International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

# Blockchain based Data Sharing System for Supply Chain

Kurniawan Winata
Department of Information Science and Technology
Zhejiang Sci-Tech University (ZSTU)
Hangzhou City, China

*Abstract*— **In this paper, we designed a consortium network based on permission blockchain to serve as secure data sharing system for supply chain. The design has the ability to provide data privacy among organization within the network. Organizations are able to share data only to specific group of organizations in the network that have been granted permission for accessing the data. The shared data are not accessible from any organizations outside the group. Data privacy is achieved by using network channel to define a group and its member. The obtained result proves that our proposed blockchain network design is successfully provides data privacy and data transparency among organization in the network.**

***Keywords—Blockchain; consortium network; supply chain.***

## I. INTRODUCTION

Blockchain technology was initially invented for financial application [1] and it was heavily used in this field. Famous public blockchain networks offer cryptocurrency and their financial services dominate blockchain use cases [2] [3] [4] [5]. Blokchain is basically a digital distributed ledger that is not only applicable for financial application but also in other sector that embracing distributed system. However, in recent years blockchain implementations are far beyond cryptocurrency. Blockchain has been implemented across many different fields ranging from financial services [6] [7], health care [8] [9], internet of things [10] [11] to government services [12] [13].

The key characteristics of blockchain technology are decentralization, security, immutability, and auditability. In traditional system, application depends on trusted central authority to provide network infrastructure and transaction verification. The dependency on central authority makes the system vulnerable. Blockchain technology has the capability to eliminate central authority and verify each transaction itself using distributed network. Blockchain can provide the same functionality with the same level of reliability without the need of central authority. This is possible because blockchain works in decentralized fashion utilizing consensus mechanism and using cryptography to secure the system.

One potential use case that benefited from blockchain is supply chain management. The advantage of using blockchain in supply chain management could be seen in several aspects. Firstly, it improves transaction speed and reduces transaction cost. Blockchain uses consensus mechanism and smart contract to enable instant settlement of transaction. It removes the need of intermediaries that could cause the delay. The involvement of intermediaries have resulted in high transaction fee, blockchain replaces third-party intermediaries and reduces all cost related to it [14]. Secondly, blockchain improves transparency and improve trust between supplier and customer. Customers can get more information about the product they purchase, including complete manufacturing history [15] [16]. The immutable ledger serves as one single source of truth that distributed among all network members therefore all parties can verify the integrity of transactions that have been committed to the blockchain. Finally, blockchain improves product traceability. In [17], retail chain walmart successfully solves food safety issue in their supply chain using blockchain technology. They use blockchain to enable food traceability to the item level, so that participants can trace each item in the supply chain. The ability to rapidly trace products in the supply chain is important to prevent or responding quickly to food contamination or other food safety issues.

Despite the fact that blockchain technology offers significant advantages to supply chain, it is facing a number of challenges. One of the concerns that need to be addressed is lack of data privacy because data are publicly available for anyone to see and verify. This might not be well suited for supply chain management because in most cases some transactions are confidential from other parties on the network that have competing business interest. In this paper, we propose a design of blockchain network that has the ability to provide data privacy on the network so that competing business interest can coexist in the same network.

Blockchain has been used in the area of supply chain. Each of them have different approach on the system design In [18], the system uses consortium blockchain with on-chain and off-chain data storage. Only nonconfidential information is stored in the blockchain and it is visible to everyone. Confidential information is stored in centralized database. In subsequent work they use the same approach [19]. In [20], BigChainDB is used to build traceability system. Data are stored in BigChainDB which act as public blockchain. All data in BigChainDB is open to any user. In [21], the work focuses on agricultural supply chain system using double chain architecture. The system uses public blockchain that has two blockchains as data storage. The first chain stores user information and the second chain stores transactions. In [22], they focus on verification system for off-chain data using Ethereum-like blockchain. In [23], Ethereum public blockchain network is used to build anti-counterfeits system. All data are open to all users.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

## II. METHOD

The proposed blockchain network is a consortium network where every member has valid identity. The network is governed by one organization in the network that acts as network administrator. It is a permissioned network which requires approval from network administrator before an organization can join the network. Our proposed blockchain network is implemented using hyperledger fabric. We design blockchain network with three organizations on the network, the overall design is shown in Fig. 1.

### *A. Identity and authorization*

Members need to prove their identity in order to participate on the network. The system uses public key infrastructure (PKI) to provide reliable and secure identity. PKI provide all network's members with digital identity attached to them. The identity is an X.509 digital certificate that is part of PKI. The identity is used for producing digital signature on transaction. All transaction made in the network must be digitally signed by the transaction's owner. The identity also determines the exact permission over resources and access to information that a members have in a blockchain network.

Identity consists of a public key and private key which form a key pair. Public key is open to all network's member but private key is a secret key that only the owner can access to it. Transaction's owner uses their private key to digitally sign a transaction. The digital signature produces by private key match with only its corresponding public key. Ordering node has the mechanism to store public key and verify its validity. When a digitally sign transaction is sent to ordering node for verification, ordering node use the corresponding public key to verify the transaction validity and use it to justify whether the owner is authorized to endorse the transaction. This mechanism allows identity to be trusted and recognized by the rest of the network without the need to revealing the member's private key.

### *B. Ordering node*

Ordering node provides shared communication channel to clients and peers, offering a broadcast service for messages containing transactions. Ordering node has three main roles in the network. The first role is managing channel authorization. Ordering node maintains the list of organizations that are allowed to create channel. It enforces basic access control for channels, restricting who can read and write data to them, and who can configure them.

The second role is block construction on the transaction flow process. Ordering node plays a role as the receiver of endorsed transaction proposal response from client. Ordering node receives endorsed transaction from many different clients concurrently. After receiving the transactions, ordering node arranges submitted transactions into a well-defined sequence and packages them into a block. The number of transaction in a block can be configured on channel configuration. The order of transaction in the block follows by strict order and does not necessarily follow the order received by ordering node. However, once a transaction has been written to a block, its position in the ledger is immutable.

The third role is block distribution. After a block is formed, ordering node will distribute the block to all peers on the channel. Each peer will validate the block and committed to the ledger. Ordering node can be managed by one or many organization in the network. Organizations who manage ordering node have authority over consortium that forms a channel.

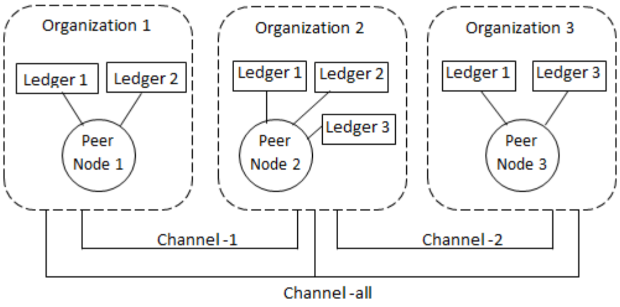


Fig. 1. Conceptual design of the proposed blockchain network

### *C. Peer node*

Peer is a network node that host instance of ledgers and instance of smart contract. Peer node can host many ledgers and many smart contracts. Ledger stores all history of transaction that result the current values. Smart contract is a code that accesses the ledger. In order to perform data manipulation on the ledger, applications need to invoke smart contact. Applications don't have direct access to the ledger. Since smart contracts are stored on peer node therefore client application must have access to peer node so that they can invoke smart contract. Although technically it is possible for peer node not to host any smart contract but most of the time ledger and smart contract coexist on peer node. Peer node has important role in the transaction flow, there are two different type transactions that require interaction between peer node and client application: ledger query and ledger update.

In ledger query process, client application connects to peer-node and submit proposal to invoke smart contract. Peer node invokes smart contract to query ledger. Smart contract is executed and query against the local ledger. Peer node then return the query result to client application. Since all peer nodes host the same exact copy of the ledger, peer node that receive proposal from client application can immediately response by querying its local copy of the ledger without the need to consult to another peer node to respond a query. Client application can connect to one or more peer-node to send query proposal to retrieve a more up to date result from different peer if there is a suspicion that query result is out of date, however normally it is not the case.

In the ledger update process, transaction starts with client application making connection to peer node and sent proposal to invoke smart contract. Peer node invokes smart contract to update the ledger. Smart contract is executed and generates update proposal response. A single peer node is not allowed to commit ledger update on its own. It requires approval from all peers in the network. Execution of smart contract does not update the ledger but it generate update proposal. Peer node sent proposed ledger update to client application. Furthermore, application sends appropriate proposed ledger update to ordering node. Ordering node will collect all proposed ledger update in a batch and packed them in to one block.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Before the new block can be added in the ledger, all peers in the network must agree to the change. Agreement between peer is achieved by using consensus mechanism. Ordering node sent the newly constructed block is broadcasted to all peers in the network for consensus. After peer nodes verifying the block and achieve consensus the block is committed to the ledger. The whole ordering process will take some times to complete, after it finish peer node will send notification of ledger update confirmation to the application.

The blockchain network is fully formed when many organization set up their peer nodes in the network. When one peer node is down the network is still fully functioning as long as one organization remains. This is how the design achieves decentralization nature of blockchain network.

### D. *Ledger*

Ledger is stores the current states and all history of transactions that led to the current state. A ledger consists of two parts: a world state and a blockchain. Each of these represents a set of fact about the current states. Ledger is hosted in the peer node. A peer node can host one or more ledger. One ledger is associated with one network channel. Consensus mechanism ensures the consistency of all copy of ledger in different peer nodes within one network channel. Logically only one ledger on each network channel even though physically the network maintain multiple copies of the ledger.

The first component is world state. A world state is a database that holds current values of a set of ledger states. The world state does not contain detail historical transaction but only the current value of a state. The world state is designed to provide easy and fast access for client application to query the current value of a state without having to calculate it by traversing the entire transaction log. In many use cases, client application just needs to query the current value of a state without the need of detail transaction history. Although generating current value of a state from blockchain is possible but it is not an efficient way to response a query because it takes time and requires some degree of computation depends on the size of blockchain. In order to address this issue the proposed approach is to separate the storage of current value of states from their historical transaction. The current value of states is stored in world state; this approach is able to speed up query process. The world state can change frequently, as states can be created, updated and deleted.

Data structure of ledger states is key-value pairs. The world state is implemented as database because it provides efficient data storage and great flexibility for data retrieval. Database technology is mature, stable and fast enough to store key value pair and it provides convenient way for data retrieval. When the ledger is first created, the world state is empty. The world state can be generated anytime by calculate it from transaction history that stored in the blockchain.

Second component of the ledger is blockchain. In this design, blockchain component is a transaction log that records all the change that resulted in the current world state. It is a historical record of all committed transaction. Transactions are stored inside the blocks that are appended to blockchain. The blockchain is immutable, it cannot be modified.

The blockchain is formed by sequence of interconnected block. Each block contains a sequence of transaction. Each transaction represents a query or update to the world state. Each block is linked to each other by storing hash value of the block's transaction and a hash of previous block's header so that all transactions on the blockchain are linked together. By linking to previous block's header the data stored in the blockchain cannot be altered without breaking the link therefore data are secure. If one peer node is cheating and break the link, all other nodes that have the correct version of blockchain will reject the illegal version of blockchain.

### E. *Smart contract*

Smart contract is basically an executable code that contains rules between different organizations. It defines transaction logic that controls the lifecycle of states stored in the world states. Smart contract is hosted in the peer node but it is logically deployed in the network hence it is available to connected client application in the network. Multiple smart contracts can be deployed in the network. Smart contract has access both to blockchain and world state. Client application does not have direct access to the ledger; therefore the only way to access the ledger is through invoking smart contract.

Smart contract deployment on the channel needs an agreement by all member of the channel before it can be executed. Smart contract is used to generate transaction which can be distributed to every node in the network. Smart contract performs four operations against the ledger:

- Create new or modify an existing value of state in the world state
- Query to retrieve the current value of a state in the world state.
- Delete the current state from world state, however its history can't be deleted.
- Query to retrieve transaction information from blockchain.

### F. *Network channel*

Within the network, there might be a group of organizations that have common goal and they need to transact with one another without involving the rest of the network's members. These organizations tend to form a private group and transact to one another. There might be several groups within the network and each group will have a need for different information to be appropriately shared. In order to accommodate private group within the network the design uses network channel. Network channel provide an efficient mechanism to do private communication.

Network channel is a private sub network of communication between two or more network members so that they can perform private and confidential transaction. Private transaction means that their transaction is not visible by other node outside the network channel. Network channel provides completely separate communication mechanism between a set of organizations. Network channel keep transaction private from the broader network. Network channel provides efficient data sharing while still maintains data and communication privacy. It bridges the gap between total transparency and privacy. Network channel provides

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

privacy from other channel and from the network but it provides transparency for organizations connected to the same channel.

Blockchain network can have multiple network channels depend on the network design. A channel consists of peer node, shared ledger, smart contract and ordering node. One organization can participate in multiple channels and one channel can have multiple organizations connected to it. Even though network channel is part of the network but it is logically distinguishable from it. Network channel has its own channel configuration that separate from network configuration. Data in a channel are completely isolated from the rest of the network, including other channel. This means that each channel has its own ledger that separate from other channel and the rest of the network. If peer node joins multiple channels then it hosts multiple ledgers, at least there is one ledger per channel. As shown in Fig. 1, ledger 1 is associated with channel-all, ledger 2 is associated with channel-1 and ledger 3 is associated with channel-2.

Through the implementation of network channel that isolate peer node and ledger by channel, it allows business competitors coexist on the same network because their need of private and confidential transaction is guaranteed. Network channel is important component in the network because it provides data privacy needed in supply chain. As shown in Fig. 1, there are three network channels in the network: channel-all, channel-1 and channel-2. Channel-1 is used for private communication between organization1 and organization 2. Channel-2 is used for private communication between organization 2 and organization 3. Channel-all is use for public communication.

## III. EXPERIMENT AND RESULT

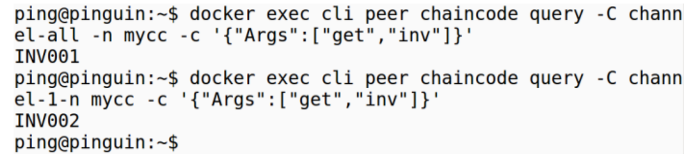
The network nodes are implemented using docker container. Each network node is running on separate docker container. In order to simulate the blockchain network, multiple connected containers run on the same machine. We use docker compose to run multi container application on localhost. The network namespace is example.com. The network has four organizations:

- Ordering node : orderer.example.com
- Organization 1 : org1.example.com (supplier)
- Organization 2 : org2.example.com (manufacturer)
- Organization 3 : org3.example.com (distributor)

Each organization in org1, org2, org3 has one peer node which label as peer0. In the network level there is one ordering node. In total we have four nodes in the network, as follows:

- orderer.example.com
- peer0.org1.example.com
- peer0.org2.example.com
- peer0.org3.example.com

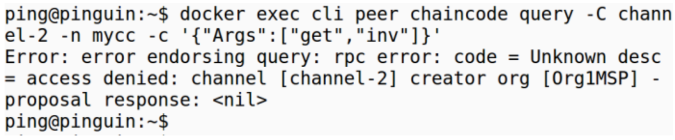
Furthermore, we create one transaction on each channel. INV001 on channel-all, INV002 on channel-1, and INV003 on channel-2. We verify the data privacy capabilities on network channel by executing query from perspective of organization1. Fig. 2 shows the query returns the correct result on channel-1 and channel-all. The result shows that organization1 has access to channel-all and channel-1.



```
ping@pinguin:~$ docker exec cli peer chaincode query -C chann
el-all -n mycc -c '{"Args":["get","inv"]}'
INV001
ping@pinguin:~$ docker exec cli peer chaincode query -C chann
el-1-n mycc -c '{"Args":["get","inv"]}'
INV002
ping@pinguin:~$
```

Fig. 2. Query result on channel-all and channel-1 by organization 1

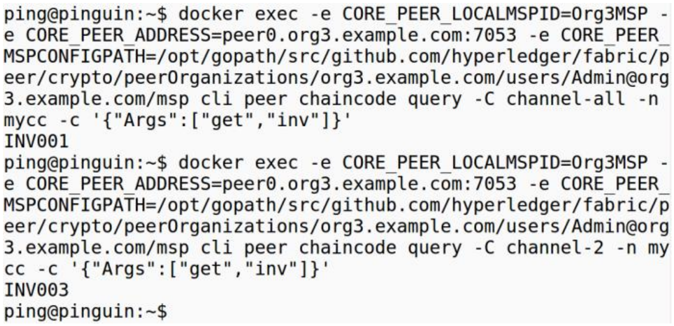
However, the query return an error message from channel-2 showing access denied as shown in Fig. 3. It is because organization 1 is not a member of network channel-2 therefore it doesn’t have any access to transactions made on channel-2



```
ping@pinguin:~$ docker exec cli peer chaincode query -C chann
el-2 -n mycc -c '{"Args":["get","inv"]}'
Error: error endorsing query: rpc error: code = Unknown desc
= access denied: channel [channel-2] creator org [Org1MSP] -
proposal response: <nil>
ping@pinguin:~$
```

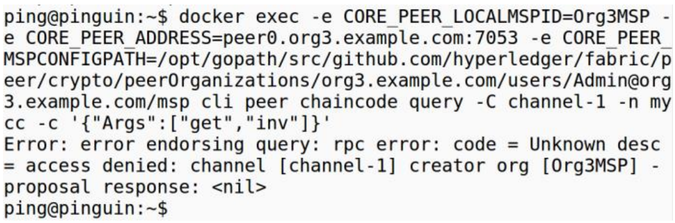
Fig. 3. Query result on channel-2 by organization 1

We got consistent result when we execute queries as organization 3. The queries are executed successfully on channel-all and channel-2 as shown in Fig. 4. Since organization 3 is not a member of channel-1, the query returns error when it is executed on channel-1 as shown in Fig. 5.



```
ping@pinguin:~$ docker exec -e CORE_PEER_LOCALMSPID=Org3MSP -
e CORE_PEER_ADDRESS=peer0.org3.example.com:7053 -e CORE_PEER_
MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/p
eer/crypto/peerOrganizations/org3.example.com/users/Admin@org
3.example.com/msp cli peer chaincode query -C channel-all -n
mycc -c '{"Args":["get","inv"]}'
INV001
ping@pinguin:~$ docker exec -e CORE_PEER_LOCALMSPID=Org3MSP -
e CORE_PEER_ADDRESS=peer0.org3.example.com:7053 -e CORE_PEER_
MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/p
eer/crypto/peerOrganizations/org3.example.com/users/Admin@org
3.example.com/msp cli peer chaincode query -C channel-2 -n my
cc -c '{"Args":["get","inv"]}'
INV003
ping@pinguin:~$
```

Fig. 4. Query result on channel-all and channel-2 by organization 3



```
ping@pinguin:~$ docker exec -e CORE_PEER_LOCALMSPID=Org3MSP -
e CORE_PEER_ADDRESS=peer0.org3.example.com:7053 -e CORE_PEER_
MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/p
eer/crypto/peerOrganizations/org3.example.com/users/Admin@org
3.example.com/msp cli peer chaincode query -C channel-1 -n my
cc -c '{"Args":["get","inv"]}'
Error: error endorsing query: rpc error: code = Unknown desc
= access denied: channel [channel-1] creator org [Org3MSP] -
proposal response: <nil>
ping@pinguin:~$
```

Fig. 5. Query result on channel-1 by organization 3

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

## IV. CONCLUSION

In this paper, we presented blockchain network design based on consortium network and network channel. The results we got from series of queries show that our network design can provide data privacy from organizations outside the channel and in the same time it ensures data transparency between members inside the channel. Network channel is an important component in providing data privacy in the network. Comparing with public blockchain network, the proposed design is able to provide secure data privacy system that is missing in the public blockchain network.

## REFERENCES

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2009.

[2] V. Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform," 2013. [Online]. Available: https://ethereum.org/en/whitepaper.

[3] D. Mazieres, "The stellar consensus protocol:," Stellar Development Foundation, 2016.

[4] D. Schwartz, N. Youngs and A. Britto, "The ripple protocol consensus algorithm," 2018.

[5] Litecoin, "Litecoin," 2011. [Online]. Available: https://litecoin.org/.

[6] M.-H. R. Tseng, S. E. Chang and T.-Y. Kuo, "Using blockchain to access cloud services : a case of financial service application," in Federated Conference on Computer Science and Information Systems, 2019.

[7] V. Chang, P. Baudier, H. Zhang, Q. Xu, J. Zhang and M. Arami, "How blockchain can impact financial services – the overview, challenges and recommendations from expert interviewees," Technological Forecasting & Social Change, 2020.

[8] M. Mettler, "Blockchain technology in healthcare, the revolution starts here," in 18th International Conference on e-Health Networking, Applications and Services , 2016.

[9] T.-T. Kuo, H.-E. Kim and L. Ohno-Machado, "Blockchain distributed ledger technologies for biomedical and health care applications," Journal of the American Medical Informatics Association, , 2017.

[10] T. M. Fernandez-Carames and P. Fraga-lamas, "A review on the use of blockchain for the internet of things," IEEE Access, 2018.

[11] H.-N. Dai, Z. Zheng and Y. Zhang, "Blockchain for internet of things: a survey," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8076 - 8094, 2019.

[12] A. Alketbi, D. Q. Nasir and D. M. A. Talib, "Blockchain for government services – use cases,security benefits and challenges," in 15th Learning and Technology Conference (L&T), 2018.

[13] E. Abodei, A. Norta, I. Azogu, C. Udokwu and D. Draheim, "Blockchain technology for enabling transparent and traceable government collaboration in public project processes of developing economies," in Conference on e-Business, e-Services, and e-Society, 21019.

[14] S. Shahab and Z. Allam, "Reducing transaction costs of tradable permit schemes using Blockchain smart contracts," Journal of urban and regional policy, vol. 51, no. 1, pp. 1-7, 2019.

[15] F. Yiannas, "A new era of food transparency powered by blockchain," The MIT Press Journals , vol. 12, no. 1-2, pp. 46-56, 2018.

[16] K. Francisco and D. Swanson, "The supply chain has no clothes: technology adoption of blockchain for supply chain transparency," MDPI logistics, vol. 2, no. 1, 2018.

[17] R. Kamath, "Food traceability on blockchain : Walmart’s pork and mango pilots with IBM," The Journal of The British Blockchain Association, vol. 1, no. 47-53, p. 1, 2018.

[18] Q. Lu and X. Xu, "Adaptable blockchain-based systems: A case study for product traceability," IEEE Software, vol. 34, no. 6, pp. 21-27, 2017.

[19] X. Xu, Q. Lu, Y. Liu, L. Zhu, H. Yao and A. V. Vasilakos, "Designing blockchain-based applications a case study for imported product traceability," Future Generation Computer Systems, vol. 92, pp. 399-406, 2019.

[20] F. Tian, "A supply chain traceability system for food safety based on HACCP, blockchain & internet of things," in International Conference on Service Systems and Service Management, 2017.

[21] L. Kaijun, B. Ya, J. Linbo, F. Han-Chi and I. V. Nieuwenhuyse, "Research on agricultural supply chain system with double chain architecture based on blockchain technology," Future Generation Computer Systems, vol. 86, pp. 641-649, 2018.

[22] F. Longo, L. Nicoletti, A. Padovano, G. d'Atri and M. Forte, "Blockchain-enabled supply chain: an experimental study," Computers & Industrial Engineering, vol. 136, pp. 57-69, 2019.

[23] K. Toyoda, P. T. Mathiopoulos, I. Sasase and T. Ohtsuki, "A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain," IEEE Access, vol. 5, pp. 17465 - 17477, 2017.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)