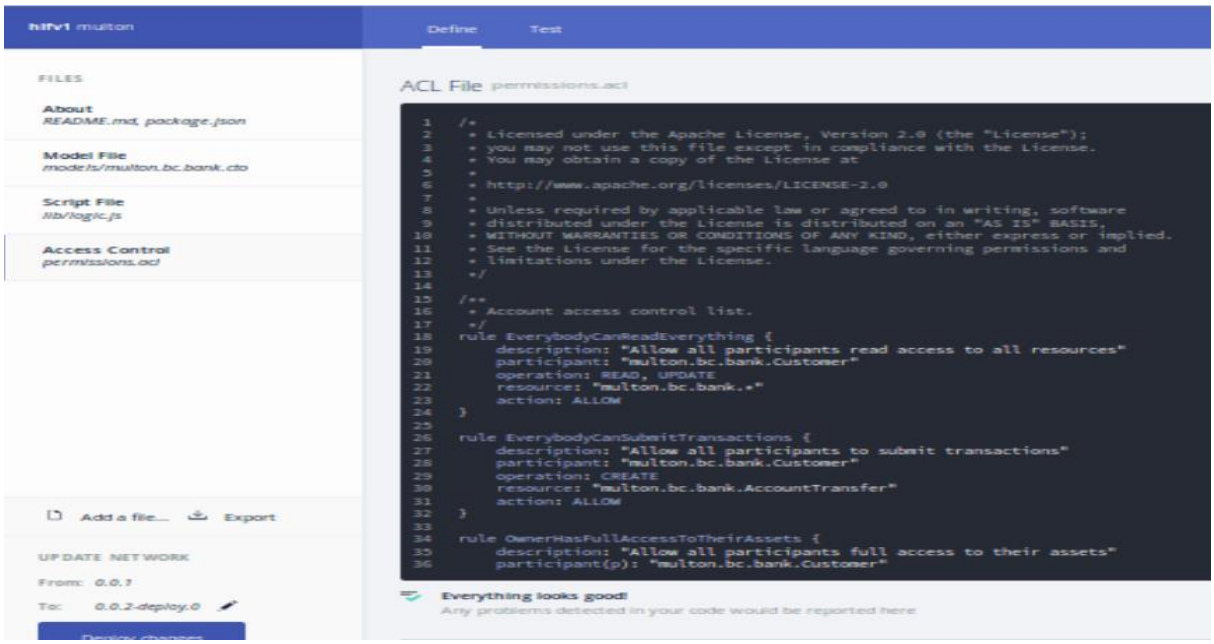

Figure 8: Access Control Interface

The access control file (.acl) for the business network characterises business network management and business access control for the specified users. The network access control uses the namespace multon.bc.bank to store all business network assets. Before transaction, new customers and new account are added to allow for transaction in the test tab.

5.1.2 Test Tab

In the test tab, you can check the usability of the business network card by using the customer and account types that have been identified on the define tab.

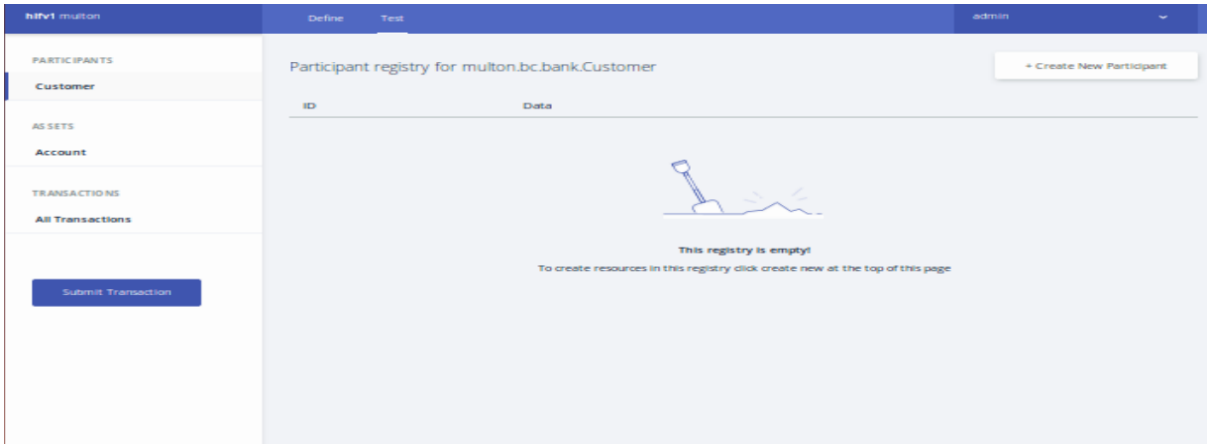

Figure 9: Customer Interface

The detail of customer, account and registry are reported to the left of the test tab as shown in figure 9. When you click on customer, account or all transactions, a registry will open that displays every complex occasion of that kind. For example, when you click customer, it will display a registry displaying the list of all customers. If you open a test tab for the first time, the registries will soon be found empty after selection. In each registry, each customer can make transaction through deposit, withdraw, transfer, or message. You can see in the 'All transactions Registry', a record of any transaction that has taken place, including some transactions that have been performed. For example, you can transfer money from one customer to another and you can see the changes of transaction record by clicking 'All Transactions'.

When you select the new participant option at the right top corner, you can build new participants that can register new customers. Here, a new customer is created as shown in figure 10.

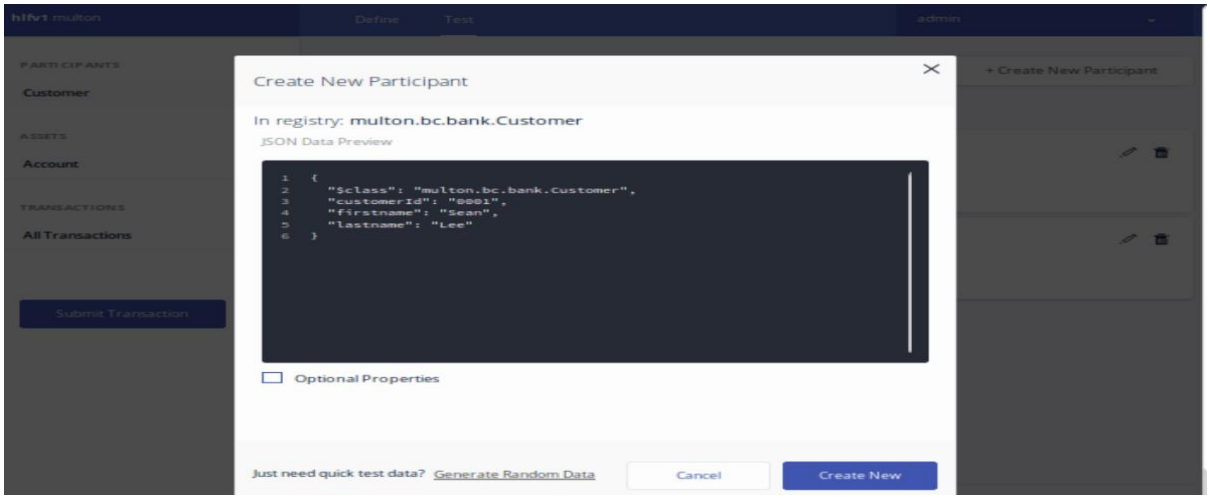

Figure 10: Create New Participant

The customer ID or ID creates by the system as a random number and then the customer's first and last name is defined. Once the customer ID has been created, it is shown in the figure below.

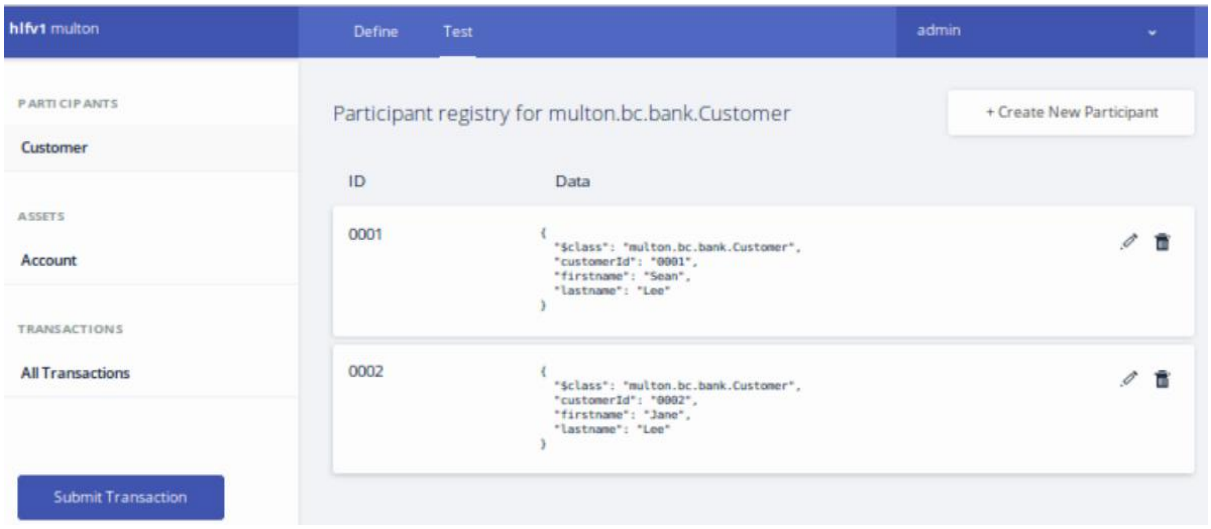

Figure 11: Customer Registry

The establishment of two customers takes place where the customer's identification is given as 0001 and 0002 respectively. The account details are added in the asset registry as illustrated in figure 12. The details like the account Id, the owner of the asset, balance of money, status of transaction, and message are defined accordingly.

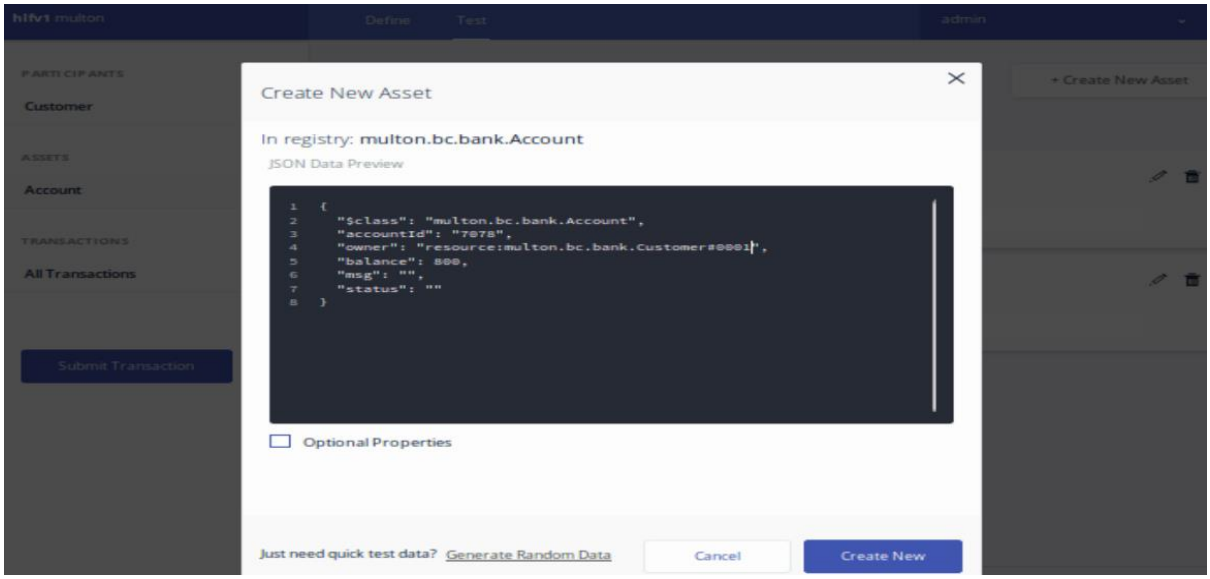

Figure 12: Create new account

The assets will be registered together with the ID, which is unique to all assets in figure 13 after the details of account is stated.

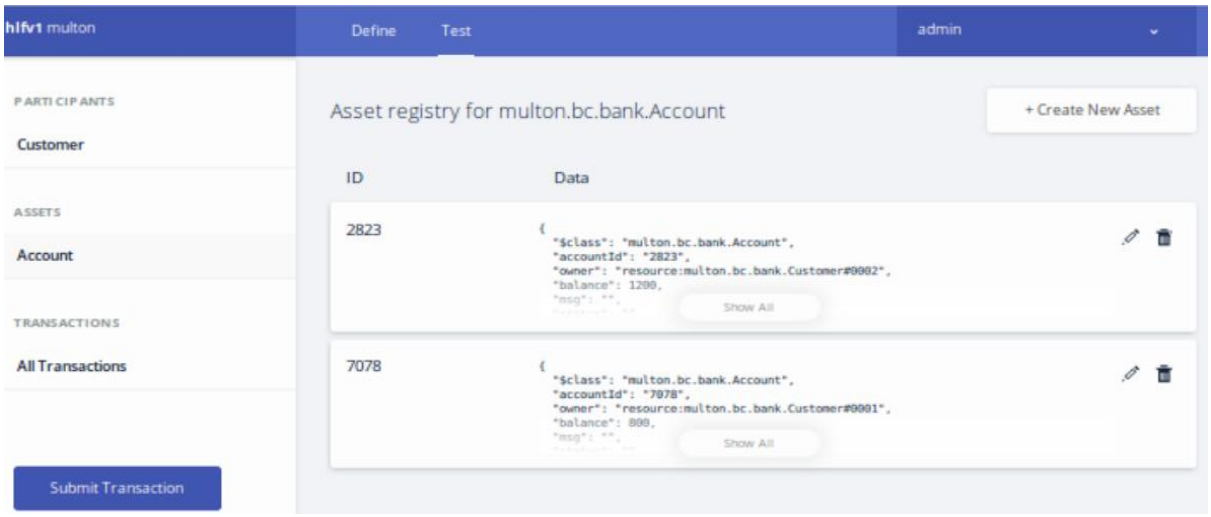

Figure 13: Account Registry

As the assets are done, the transaction is the next step. To make a transaction, first press the 'Submit Transaction' button in the left-hand corner. Then account transfer details are opened as shown in figure 14.

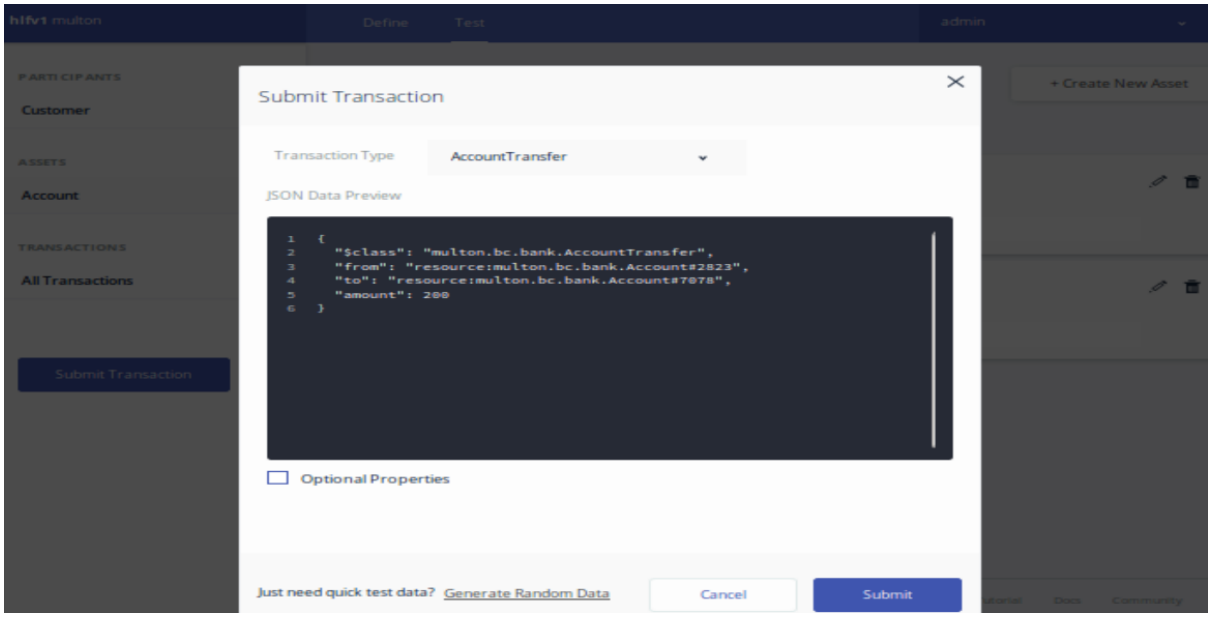

Figure 14: Submit Transaction

The transaction takes place between two customers, in which the customer transfers his/her funds to another customer. There are two important items to consider. The first thing is that the account ID should be associated with the customer ID we seek to exchange. Secondly, the customer we want to deal with the asset. The account we are attempting to transfer in here is Account ID 2823, which should be labelled as account: "resource: multon.bc.bank.Account#2823" to the new customer, which is the account ID 7078. The term 'new customer' should be defined as "resource: multon.bc.bank.Account#7078". Amount transfer is 200 euro. After clicking the transaction submitted, the transaction between customer 0001 and customer 0002 would be successful.

After the transaction has ended, we can see that the amount of account with ID 7078 had changed from 800 euro to 1000 euro previously owned by Customer #0002 would has moved to the customer's ownership from customer #0002 to customer #0001 as shown in the figure as below.

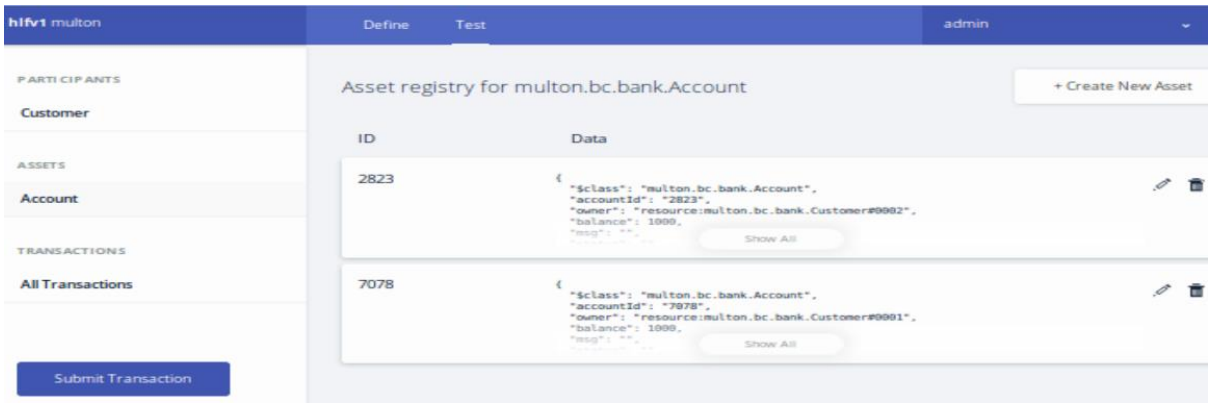

Figure 15: Interface showing the asset update after a valid transaction

Transactions can be seen at the transaction registry from the timestamp format of the transaction to the type of entry and participants. 'Date' is a blockchain record time showing when the transaction is performed. 'Entry Type' represents the type of transaction and 'Participant' represent the current participant who submits the transaction. The dashboard also includes 'view record' which shows the transaction details. All transactions appear in figure 16.

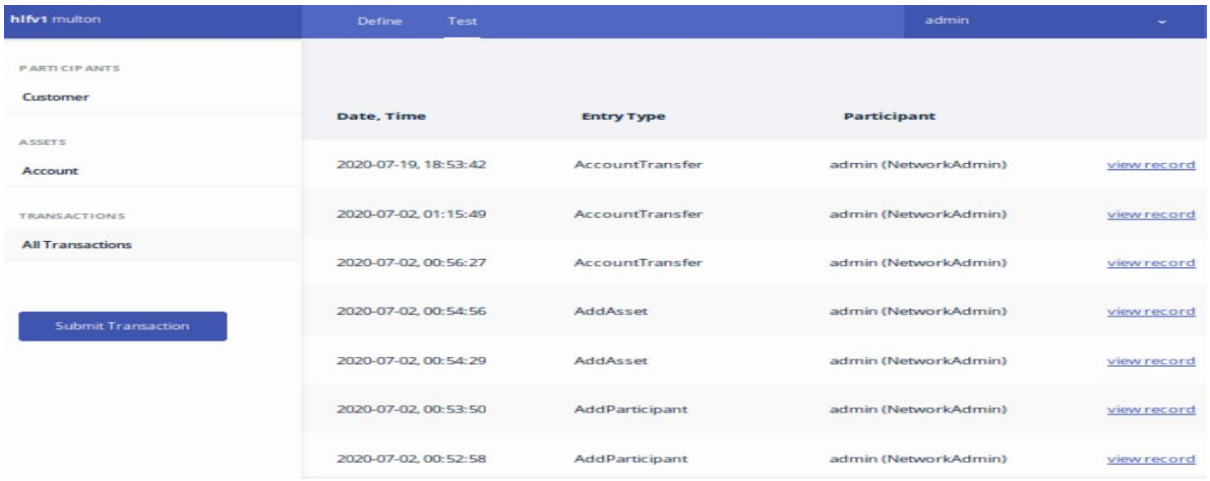

Figure 16: Transaction history in 'All Transaction' interface

5.2 REST API

The final step is to integrate the blockchain network with the bank application to facilitate bank transactions between the customers. One of Hyperledger Composer's main advantages is that it can provide a ready-to-use REST server with an exposed business network for each asset, participant and transaction network for application development. The interdependent relationship between Composer Playground and REST API allows users to update their data

like blockchain transactions in Multon front-end. The developer accesses all necessary integration data through the REST server API as shown in figure 17. For example, the REST server API displays the account, customer, account transfer, deposit, and other variables which you can access via the HTTP GET and POST requests. REST server was built and accessible on http://localhost:3000/explorer/#/.

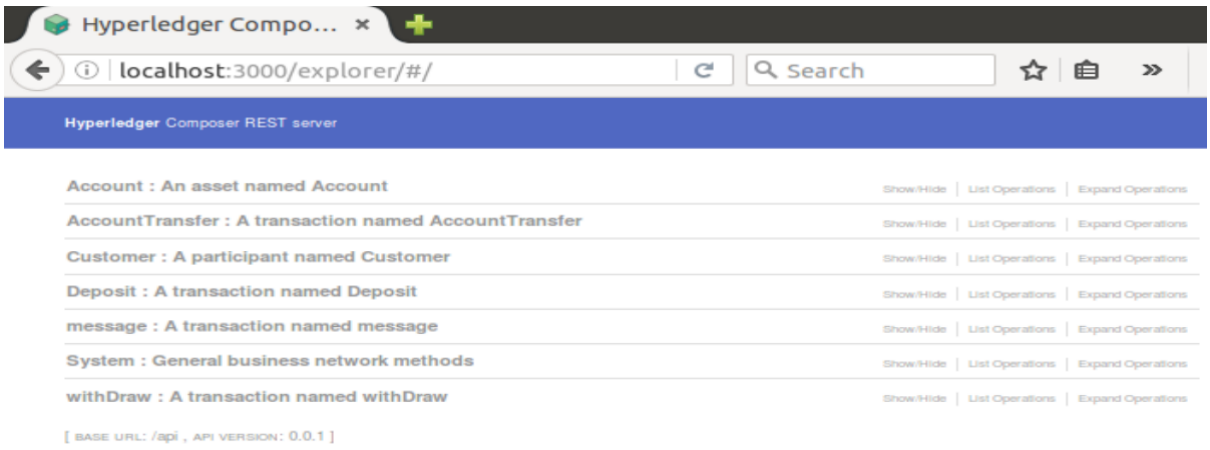

Figure 17: Hyperledger Composer REST API

5.3 Angular

The blockchain client front-end is developed with a JavaScript SDK to handle and communicate with the deployed business network with a difference of APIs. Yeoman (generic scaffolding tool) is used to generate the Angular skeleton App that starts in the blockchain application and then the Angular app will connect to the REST API that 'Composer Rest Server' command is used to create the front end. As it is in the testing phrase, the customers do not require to authenticate themselves to the REST server. The Multon Dapp is accessible on http://localhost:4200/. After clicking the invoke button, 'AccountTransfer' is readied to make the transaction in the transactions interface in figure 18.

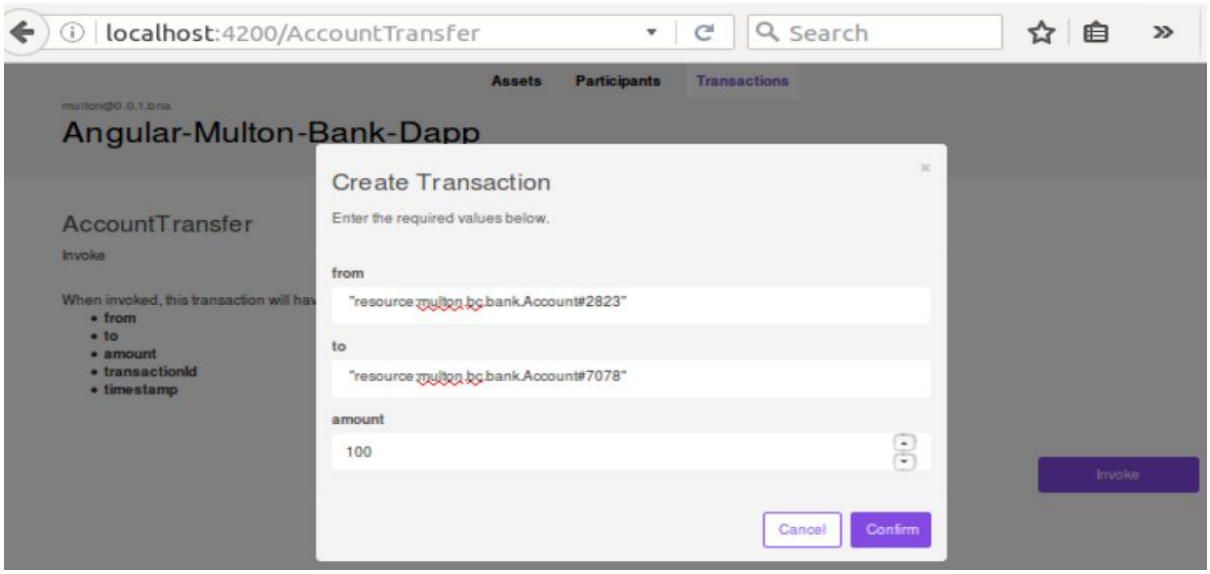

Figure 18: Account transfer in Transactions Interface

6 Result and Evaluation

6.1 Security

Hyperledger Fabric has the modular architecture that allow consensus to be plug-and-play. The important metrics about Hyperledger Fabric is it allows the transaction to occur in anonymous through the private channel. Only certain participants are allowed to access the smart contract through access control in Hyperledger Fabric. Unique identity of the participant is shown in figure 19.

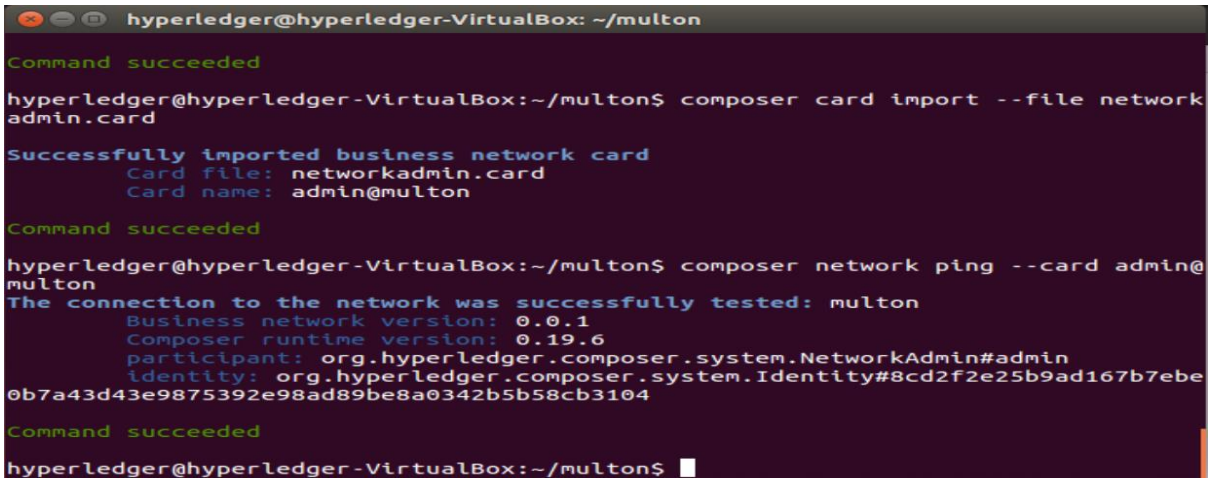

Figure 19: Unique Identity for Hyperledger Composer

The record contains technical details of transactions such as the transaction ID and the timestamp. Figure 20 show the transaction activities in the Composer by creating a hash ID for each transaction which shows the immutability of blockchain features. The important thing to note is the transaction details store cannot be edited or deleted. This show the transparency of blockchain technologies. For example, the transaction ID and timestamp are unique for each transaction.

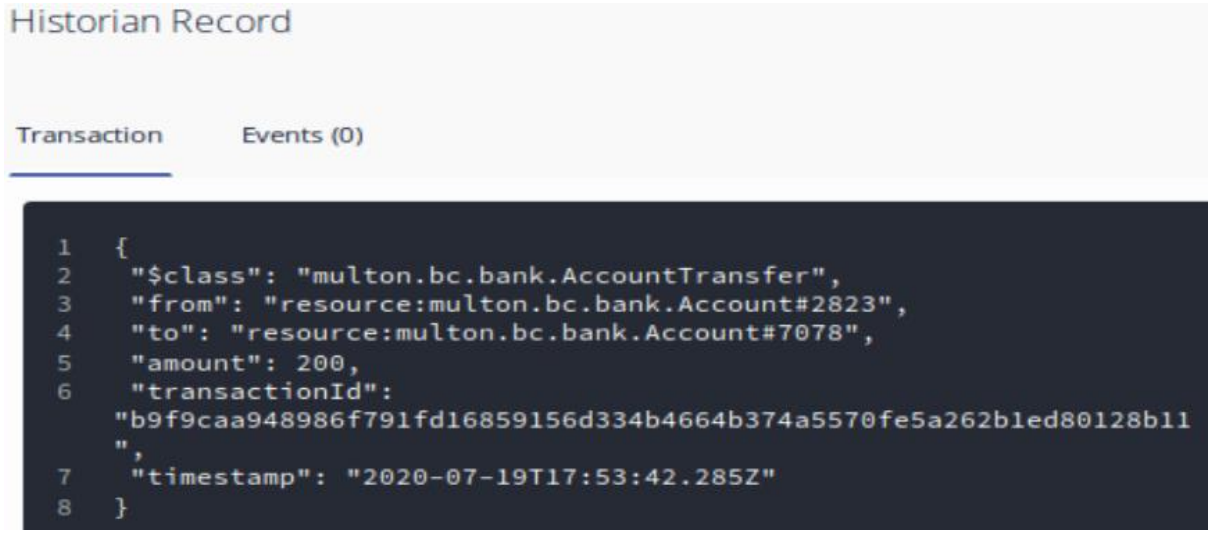

Figure 20: 'AccountTransfer' Transaction in Composer Playground.

6.2 Scalability

6.2.1 Experiment 1 (5 users)

In experiment 1, Apache JMeter testing tool is used to evaluate the REST API for Multon Dapp currently under development in a real-life internet scenario. System performance is evaluated using Apache JMeter to evaluate average response time. The experiments are done on 1 CPU with 12 GB RAM. HTTPs requests are generated with Apache JMeter where the load is transmitted with 5 user threads over a 1-second round-up period. This mean that 5 users are connected to server simultaneously per second. Table 2 shows the result of a typical test run with the simulated users of 5. The application's average response time is 188.60ms without any reported dropouts.

Table 2: JMeter and Performance Evaluation of 5 users

timestamp	elapsed	thread name	success	URL
1.59E+12	188	Thread Group 1-2	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	188	Thread Group 1-4	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	189	Thread Group 1-5	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	187	Thread Group 1-3	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	191	Thread Group 1-1	TRUE	http://localhost:3000/explorer/#/AccountTransfer
	188.60			

6.2.2 Experiment 2 (50 users)

In experiment 2, HTTPs requests are generated with Apache JMeter where the load is transmitted with 50 user threads over a 10-second round-up period. This mean that 5 users are connected to server simultaneously per second. Table 3 shows the result of a typical test run with the simulated users of 50. The application's average response time is 43.04ms without any reported dropouts.

Table 3: JMeter and Performance Evaluation of 50 users

timestamp	elapsed	thread name	success	URL
1.59E+12	248	Thread Group 1-3	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	8	Thread Group 1-6	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	253	Thread Group 1-2	TRUE	http://localhost:3000/explorer/#/AccountTransfer
:	:	:	:	:
:	:	:	:	:
1.59E+12	104	Thread Group 1-49	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	8	Thread Group 1-50	TRUE	http://localhost:3000/explorer/#/AccountTransfer
	43.04			

6.2.3 Experiment 3 (500 users)

In experiment 3, HTTPs requests are generated with Apache JMeter where the load is transmitted with 500 user threads over a 100-second round-up period. This mean that 5 users

are connected to server simultaneously per second. Table 4 shows the result of a typical test run with the simulated users of 500. The application's average response time is 9.93ms without any reported dropouts.

Table 4: JMeter and Performance Evaluation of 500 users

timestamp	elapsed	thread name	success	URL
1.59E+12	69	Thread Group 1-8	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	233	Thread Group 1-1	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	258	Thread Group 1-3	TRUE	http://localhost:3000/explorer/#/AccountTransfer
:	:	:	:	:
:	:	:	:	:
1.59E+12	3	Thread Group 1-499	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	18	Thread Group 1-500	TRUE	http://localhost:3000/explorer/#/AccountTransfer
	9.93			

6.2.4 Experiment 4 (5000 users)

In experiment 4, HTTPs requests are generated with Apache JMeter where the load is transmitted with 5000 user threads over a 1000-second round-up period. This mean that 5 users are connected to server simultaneously per second. Table 5 shows the result of a typical test run with the simulated users of 5000. The application's average response time is 9.57ms without any reported dropouts.

Table 5: JMeter and Performance Evaluation of 5000 users

timestamp	elapsed	thread name	success	URL
1.60E+12	271	Thread Group 1-4	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.60E+12	276	Thread Group 1-5	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.60E+12	26	Thread Group 1-9	TRUE	http://localhost:3000/explorer/#/AccountTransfer
:	:	:	:	:
:	:	:	:	:
1.60E+12	1	Thread Group 1-4999	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.60E+12	2	Thread Group 1-5000	TRUE	http://localhost:3000/explorer/#/AccountTransfer
	9.57			

6.2.5 Experiment 5 (6000 users)

In experiment 5, HTTPs requests are generated with Apache JMeter where the load is transmitted with 6000 user threads over a 1200-second round-up period. This mean that 5 users are connected to server simultaneously per second. Table 6 shows the result of a typical test run with the simulated users of 6000. The application's average response time is 7.61ms without any reported dropouts.

Table 6: JMeter and Performance Evaluation of 6000 users

timestamp	elapsed	thread name	success	URL
1.59E+12	985	Thread Group 1-13	TRUE	http://localhost:3000/explorer/#/AccountTransfer

1.59E+12	76	Thread Group 1-23	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	985	Thread Group 1-11	TRUE	http://localhost:3000/explorer/#/AccountTransfer
:	:	:	:	:
:	:	:	:	:
1.59E+12	1	Thread Group 1-5999	TRUE	http://localhost:3000/explorer/#/AccountTransfer
1.59E+12	2	Thread Group 1-6000	TRUE	http://localhost:3000/explorer/#/AccountTransfer
	7.61			

6.3 Discussion

After spending a couple of weeks, the private blockchain is chosen which prevents unauthorised sharing in order to limit the management and access of network participants. Two other reasons justify our use of a private network blockchain: (a) promote security and (b) increase scalability in data transactions. This study focusses only on private blockchains rather than public blockchains due to data security and scalability issues, as well as due to Multon's specific financial sector.

Before starting the development of the blockchain application itself, the blockchain network had been set up. Development of Hyperledger fabric network is not easy as the architecture aspects and components of the blockchain network had to be learned. The deployment and network start-up are the most common issues encountered during the environment setup. Although the set-up of the network is difficult, development of blockchain applications using the Hyperledger Composer platform help speed up development of activities. It allows the Multon Dapp to deploy rapidly. Then, the research question has been discussed to accomplish the goal for the security and scalability in Multon Dapp.

Private blockchains provide a high degree of data security than public blockchains as only approved users have access to the network. Some data may also need to be shared only with specific participants within the blockchain consortium network to prevent information leakages to competitors and to protect customer data. It works on the basis of a decentralised network that has no single point of failure. The Multon protects the bank information from security breaches, such as unauthorised access. In cases like these, companies can create separate private channels in the larger blockchain network to deal privately with a subset of the nodes. This is the conceptual approach that allows Hyperledger Fabric to create sub-sets of members via private channels for its blockchains.

Decentralised web application, Multon is scalable to handle an increased amount of customer. By using permissioned blockchain, we can accommodate larger quantities of participants and transactions. Permissioned blockchain in Multon is able handle more transactions with longer chains and lower response time in term of scalability. The results are collected with Apache JMeter in virtual server show that the average response time decreased when number of users increase when 5 users are connected simultaneously to the REST server per second in figure 21. In order to avoid the higher response time when number of users increase, each experiment is tested in different virtual server. As the number of users increase, it will consume much time in doing the experiment. Author may encounter the breakdown of the server when running with different experiments. Thus, different virtual server has cloned to prevent the undesired condition.

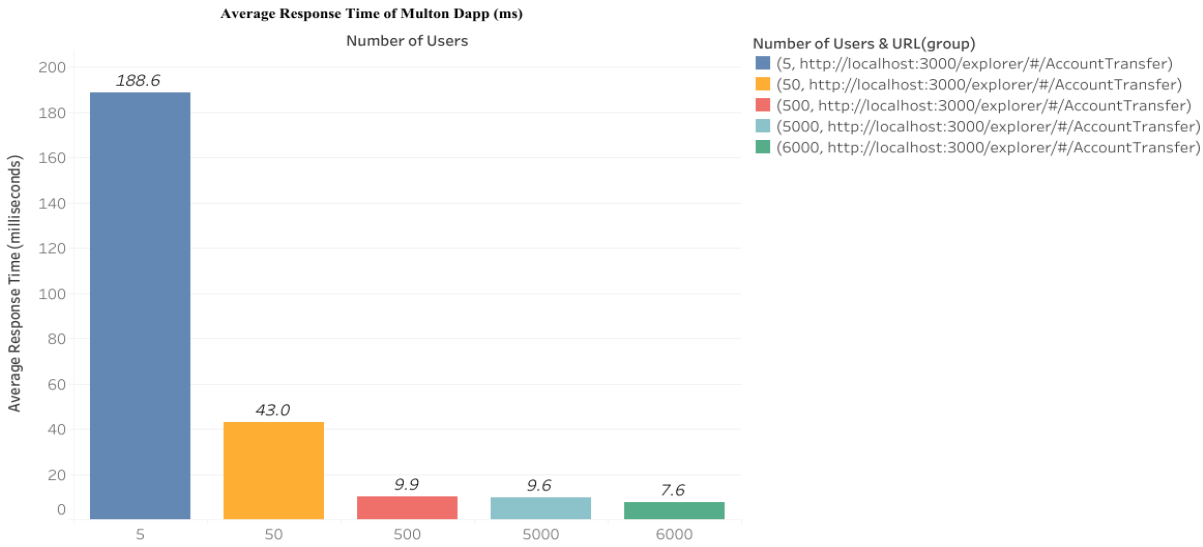

Figure 21: Average response time of Multon Dapp[5]

7 Conclusion and Future Work

The use of permissioned blockchain in financial sector plays an important role in the traditional financial market. Using Hyperledger as a private permissioned blockchain introduces scalability and security compared to public blockchains solutions. The scope of this project is in finding ways to build the Hyperledger Fabric using the Hyperledger Composer tool in creating blockchain business networks with the utmost security. The combination of model files, script files and acl files can archive as the .bna file and ready to deploy in Hyperledger fabric environment.

In this project, the system architecture is proposed to prove the existence of security and scalability in permissioned blockchain. The implementation of Multon Dapp through the chaincode is demonstrated. The system that run without intermediary and single point of failure in the system through the separate private channels has been discussed. When making transaction between peers, security is shown through the unique Id and timestamp as transactions once made are irreversible. Performance evaluation of the proposed system in scalability is achieved using the JMeter for different experiments. The results indicate the average of response time decrease when users increase. These prove that Multon can accommodate many users and handle huge amount of transactions.

In future work, more blockchain consensus algorithms like Proof of Stake (PoS) are intended to evaluate and carry their performance to improve our evaluation methodology. Exploring the performance between private and public blockchains are interesting when different execution of code snippet between two platform and how they affect the final performances. Such a blockchain banking system will end corruption and identity fraud in current banking systems because all banking transactions are registered, protected and recorded in a blockchain. Tracing of any transactions are monitored regardless of the time as the data stored in the blockchain is time-stamped and irreversible. In the evaluation part, Hyperledger

[5]https://public.tableau.com/views/Multon/Sheet1?:language=en&:display_count=y&publish=yes&:toolbar=n&:origin=viz_share_link

Caliper will be used to replace the JMeter with acceptable performance metrics to compare the difference performance between them.

References

Abdulkader, O., Bamhdi, A. M., Thayananthan, V. & Elbouraey, F., July 30-31, 2019. *IBMSDC: Intelligent Blockchain based Management System for protecting Digital Currencies Transactions.* London, United Kingdom, 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4).

Ahmed, K. B. & Kumar, D., January 10-11, 2019. *Blockchain use Cases in Financial Services for Improving Security.* Coimbatore, India, 2019 Third International Conference on Inventive Systems and Control (ICISC).

Chang, V. et al., 2020. How Blockchain can impact financial services – The overview, challenges and recommendations from expert interviewees. *Technological Forecasting and Social Change,* September, Volume 158, pp. 1-12.

Chua, P. H. T., Li, Y. & He, W., April 2-4, 2019. *Adopting Hyperledger Fabric Blockchain for EPCglobal Network.* Phoenix, AZ, USA, 2019 IEEE International Conference on RFID (RFID).

Dinh, T. T. A. et al., May 14-19, 2017. *BLOCKBENCH: A framework for analyzing private blockchains.* Hilton, Chicago, United States, 2017 ACM SIGMOD International Conference on Management of Data, SIGMOD 2017.

Ermyas Abebe, D. B. C. G. Y. H. D. K. P. N. P. V. R. C. V., December 9-13, 2019 . *Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer (industry track).* Davis; United States, 20th International Middleware Conference Industrial Track, Middleware Industry 2019 - Part of Middleware 2019.

Hao, Y. et al., June 26-30, 2018. *Performance Analysis of Consensus Algorithm in Private Blockchain.* Changshu, China , 2018 IEEE Intelligent Vehicles Symposium (IV).

Kim, S., Kim, J. & Kim, D., 2020. Implementation of a Blood Cold Chain System Using Blockchain Technology. *Applied Sciences (Switzerland),* 1 May, 10(9), pp. 1-15.

Košt'ál, K. et al., 2019. Management and monitoring of IoT devices using blockchain. *Sensors (Switzerland),* 2 February, 19(4), pp. 1-12.

Kuzlu, M., Pipattanasomporn, M., Gurses, L. & Rahman, S., July 14-17, 2019. *Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability.* Atlanta, GA, USA, 2019 IEEE International Conference on Blockchain (Blockchain).

Lee, Y.-T., Lin, J.-J., Hsu, J. Y.-J. & Wu, J.-L., 2020. A time bank system design on the basis of hyperledger fabric blockchain. *Future Internet,* 1 May , 12(5), pp. 1-16.

Pongnumkul, S., Siripanpornchana, C. & Thajchayapong, S., 31 July-3 Aug. , 2017 . *Performance Analysis of Private Blockchain Platforms in Varying Workloads.* Vancouver, BC, Canada, 2017 26th International Conference on Computer Communication and Networks (ICCCN).

Popova, N. A. & Butakova, N. G., January 28-31, 2019. *Research of a Possibility of Using Blockchain Technology without Tokens to Protect Banking Transactions.* Saint Petersburg and Moscow, Russia, 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).

Ranjan, S. et al., August 8-10, 2019. *Network System Design using Hyperledger Fabric: Permissioned Blockchain Framework.* Noida, India, 2019 Twelfth International Conference on Contemporary Computing (IC3).

Sakho, S., Jianbiao, Z., Essaf, F. & Badiss, K., December 6-9, 2019. *Improving Banking Transactions Using Blockchain Technology.* Chengdu, China, 2019 IEEE 5th International Conference on Computer and Communications (ICCC).

Singh, K., Singh, N. & Kushwaha, D. S., September 28-29, 2018. *An Interoperable and Secure E-Wallet Architecture based on Digital Ledger Technology using Blockchain.* Greater Noida, Uttar Pradesh, India, 2018 International Conference on Computing, Power and Communication Technologies (GUCON).

S, M. M., Haritha, V., S, S. M. & M, T., May 15-17,2019. *Comparison on Hyper ledger Fabric and Hyper ledger Composer of Block Chain Technology.* Madurai, India, 2019 International Conference on Intelligent Computing and Control Systems (ICCS).

Tanwar, S., Parekh, K. & Evans, R., 2020. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *Journal of Information Security and Applications,* February, Volume 50, pp. 1-13.

Vukoli´c, M., April 2, 2017. *Rethinking permissioned blockchains.* Abu Dhabi; United Arab Emirates, 1st ACM Workshop on Blockchain, Cryptocurrencies and Contracts, BCC 2017.

Wang, X. et al., July 2-7, 2018. *Inter-Bank Payment System on Enterprise Blockchain Platform.* San Francisco, CA, USA , 2018 IEEE 11th International Conference on Cloud Computing (CLOUD).

Wu, B. & Duan, T., June 21-23, 2019. *The advantages of blockchain technology in commercial bank operation and management.* Nanchang University Nanchang; China, 4th International Conference on Machine Learning Technologies, ICMLT 2019.

Yoo, S., 2017. Blockchain based financial case analysis and its implications. *Asia Pacific Journal of Innovation and Entrepreneurship,* 4 December, 11(3), pp. 312-321.