Vehicle Tracking System using Hyperledger Fabric

Aditya Vishnu Tiwari(2021202029)
Mayur Kumar(2021202028)
Utkarsh Tripathi(2021202007)

Research in Information Security Project Report

INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY
HYDERABAD

Department Name: CSIS University Name: IIIT Hyderabad
Date: 14 November 2022

Abstract

The drastic increase in purchasing power has led to substantial rise in vehicle purchase by the masses in our country. This brings into question the requirement of a major central system that can register, track and update the details and verification of these vehicles in a transparent and honest manner. This motivates the integration of blockchain to create such service that can be used by regional transport offices(RTOs) , Public and Law enforcements in an efficient manner. The distributed and immutable nature of blockchain helps with the above issues.) ...

Keywords Blockchain, Hyperledger, Fabric, smart contracts, asset transfer and ownership

Contents

1 Introduction

Blockchain: Blockchain is a decentralised, trustworthy distributed ledger that consists of a list of chronologically ordered blocks on a peer-to-peer network. Every block is made up of a hash of the previous block, forming a chain. The genesis block is the first block in the blockchain, and the block preceding a given block is referred to as its parent block. A block has a header and a body. The block header is made up of:

- Version: This property specifies the block validation rules.
- Previous block hash: This ensures that the previous block's header cannot be changed without changing the current block's header.
- Timestamp: The current time of block creation.
- Merkle root hash: The Merkle root is calculated using the hash values of all transactions in the block. This is done to ensure that transactions cannot be changed without changing the header.
- Nonce: A nonce is a four-byte unique number that is used only once during communication.

The following are the fundamental components of blockchain architecture:

- Node/Peer: A device (such as a computer) that exists within a blockchain network and has a copy of all transactions.
- Transaction: The exchange of data between two blockchain addresses.
- Block: It is used to store a group of transactions that are distributed among all network nodes.
- Chain: A chronologically ordered series of blocks.
- Miners: Selected nodes perform block verification before adding any transactions to a new block node.
- Consensus: This is a mechanism that allows all of the blockchain network's peers to agree on any transaction in the blockchain.

Hyperledger Fabric: Hyperledger Fabric is designed to serve as a foundation for developing modular applications or solutions. Hyperledger Fabric enables the plug-and-play integration of components such as consensus and membership services. Its modular and versatile design caters to a wide range of industry use cases. It provides a novel approach to consensus that enables performance at scale while maintaining privacy. The key characteristics of Hyperledger Fabric and what sets it apart from other distributed ledger technologies:

- Permissioned architecture
- Highly modular
- Open smart contract model
- Multi-language smart contract support
- Queryable data (key-based queries and JSON queries)

2 Network Model

One of Hyperledger's blockchain initiatives is Hyperledger Fabric. It has a ledger, employs smart contracts, and is a method for participants to manage their transactions, just like other blockchain technologies.

The distinction between Hyperledger Fabric and other blockchain systems is that it is private and permissioned. Members of a Hyperledger Fabric network enrol through a trusted Membership Service Provider, rather than an open permissionless system that enables unknown identities to participate in the network (requiring protocols like "proof of work" to confirm transactions and protect the network) (MSP).

Hyperledger Fabric also allows for the creation of channels, which allow a group of participants to construct a distinct ledger of transactions. This is a very significant choice for networks where some participants may be rivals and do not want every transaction they make — for example, a special pricing they provide to some participants but not others — to be known to all participants. If two participants construct a channel, only those two people have copies of the ledger for that channel.

2.1 Shared Ledger

The ledger subsystem of Hyperledger Fabric consists of two components: the global state and the transaction log. Each participant has a copy of the ledger for each Hyperledger Fabric network to which they are a member.

The world state component specifies the ledger's status at a certain point in time. It is the ledger database. The transaction log component records all transactions that resulted in the current value of the world state; it represents the world state's update history. The ledger is thus a mix of the global state database and the history of transaction logs.

The global state is stored in a replaceable data store on the ledger. This is a LevelDB key-value store database by default. There is no requirement for the transaction log to be pluggable. It merely logs the before and after values of the blockchain network's ledger database.

2.2 Smart Contracts

When an application wants to communicate with the blockchain, it will call a Hyperledger Fabric smart contract written in chaincode. In most situations, chaincode communicates with the ledger's database component, the world state (through queries, for example), rather than the transaction log.

Chaincode may be written in a number of computer languages. Chaincode in Go, Node.js, and Java is currently supported.

2.3 Privacy

Participants in a Business-to-Business (B2B) network may be particularly careful about how much information they provide, depending on the demands of a network. Privacy will not be a major worry for other networks.

Hyperledger Fabric is compatible with both networks where privacy (through channels) is a critical operating need and networks that are relatively open.

2.4 Consensus

Transactions must be recorded in the ledger in the sequence in which they occur, even if they involve separate groups of network members. To do this, the sequence of transactions must be determined, as well as a way for rejecting bad transactions that have been introduced into the ledger in mistake (or maliciously).

2.5 Hyperledger Fabric Model

The fundamental design elements woven into Hyperledger Fabric that enable it to deliver on its promise of providing a complete, yet adaptable, business blockchain solution:

- Assets: Asset definitions allow the network to exchange nearly anything with monetary worth, from entire meals to vintage vehicles to currency futures.
- Chaincode: Chaincode execution is separated from transaction ordering, reducing the needed degrees of trust and verification across node types while maximising network scalability and speed.

- Ledger Features: The immutable, shared ledger contains the whole transaction history for each channel and offers SQL-like query capabilities for fast auditing and dispute settlement.

- Privacy: Channels and private data collectors enable private and secret multilateral transactions that are typically required by competing enterprises and regulated sectors that trade assets on a shared network.

- Security and Membership Services: Permissioned membership creates a trustworthy blockchain network in which users know that all transactions may be identified and traced by authorised regulators and auditors.

- Consensus: A distinct approach to consensus allows the enterprise's required flexibility and scalability.

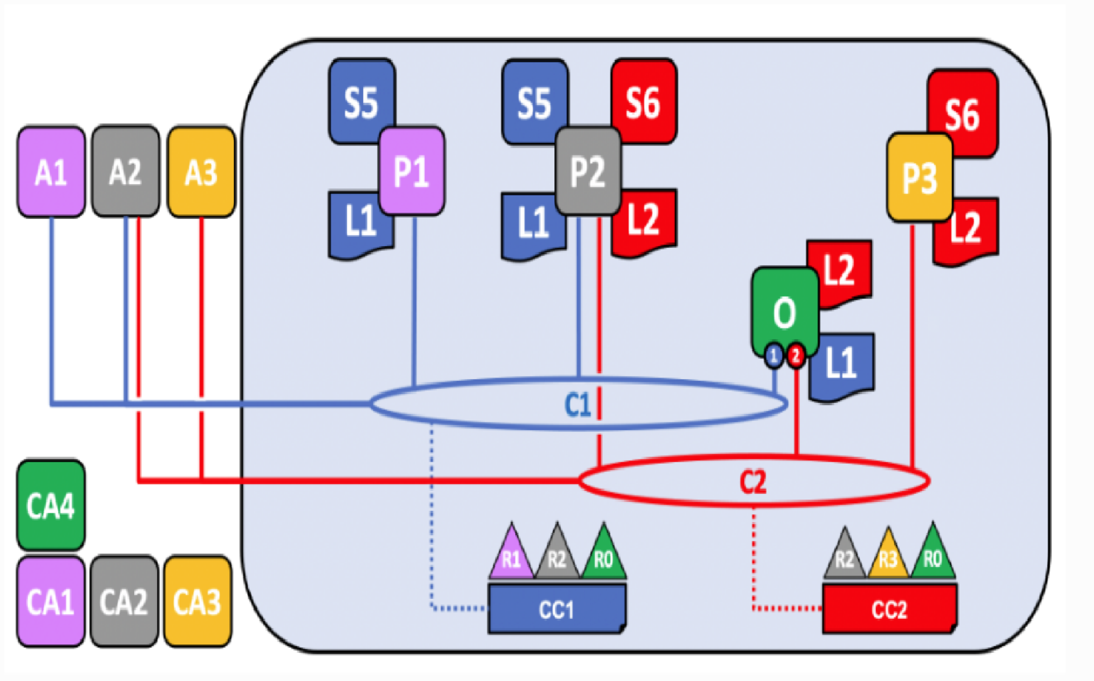


Figure 1: General Hyperledger Model.

Here Rx represents the organizations which are part of the network, Cx refers to the channels inside the network, CCx is the channel configuration which is agreed to by all or a subset of organizations, CA represents the certification authority and every organizations is associated with its own CA, Px refers to the peers/nodes, peers host ledgers and chaincode and are the physical points at which organizations that transact on a channel connect to the channel,one peer can belong to multiple channels.O is the orderer, orderer is in charge of organising transactions into Blocks and distributing them to Anchor. Lx is the copy of the ledger of the channel which is updated with each block. Ax refers to the client application which is associated with its organization, it communicates with the network using smart contracts(Sx).

3 Attack Model

- Sybil Attack : In a Sybil assault, the attacker subverts a peer-to-peer network's reputation system by generating a large number of pseudonymous identities and using them to obtain disproportionately huge influence. Because the MSP is hacked, an attacker might flood the network with bogus identities and take advantage of the majority.

- Boycott Attack : Assume two organisations, Org1 and Org2, are managed by the same MSP. If he regained control of MSP-admin M, he might change the existing policies and refuse to issue certificates to members of Org2, preventing them from connecting to the network.

- Sabotage Attack: Ordering Service Nodes (OSN) are in charge of gathering and organising transactions into blocks. Assume the OSN intends to harm the organisation OrgA. Assume that Pi are the OrgA nodes. O does not include Pi transactions during block consolidation. This would basically disable OrgA.

- DDoS Attack :Fabric becomes more vulnerable to DDoS attacks when the Ordering Service is centralised. To some extent, OS employs CFT protocols like as Kafka, Raft, and others to address this issue. A potential DDoS attack is discussed further down - Validating peers can use the fetchfunction to retrieve blocks from the operating system. If the blockchain network has a high number of peers, the OS may be overwhelmed by constantfetch requests.

4 Comparison between Fabric & Ethereum

	Hyperledger Fabric	Ethereum
Blockchain Type	Permissioned and Private	Permissionless and Private or Public
Objective	Suited for enterprises and B2B applications	Suited for B2C applications
Cryptocurrency	None	Ether
Smart Contract Language	Javascript, GO and Java	Solidity
Consensus Mechanism	Pluggable Consensus Mechanism. Mining not required	PoW consensus Mechanism. Mining is required to reach the consensus
Confidentiality	Confidential Transactions	Transparent
Transactions Per Second (tps)	Greater than 2000tps	Approx 20 tps

5 Our Proposed Scheme

Blockchain is an unchangeable distributed ledger that is shared by everyone on a network. Every participant interacts with the blockchain by utilising a public-private cryptographic key pair. Furthermore, the records recorded on the blockchain are immutable, making them extremely difficult to alter with and therefore offering improved security. A system like Hyperledger Fabric also has functionality for maintaining users and roles, which assist protect and identify owners. The government can use a blockchain's feature set to eliminate the challenges associated with the traditional registration procedure. A distributed ledger can be set up between the buyer, seller, and registration authority. Registration information may be kept in and retrieved from the blockchain, and this information is immutable, which means it cannot be changed by anybody.

6 Implementation

6.1 Structure

There are two stakeholders involved in this project: RTO & General public.

- General public are the people who wish to enroll their vehicles on this network. They need to be explicitly registered on the network to be able to buy/sell the vehicles registered on it.

- RTO has multiple roles. For example, if a user wishes to register himself/herself on the system, then the RTO must validate the identity of the user before adding them to the system. Talking of another use case, suppose a user wishes to register their vehicle on the network. They would first raise a request to the RTO who, in turn, will register the Vehicle on the ledger after validation.

6.2 Workflow

The entire flow of this case study can be divided into three parts:

- User Registration
- Vehicle Registration
- Vehicle Transfer

We will look into each of these subparts individually.

6.2.1 User Registration

This process comprises of the following steps:

- A user with permission to access the network raises a request to the RTO to store their data/credentials on the ledger.
- The request gets stored on the ledger.
- The RTO reads the request and stores the user's data/credentials on the ledger after validating their identity manually.
- Each user has the capability to fill their wallet and put a price on their vehicles. All transactions on this network can be carried out only with this currency. Initially each user has zero currency in their wallet.

6.2.2 Vehicle Registration

This process comprises of the following steps:

- A user added to the vehicle registration system raises a request to the RTO to register their vehicle on the network.
- The request gets stored on the ledger.
- The RTO reads the request and stores the vehicle on the ledger after validating the data present in the request. For example, suppose User X registers himself on the network and raises a request to register his newly purchased car. Now, the RTO needs to verify whether the vehicle is real and whether it actually

belongs to Mr X, before registering it on the ledger.In this process, the vehicle gets only registered on the system. It can be purchased by other registered users only if its owner explicitly wishes to list it for sale.

6.2.3 Vehicle Transfer

This process comprises of the following steps:

- The owner of the vehicle must put the vehicle on sale.
- The buyer of the vehicle must ensure that the amount of currency they have is greater than or equal to the price of the vehicle. If not, then the user must recharge their account.
- If the two criteria above are satisfied, then the ownership of the vehicle changes from the seller to buyer and currency equal to the price of the vehicle is transferred from the buyer's account to the seller's account.

6.3 Assets on the ledger

- General Public: Each user's data/credentials, such as name, email Id, Aadhar number, etc., need to be captured before they can be allowed to buy/sell vehicles on the network. The credentials of each user are stored as states on the ledger.
- Requests: The buyers need to raise a request to the RTO in order to register themselves on the network or to buy a vehicle on the ledger. These requests get stored on the ledger.
- Vehicle: Vehicles owned by vehicle owners registered on the network are stored as assets on the ledger.

6.4 Capabilities of Stakeholders

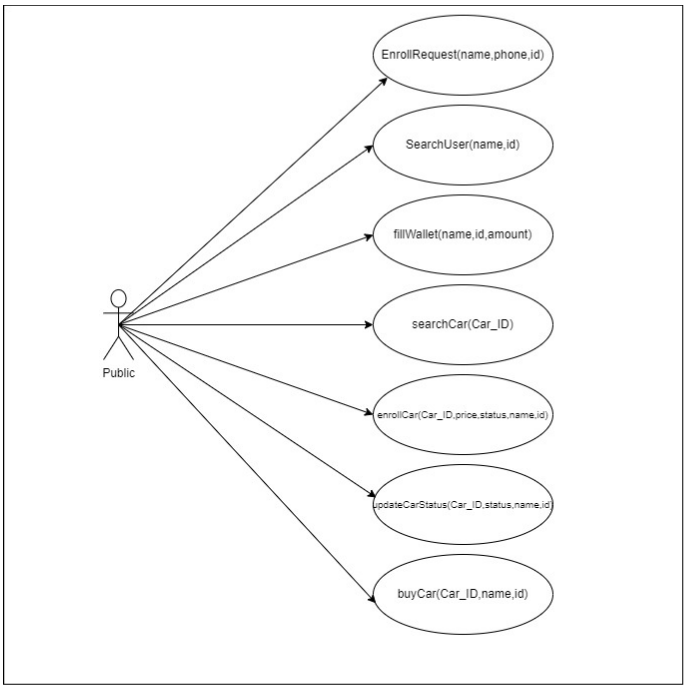

EnrollRequest(name,phone,id)
SearchUser(name,id)
fillWallet(name,id,amount)
searchCar(Car_ID)
enrollCar(Car_ID,price,status,name,id)
updateCarStatus(Car_ID,status,name,id)
buyCar(Car_ID,name,id)
Public

Figure 2: Public Capabilities

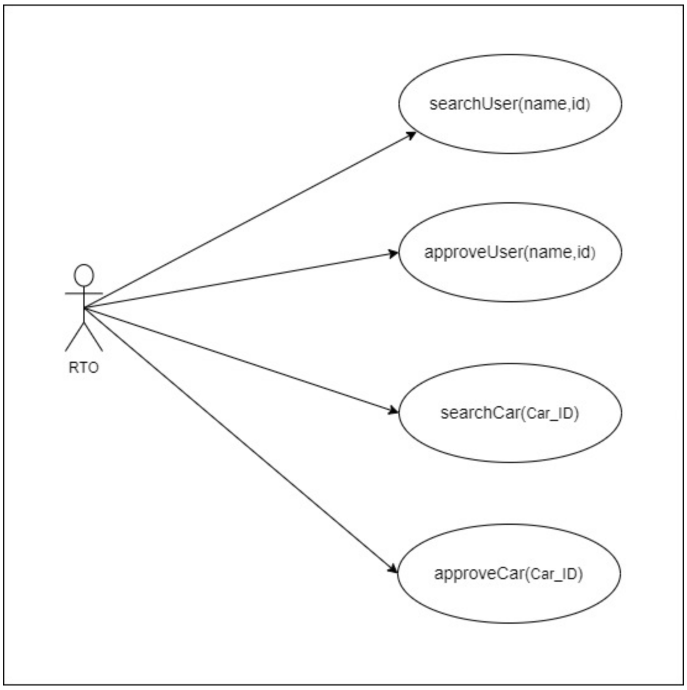
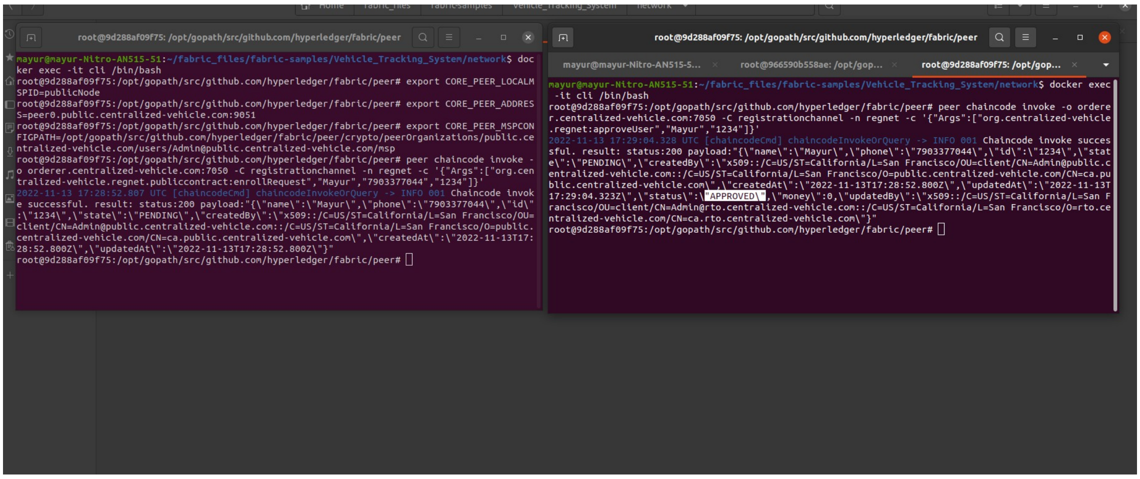


Figure 3: RTO Capabilities

6.5 Output



root@9d288af09f75: /opt/gopath/src/github.com/hyperledger/fabric/peer

```
mayur@mayur-Nitro-AN515-51:~/fabric_files/fabric-samples/Vehicle_Tracking_System/network$ doc
ker exec -it cli /bin/bash
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer# export CORE_PEER_LOCALM
SPID=publicNode
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer# export CORE_PEER_ADDRES
S=peer0.public.centralized-vehicle.com:9051
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer# export CORE_PEER_MSPCON
FIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOrganizations/public.ce
ntralized-vehicle.com/users/Admin@public.centralized-vehicle.com/msp
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -
o orderer.centralized-vehicle.com:7050 -C registrationchannel -n regnet -c '{"Args":["org.cen
tralized-vehicle.regnet.publiccontract:enrollRequest","Mayur","7903377044","1234"]}'
2022-11-13 17:28:52.807 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invok
e successful. result: status:200 payload:"{\"name\":\"Mayur\",\"phone\":\"7903377044\",\"id\"
:\"1234\",\"state\":\"PENDING\",\"createdBy\":\"x509::/C=US/ST=California/L=San Francisco/OU=
client/CN=Admin@public.centralized-vehicle.com::/C=US/ST=California/L=San Francisco/O=public.
centralized-vehicle.com/CN=ca.public.centralized-vehicle.com\",\"createdAt\":\"2022-11-13T17:
28:52.800Z\",\"updatedAt\":\"2022-11-13T17:28:52.800Z\"}"
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

root@9d288af09f75: /opt/gopath/src/github.com/hyperledger/fabric/peer

mayur@mayur-Nitro-AN515-5...
root@966590b558ae: /opt/gop...
root@9d288af09f75: /opt/gop...

```
mayur@mayur-Nitro-AN515-51:~/fabric_files/fabric-samples/Vehicle_Tracking_System/network$ docker exec
 -it cli /bin/bash
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o ordere
r.centralized-vehicle.com:7050 -C registrationchannel -n regnet -c '{"Args":["org.centralized-vehicle
.regnet:approveUser","Mayur","1234"]}'
2022-11-13 17:29:04.328 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001 Chaincode invoke succes
sful. result: status:200 payload:"{\"name\":\"Mayur\",\"phone\":\"7903377044\",\"id\":\"1234\",\"stat
e\":\"PENDING\",\"createdBy\":\"x509::/C=US/ST=California/L=San Francisco/OU=client/CN=Admin@public.c
entralized-vehicle.com::/C=US/ST=California/L=San Francisco/O=public.centralized-vehicle.com/CN=ca.pu
blic.centralized-vehicle.com\",\"createdAt\":\"2022-11-13T17:28:52.800Z\",\"updatedAt\":\"2022-11-13T
17:29:04.323Z\",\"status\":\"APPROVED\",\"money\":0,\"updatedBy\":\"x509::/C=US/ST=California/L=San F
rancisco/OU=client/CN=Admin@rto.centralized-vehicle.com::/C=US/ST=California/L=San Francisco/O=rto.ce
ntralized-vehicle.com/CN=ca.rto.centralized-vehicle.com\"}"
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

Figure 4: Successful invocation of Transactions

7 Contribution

Roll Number	Name	Contribution
2021202007	Utkarsh Tripathi	Implementation & Report Documentation
2021202028	Mayur Kumar	Implementation & Report Documentation
2021202029	Aditya V. Tiwari	Implementation, Literature Review & Presentation

8 Future Works and Improvements

- Many modifications may be made to improve the solution and turn it into a production-ready application. Even while blockchain and fabric give a high level of security by default in their structure and idea, they must solve the security problems outlined above in order to successfully secure vehicular information. As the network grows, more organisations and peers connect to the channel, and in order to manage a large number of transaction requests and approvals, numerous ordering peers are required to speed up the process. Apache Kafka, a distributed event streaming platform open source, is a promising technology for managing many ordering nodes.

- A frontend application needs to be created for the users to have an easy and interactive interface to connect with the blockchain

- A police verification system can also be added which will be useful in case of thefts or similar mishappenings.

- Another subsystem can be added to track the on road vehicle life time so that after the threshold the vehicle data becomes useless and further transactions won't be possible.

9 Conclusion

Hyperledger Fabric is a potential blockchain architecture that includes ideas like as policies, smart contracts, and the supply of secure identities, all of which help to keep information secure and under control. It allows the car registration system to communicate with many participants. Users can simply trace their history. Sellers will no longer need to carry vehicle records, and digital records will improve significantly. A car registration scenario falls within the category of private and closed blockchain, and this solution may effectively infer that it is an encouraging foundation for this type of blockchain. It offers a dependable and safe alternative for maintaining these documents.

10 References

- DifferentTypesofHyperledgerTechnologies.https://medium.com/@vinshublockcluster/different-types-of-hyperledger-technologies-929a67c98c32.
- Downloadfabricsamplesprocedure.In:url:https://hyperledgerΘfabric.readthedocs.io/en/release-2.2/test_network.html#before-you-begin.
- https://hyperledger-fabric.readthedocs.io/en/releaseΘ2.2/ledger/ledger.html
- SmartContractsandChaincode.https://hyperledger-fabric.readthedocs.io/en/release-2.2/smartcontract/smartcontract.html.
- https://www.researchgate.net/publication/337276742_Ripping_the_Fabric_Attacks_and_Mitigations_on_Hyperledger_Fabric