TRADE FINANCE APPLICATION USING HYPERLEDGER FABRIC

[1]Pujah B.
[2]Gokilavani S.
Department of Information Technology
[1,2]Meenakshi Sundararajan Engineering College, Kodambakkam, Ch - 24

ABSTRACT

This paper explores the shift in trade finance using blockchain technology and Hyperledger fabric. Trade finance refers to financial transactions that may be both domestic and international, which relate to trade receivables finance and global trade. One of the difficulties involved in trade finance is that the large volume of paper documents that still structure much of the data flow. Banks are seeking to cut back costs and increase efficiency by replacing the flow of paper for trade finance with digital data flows. Blockchain technology ensures that data is tamper-proof (highly secured), traceable, and trustworthy. The certificate authority, channel, Private Data Collection, etc are the basic framework and privacy protection mechanisms of Hyperledger Fabric. The transactions done through blockchain are tamper-proof as it does not involve a third-party involvement and this blockchain system is decentralized.

I. INTRODUCTION:

Hyperledger (or the Hyperledger project) is an umbrella project of open-source blockchains and related tools, started in December 2015 by the Linux Foundation, and has received contributions from IBM, Intel, and SAP Ariba, to support the collaborative development of blockchain-based distributed ledgers.

While still in the PoC stage, the financial industry is the furthest along in implementing distributed ledger technology. Benefits will include: streamlined settlement, improved liquidity, supply chain optimization, increased transparency, and new products/markets.

The extensive involvement of leading fintech, banking, settlement, and consulting firms in Hyperledger, in combination with the Linux Foundation's legal and technical.

Blockchain and Hyperledger technologies are hot topics today. Hyperledger Fabric and Hyperledger Composer are open source projects that help organizations create private, permissioned blockchain networks. These find applications in finance, banking, supply chain, and IoT among several other sectors. This book will be an easy reference to explore and build blockchain networks using Hyperledger technologies.

The book starts by outlining the evolution of blockchain, including an overview of relevant blockchain technologies. You will learn how to configure Hyperledger Fabric and become familiar with its architectural components. Using these components, you will learn to build private blockchain networks, along with the applications that connect to them. Starting from principles first, you'll learn to design and launch a network, implement smart contracts in chaincode, and much more.

By the end of this book, you will be able to build and deploy your own decentralized applications, handling the key pain points encountered in the blockchain life cycle.

Developed by Satoshi Nakamoto in 2009, Bitcoin is the first decentralized public ledger system. Since then, a number of similar blockchain-based cryptocurrencies have emerged. Blockchain is a distributed data processing protocol for retaining a public distributed ledger in a Peer-to-peer (P2P) network. Transaction data is recorded in blocks, and these blocks form a linked list (i.e., chain) of blocks. Each node in the network stores and maintains an entire copy of the ledger without requiring a central authority. In blockchain-based cryptocurrencies, each block contains the hash value of the previous block, making it hard to manipulate the transactions within. Normally, a consensus protocol is used to guarantee the data integrity among the nodes of the blockchain P2P network. There are several different consensus protocols used in different types of blockchains. Proof-of-Work (PoW) is the most commonly used consensus protocol in blockchain-based cryptocurrencies. Major blockchains such as Bitcoin and Ethereum are both using different varieties of PoW protocol. In the PoW protocol, each node is competing to find a nonce value to produce a hash that meets certain criteria. The difficulty of calculating such a

nonce value can be calculated based on the criteria of the hash value. When such a nonce value is found, a block is generated and broadcasted to the P2P network. Depending on different varieties of protocol, peer nodes always accept the longest chain or the chain with the largest total difficulty repeatedly to continuously expand the blockchain. PoW utilizes this mechanism to determine which node has the right to seal a block.

This process is also called mining. In such a mechanism, a peer node with greater computing speed (or sometimes called hash rate power) can calculate nonce value faster than a peer node with less computing speed and thus has a higher probability of getting the right to seal a new block. However, this mechanism has a drawback. A selfish node with hash rate power higher than the total hash rate power of the rest nodes combined can compromise the blockchain system by causing double-spending and selfish mining, etc. This is commonly referred to as a 51% attack. Some studies have proposed ways to avoid such kind of attacks. Eyal and Sirer in 2014 proposed a Two-Phase PoW(2P-PoW) solution preventing the formation of a mining pool with huge hash power. In this solution, the second phase PoW requires a signature from the private key of the coinbase address. When the second PoW is sufficiently difficult, pool operators have to give out this private key to the pool miners in order to perform a calculation faster than all the peer nodes. Ruffing et al. in 2015 proposed contracts to penalize attackers attempting a double-spending attack. Solat and Potop-Butucaru proposed ZeroBlock in 2016. The mechanism in ZeroBlock requires a block to be accepted by its peers within a certain time interval after the timestamp of the block. Otherwise, the block is expired. This mechanism prevents attacker nodes from selfish mining for a long period of time. J. Bae and H. Lim in 2018 proposed a solution to randomly select a certain group of miners to have the right to mine the next block. In the present paper, we introduce a new solution by utilizing a historically weighted difficulty to determine the total chain difficulty. In such a modified algorithm, a branch of blockchain has a greater total historical weighted difficulty if the miners of such a branch have a higher coverage rate in previous blocks. We demonstrate that, in reality, such an algorithm can increase the monetary cost and the time cost of 51% attacks by a factor of at least two orders of magnitude.

II. HISTORY:

We propose an improved technique to calculate the total difficulty of a branch. This technique takes into account the distribution of miner addresses in the last certain amount of blocks of the blockchain. We call this protocol Historical Weighted Difficulty based Proof-of-Work (HWD-PoW) protocol. The assumption is that in an honest blockchain branch, miners of new blocks will most likely be the miners who mined the previous blocks, and the distribution will reflect the ratio in history. Furthermore, in a malicious blockchain branch, the distribution of miners of new blocks will most likely be controlled by the attackers, which will be different from the regular distribution of miners in history. Therefore, when the distribution of the miners in history is considered, one can easily distinguish an honest blockchain branch from a malicious one. Under the proposed mechanism, branch with miners of less representation in the previous blocks will earn less weight in the total difficulty calculation. Therefore, to perform a 51% attack, the malicious miners have two choices: either to mine a much longer branch or to build up miner representation in the previous blocks to build up the credibility.

III. RELATED WORK:

a) Create the Genesis Block

geth --datadir=./data init start.json

b) Start a local instance of Ethereum

Mac environment:

geth --networkid 999 --ipcpath ~/Library/Ethereum/geth.ipc --rpc --rpcaddr "127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport "8545" --datadir=./data --rpccorsdomain "*" console

Linux environment :

geth --networkid 999 --ipcpath $HOME/.ethereum/geth.ipc --rpc --rpcaddr "127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport "8545" --datadir=./data --rpccorsdomain "*" console.

Windows environment:

geth --networkid 999 --rpc --rpcaddr "127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport "8545" --datadir=./data --rpccorsdomain "*" console

c) Create the Main Account

web3.personal.newAccount("password")

d) View the list of accounts

web3.personal.listAccounts

f) Get the balance associated with the main account

web3.fromWei(eth.getBalance(eth.coinbase));

g) Attach to Geth console

Linux Environment - geth attach ~/Library/Ethereum/geth.ipc

Windows environment - geth attach

or

geth attach ipc:\\.\pipe\geth.ipc (for geth 1.8 and above)

Note, if you have a time out issue, kindly check the IPC endpoint value in Step 1 b above when you start the ethereum local instance. Look at IPC endpoint opened value, and the value could be

different for your setup , like the following - \\\\.\\pipe\\~Library/Ethererum/geth.ipc and use this value in get attach, i.e get attach \\\\.\\pipe\\~Library/Ethererum/geth.ipc

h) Start Miner

miner.start(2)

i) Run end-to-end execution

./tradee2e.sh -up

IV. OUTPUT:

```
INFO [03-09|23:13:43.020] Maximum peer count                       ETH=50 LES=0 total=50
INFO [03-09|23:13:43.096] Allocated cache and file handles         database=C:\\Users\\poojah\\first-edition\\ethereum\
\setup\\data\\geth\\chaindata cache=16.00MiB handles=16
INFO [03-09|23:13:43.122] Writing custom genesis block
INFO [03-09|23:13:43.125] Persisted trie from memory database      nodes=0 size=0.00B time=0s gcnodes=0 gcsize=0.00B gc
time=0s livenodes=1 livesize=0.00B
```

```
> INFO [03-09|23:15:39.023] Mapped network port                     proto=udp extport=30303 intport=30303 interface="U
PNP IGDv1-IP1"
INFO [03-09|23:15:39.033] Mapped network port                      proto=tcp extport=30303 intport=30303 interface="UPN
P IGDv1-IP1"
INFO [03-09|23:15:39.371] New local node record                    seq=2 id=86393a31aafac00d ip=10.217.247.43 udp=30303
 tcp=30303
```

```
PS C:\Users\poojah> geth --networkid 999 --rpc --rpcaddr "127.0.0.1" --rpcapi="db,eth,net,web3,personal,web3" --rpcport
 "8545" --datadir=./data --rpccorsdomain "*" console
INFO [03-09|22:35:05.415] Maximum peer count                       ETH=50 LES=0 total=50
INFO [03-09|22:35:05.497] Starting peer-to-peer node               instance=Geth/v1.9.9-stable-01744997/windows-amd64/g
o1.13.4
INFO [03-09|22:35:05.505] Allocated trie memory caches             clean=256.00MiB dirty=256.00MiB
INFO [03-09|22:35:05.509] Allocated cache and file handles         database=C:\\Users\\poojah\\data\\geth\\chaindata ca
che=512.00MiB handles=8192
INFO [03-09|22:35:05.556] Opened ancient database                  database=C:\\Users\\poojah\\data\\geth\\chaindata\\a
ncient
INFO [03-09|22:35:05.562] Writing default main-net genesis block
INFO [03-09|22:35:05.783] Persisted trie from memory database      nodes=12356 size=1.79MiB time=39.8925ms gcnodes=0 gc
size=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [03-09|22:35:05.795] Initialised chain configuration          config="{ChainID: 1 Homestead: 1150000 DAO: 1920000
DAOSupport: true EIP150: 2463000 EIP155: 2675000 EIP158: 2675000 Byzantium: 4370000 Constantinople: 7280000 Petersburg:
 7280000 Istanbul: 9069000, Muir Glacier: 9200000, Engine: ethash}"
INFO [03-09|22:35:05.810] Disk storage enabled for ethash caches   dir=C:\\Users\\poojah\\data\\geth\\ethash count=3
INFO [03-09|22:35:05.816] Disk storage enabled for ethash DAGs     dir=C:\\Users\\poojah\\AppData\\Local\\Ethash count=
2
INFO [03-09|22:35:05.821] Initialising Ethereum protocol           versions="[64 63]" network=999 dbversion=<nil>
WARN [03-09|22:35:05.825] Upgrade blockchain database version      from=<nil> to=7
INFO [03-09|22:35:05.830] Loaded most recent local header          number=0 hash=d4e567…cb8fa3 td=17179869184 age=50y11
mo17h
INFO [03-09|22:35:05.837] Loaded most recent local full block      number=0 hash=d4e567…cb8fa3 td=17179869184 age=50y11
mo17h
INFO [03-09|22:35:05.842] Loaded most recent local fast block      number=0 hash=d4e567…cb8fa3 td=17179869184 age=50y11
mo17h
INFO [03-09|22:35:05.857] Regenerated local transaction journal    transactions=0 accounts=0
INFO [03-09|22:35:05.873] Allocated fast sync bloom                size=512.00MiB
INFO [03-09|22:35:05.893] New local node record                    seq=1 id=943808f35e79c7cf ip=127.0.0.1 udp=30303 tcp=30303
INFO [03-09|22:35:05.904] Started P2P networking                   self=enode://1185154a1c0f75443c49d18a29cb23aee32194294c49bde02e3ed30b73643f771846f89392bdfd68aefd237
d330e65c54d4d10a40ae4012672e29b169f7a4890@127.0.0.1:30303
INFO [03-09|22:35:05.906] IPC endpoint opened                      url=\\\\.\\pipe\\geth.ipc
INFO [03-09|22:35:05.924] HTTP endpoint opened                     url=http://127.0.0.1:8545 cors=* vhosts=localhost
INFO [03-09|22:35:05.954] Initialized fast sync bloom              items=12356 errorrate=0.000 elapsed=76.794ms
```

```
> web3.personal.newAccount("password")
INFO [03-09|23:17:38.413] Your new key was generated              address=0x3DeF12A34a03D5F9aa984743C1d04266DF3f5b74
WARN [03-09|23:17:38.421] Please backup your key file!            path=C:\\Users\\poojah\\first-edition\\ethereum\\set
up\\data\\keystore\\UTC--2020-03-09T17-47-37.028191400Z--3def12a34a03d5f9aa984743c1d04266df3f5b74
WARN [03-09|23:17:38.430] Please remember your password!
"0x3def12a34a03d5f9aa984743c1d04266df3f5b74"
> web3.personal.listAccounts
["0x3def12a34a03d5f9aa984743c1d04266df3f5b74"]
> web3.fromWei(eth.getBalance(eth.coinbase));
INFO [03-09|23:18:41.327] Etherbase automatically configured      address=0x3DeF12A34a03D5F9aa984743C1d04266DF3f5b74
0
```

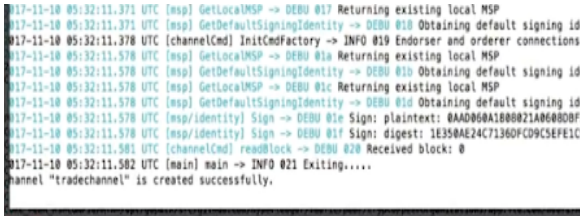
```
> miner.start(2)
INFO [03-09|23:21:34.643] Updated mining threads                  threads=2
INFO [03-09|23:21:34.647] Transaction pool price threshold updated price=1000000000
null
IN> FO [03-09|23:21:34.651] Commit new mining work                  number=1 sealhash=c15d3a…5677ac uncles=0 txs=0 gas
=0 fees=0 elapsed=998.1µs
INFO [03-09|23:21:36.033] Generating DAG in progress              epoch=1 percentage=0 elapsed=796.869ms
INFO [03-09|23:21:36.836] Generating DAG in progress              epoch=1 percentage=1 elapsed=1.599s
INFO [03-09|23:21:37.667] Generating DAG in progress              epoch=1 percentage=2 elapsed=2.430s
INFO [03-09|23:21:38.462] Generating DAG in progress              epoch=1 percentage=3 elapsed=3.225s
INFO [03-09|23:21:39.272] Generating DAG in progress              epoch=1 percentage=4 elapsed=4.036s
INFO [03-09|23:21:40.131] Generating DAG in progress              epoch=1 percentage=5 elapsed=4.894s
INFO [03-09|23:21:41.031] Generating DAG in progress              epoch=1 percentage=6 elapsed=5.794s
INFO [03-09|23:21:41.892] Generating DAG in progress              epoch=1 percentage=7 elapsed=6.656s
INFO [03-09|23:21:42.715] Generating DAG in progress              epoch=1 percentage=8 elapsed=7.478s
INFO [03-09|23:21:43.601] Generating DAG in progress              epoch=1 percentage=9 elapsed=8.364s
INFO [03-09|23:21:44.393] Generating DAG in progress              epoch=1 percentage=10 elapsed=9.156s
INFO [03-09|23:21:45.233] Generating DAG in progress              epoch=1 percentage=11 elapsed=9.997s
INFO [03-09|23:21:46.065] Generating DAG in progress              epoch=1 percentage=12 elapsed=10.828s
INFO [03-09|23:21:46.968] Generating DAG in progress              epoch=1 percentage=13 elapsed=11.731s
INFO [03-09|23:21:47.847] Generating DAG in progress              epoch=1 percentage=14 elapsed=12.611s
INFO [03-09|23:21:48.653] Generating DAG in progress              epoch=1 percentage=15 elapsed=13.417s
INFO [03-09|23:21:49.486] Generating DAG in progress              epoch=1 percentage=16 elapsed=14.249s
INFO [03-09|23:21:50.383] Generating DAG in progress              epoch=1 percentage=17 elapsed=15.146s
INFO [03-09|23:21:51.258] Generating DAG in progress              epoch=1 percentage=18 elapsed=16.022s
INFO [03-09|23:21:52.132] Generating DAG in progress              epoch=1 percentage=19 elapsed=16.895s
INFO [03-09|23:21:53.022] Generating DAG in progress              epoch=1 percentage=20 elapsed=17.785s
```

```
/root/hyperledger-fabric/bin/cryptogen

Generate certificates using cryptogen tool ..

app.fte.com
bnk.com
shp.com
```

```
/root/hyperledger-fabric/bin/configtxgen
.. Generating Orderer Genesis block ...
```



V. EXISTING WORK:

European option binomial tree method in the carbon trading exchange on the condition "Carbon finance"

With the increasingly evident effects of greenhouse gases, a low carbon economy becomes a national inevitable choice for each country's sustainable development. Greenhouse gas emissions at an increasing scarcity of prominent gradually developed into a scarce commodity, promoting the carbon emissions carbon trading market as the center of production and development. With

the "United Nations Framework Convention on Climate" and the "Kyoto Protocol" signed, Industrial countries have been forced that greenhouse gas emissions must be reduced by 5.2%, which based on the average

reduction in 1990, from 2008 to 2012, the first commitment period. Major countries and enterprises should be divided into a certain amount of allowanced carbon emissions. The unused quota can be traded in the market trading system, under the condition that the task of reducing emissions are completed, which also increased the enthusiasm of participants in environmental protection. From the "Kyoto Protocol" comes into effect in 2005, the global carbon trading market comes into the rapid development stage. Carbon trading was still to maintain a high growth trend during the financial crisis in 2008. Carbon trading volume jumped from 2.7 billion tones in 2007 to 59 million tons in 2008, increased by 150.56%. Turnover high up to 1250 million Euros, compared to 40 billion Euros in 2007 increased by 212.5%. The EU emissions trading system (EU ETS) in the lead in the global carbon market, a large annual volume 3.776 billion tons in 2008, the transaction amounted to 90 billion Euros, accounting for 64% of the global trading volume and turnover 72% ."Carbon finance" is the financial system and transactions associated with carbon emissions. In the carbon trading market, it can simplify the complexities of carbon trading through innovative tools. Besides, it can also provide investment financing the development and strengthening of financial support, increased market activity. But we can not blind optimism, there are still potential problems. Based on the source that international cooperation unity in the international market has not been established inadequately, the split OTC trading markets and exchange-traded, compulsory and voluntary exchange transactions, transactions between different trading systems, and other issues, there are still hinder obstructions in this virtual product mobility. In addition, emission reduction units can not be achieved yet cross-market trading based on different systems. Therefore, this article selected option as the access point, compared with its own characteristics, American options.

Mechanism of Credit Guarantee Pricing in Finance Trade

Credit guarantee plays an important role in financial trade, especially in small and medium enterprise finance. Credit guarantee profession is a high-risk profession, and for guarantee agency, the scientific mechanism of guarantee pricing can strengthen the ability to control risk, so the problem of guarantee pricing has been emphasized by the international academic circle. Concerning the approach to guarantee a pricing problem, it will deserve to be deepened and developed, which will possess important significance theoretically and practically. Guarantee pricing is just compensation for guarantee risk scale, which response to the continuous development of guarantee business, so it is pretty urgent to approach the guarantee pricing problem. So far as a study of the risk pricing methods, Duan approached the pricing model of deposits insurance, Lai approached the numerical pricing method in a two-case condition. Up-to-date, references to guarantee to price is very rare, the past approach is confined to empirical pricing. Owing to the defect of empirical pricing, references represent an 0

The Design of the International Settlement and Trade Finance System

With the world economic integration with the global economic-financial, global trade environment and financial environment have changed dramatically. The vigorous development of international trade and financial liberalization of the increasingly deep not only provides paused for international trade settlement of a sharp rise in the development of space but also caused the method of payment and the profound change of the business content, etc. These changes, on the one hand, driven by the changes in international trade activities, on the other hand, they also help promote the development of international trade. International settlement and trade finance system is an international business deal with one of the important systems of commercial Banks. Promote international clearing bank in trade finance business has become an important means of service quality and business premise, in the light of the operation of derivatives trade finance business and policy risk management, the system achieved the goal of risk monitoring, realize the scientific management of trade financing business and real-time

monitoring, for the healthy development of commercial Banks trade finance business provides effective technical support. This paper mainly introduces the international settlement and trade finance system under the background of international financial application design, and represented by the bank for international settlements trade financing system, gives the bank of international settlement and trade finance system design and implementation. Especially on the overall planning and design process of the system, and the localization, the design principle of the system is introduced and the relationship between the various related systems, international settlement, and trade finance was introduced in detail the implementation of the various business functions and the detailed design, and application based on B/S structure of the system is discussed the corresponding implementation.

VI. EXISTING SYSTEM:

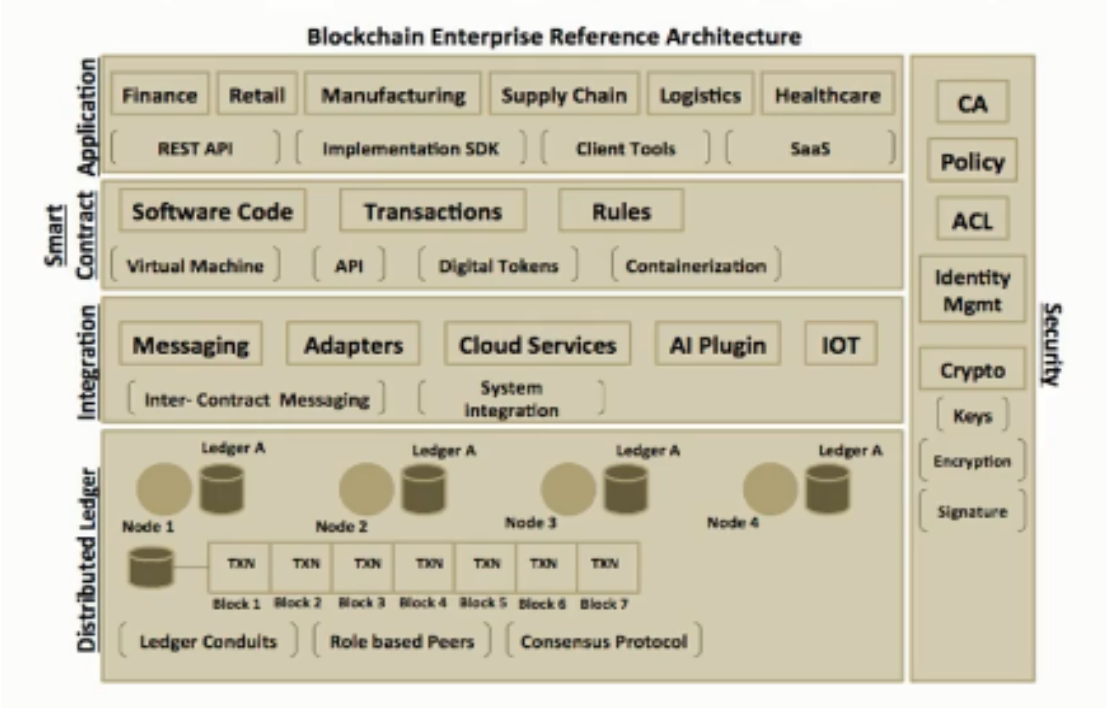
Hyperledger (or the Hyperledger project) is an umbrella project of open-source blockchains and related tools, started in December 2015 by the Linux Foundation, and has received contributions from IBM, Intel, and SAP Ariba, to support the collaborative development of blockchain-based distributed ledgers.

While still in the PoC stage, the financial industry is the furthest along in implementing distributed ledger technology. Benefits will include: streamlined settlement, improved liquidity, supply chain optimization, increased transparency, and new products/markets.

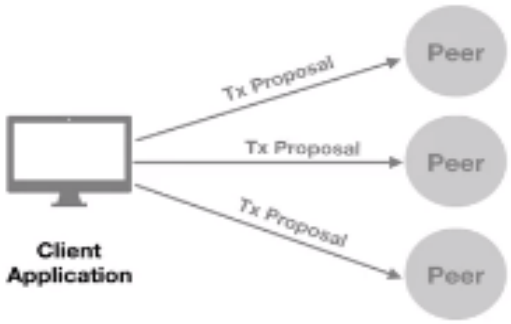
The extensive involvement of leading fintech, banking, settlement, and consulting firms in Hyperledger, in combination with the Linux Foundation’s legal and technical.

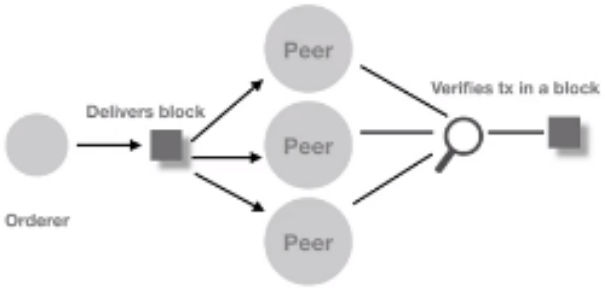
Blockchain and Hyperledger technologies are hot topics today. Hyperledger Fabric and Hyperledger Composer are open source projects that help organizations create private, permissioned blockchain networks. These find applications in finance, banking, supply chain, and IoT among several other sectors. This book will be an easy reference to explore and build blockchain networks using Hyperledger technologies.

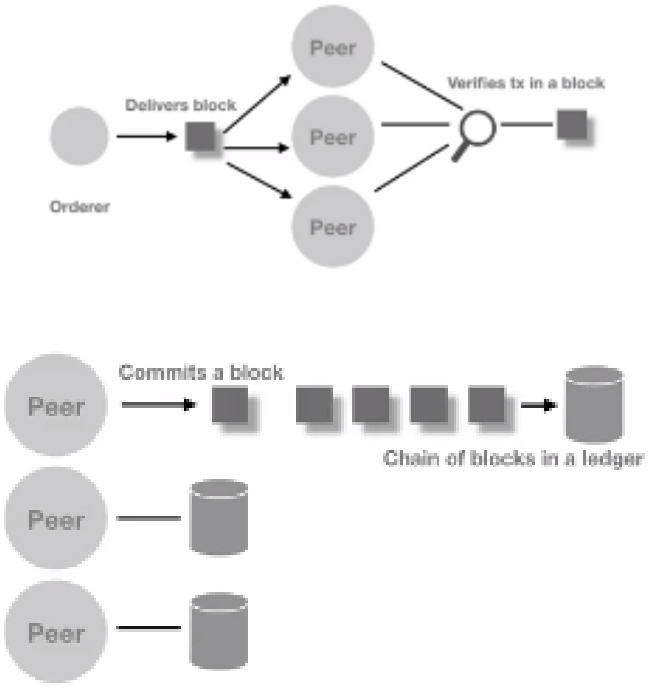
VII. EXPERIMENTAL SETUP & RESULT ANALYSIS:



VIII. WORKING:







IX. FUTURE ENHANCEMENT:

In the future, using Blockchain all the applications can do the transaction without tampering and loopholes. It introduces the international settlement and trade finance system under the background of international financial application design, and represented by the bank for international settlements trade financing system, gives the bank of international settlement and trade finance system design and implementation. Especially on the overall planning and design process of the system, and the localization, the design principle of the system is introduced and the relationship between the various related systems, international settlement, and trade finance was introduced in detail the implementation of the various business functions and the detailed design, and application based on B/S structure of the system is discussed the corresponding implementation.

X. CONCLUSION:

Hence the Trade finance application has done the transactions using hyperledger fabric. So by using the hyperledger fabric, any application can be transformed into the blockchain-based application. By using blockchain, security and authentication will be assured.