INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Blockchain Solution to Vehicle Registration System using Hyperledger Fabric

Project presentation

Aditya Vishnu Tiwari(2021202029),

Utkarsh Tripathi (2021202007)

09 May 2023

Content

Introduction

- Solution for vehicle registaration and ownership transfer.
- A transparent, immutable, shared and secure user and vehicular data ledger can help law enforcement, users and Transport Offices alike.
- Ensuring data integrity, confidentiality and privacy of users while sharing the data.

What Is Blockchain?

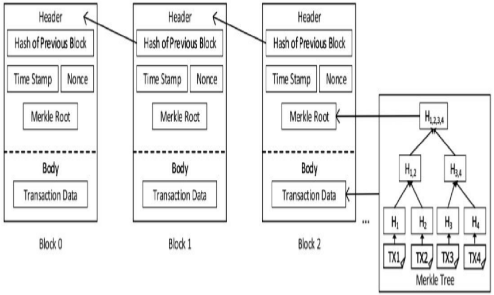


Figure: Block Structure

What is Blockchain?

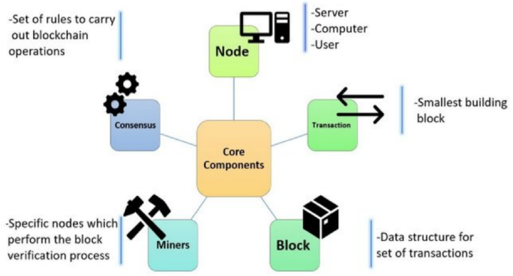

-Set of rules to carry out blockchain operations
Node
-Server
-Computer
-User
Consensus
Transaction
-Smallest building block
Core Components
-Specific nodes which perform the block verification process
Miners
Block
-Data structure for set of transactions

Figure: Blockchain core components

Types Of Blockchain

- Public Blockchain (Permissionless):
  - Everyone can access the public blockchain and participate in the transactions.
  - Examples are Bitcoin, Litecoin, and Ethereum
- Private Blockchain (Permissioned):
  - Restrictions on who can join the network and who can participate in the transactions.
  - Used by organizations or companies for its internal usage.
  - Example, Hyperledger Fabric.

Hyperledger Fabric Architecture

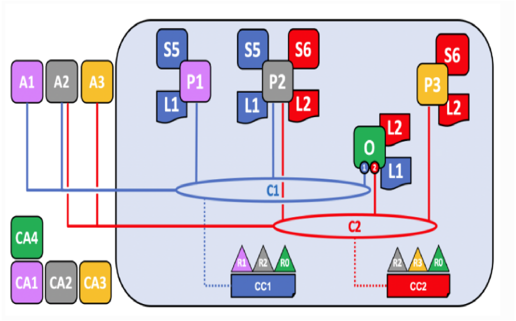


Figure: Hyperledger Fabric Architecture

Why Blockchain and Fabric?

- Blockchain stores data cryptographically secure
- Authentication and authorization - fabric provides CA and MSP components which provide secure indetities like private key and certificates and validation done when make connection to network.
- Confidentiality - fabric is a permisionned blockchain framework.
- Availability - distributed nature of blockchain makes data available to all permissioned systems.
- Data integrity - blockchain records are immutable
- Scalablity - New organization, peers and users with different roles.
- Pluggable modules

Implementation

- **Stakeholders**
  - RTO
  - General public.
- **Logical Flow**
  - User Registration:
    - A user raises a request to the RTO to store their data/credentials on the ledger.
    - request gets stored on the ledger.
    - RTO reads the request and stores the user’s data/credentials on the ledger after validating.
    - Adding balance to the wallet.
  - vehicle Registration:
    - A user raises a request to the RTO to register their vehicle on the network.
    - request gets stored on the ledger.
    - RTO reads the request and stores the vehicle on the ledger after validating

Implementation

- vehicle Transfer:
  - owner of the vehicle must put the vehicle on sale
  - 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 the two criteria above are satisfied, then the ownership of the vehicle changes from the seller to buyer & currency equal to the price of the vehicle is transferred

- **Assets**
  - 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.
  - 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.
  - Vehicle:Vehicles owned by vehicle owners registered on the network are stored as assets on the ledger.

Our Architecture

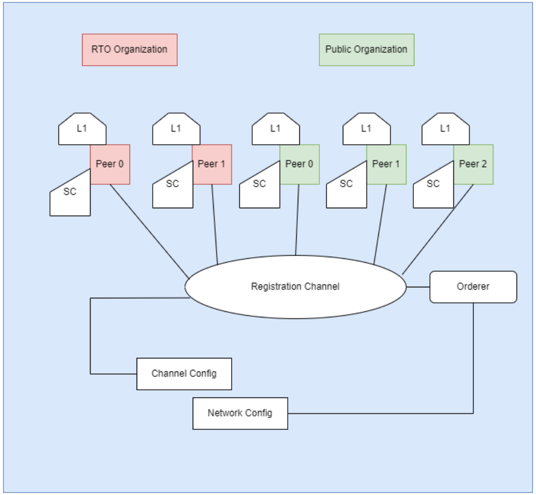


Figure: Our Architecture

Use Cases

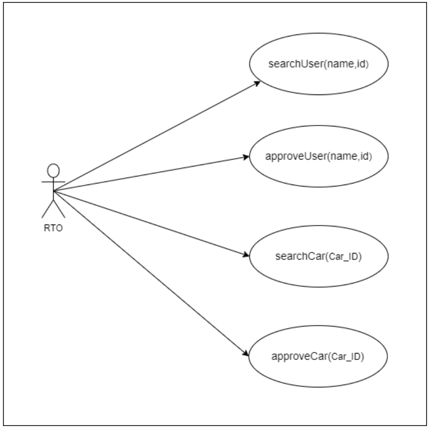


Figure: Use Cases

Use Cases

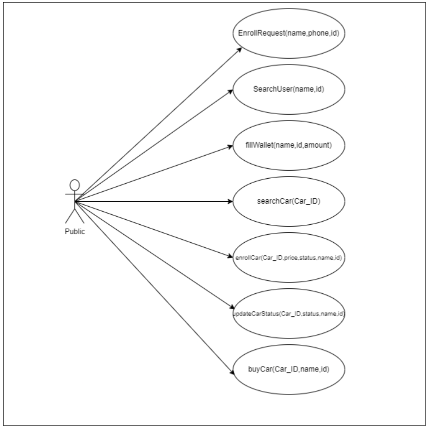
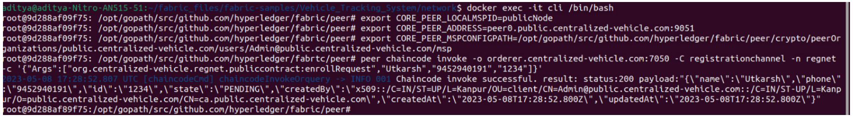

EnrollRequest(name,phone,id)
SearchUser(name,id)
fillWallet(name,id,amount)
searchCar(Car_ID)
enrollCar(Car_ID,price,status,name,id)
updateCarStatus(Car_ID,status,name,id)
buyCar(Car_ID,name,id)
Public

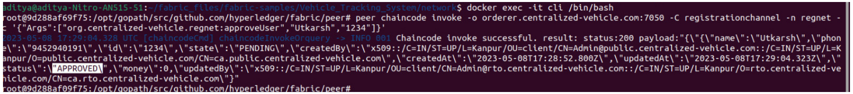
Figure: Use Cases

Output



```
aditya@aditya-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# export CORE_PEER_LOCALMSPID=publicNode
root@9d288af09f75: /opt/gopath/src/github.com/hyperledger/fabric/peer# export CORE_PEER_ADDRESS=peer0.public.centralized-vehicle.com:9051
root@9d288af09f75: /opt/gopath/src/github.com/hyperledger/fabric/peer# export CORE_PEER_MSPCONFIGPATH=/opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/peerOr
ganizations/public.centralized-vehicle.com/users/Admin@public.centralized-vehicle.com/msp
root@9d288af09f75: /opt/gopath/src/github.com/hyperledger/fabrtc/peer# peer chaincode invoke -o orderer.centralized-vehicle.com:7050 -C registrationchannel -n regnet
-c '{"Args":["org.centralized-vehicle.regnet.publiccontract:enrollRequest","Utkarsh","9452940191","1234"]}'
2023-05-08 17:28:52.807 UTC [chaincodeCmd] chaincodeInvokeOrquery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"{\"name\":\"Utkarsh\",\"phone\"
:\"9452940191\",\"id\":\"1234\",\"state\":\"PENDING\",\"createdBy\":\"x509::/C=IN/ST=UP/L=Kanpur/OU=client/CN=Admin@public.centralized-vehicle.com::/C=IN/ST-UP/L=Kanp
ur/O=public.centralized-vehicle.com/CN=ca.public.centralized-vehicle.com\",\"createdAt\":\"2023-05-08T17:28:52.800Z\",\"updatedAt\":\"2023-05-08T17:28:52.800Z\"}"
root@9d288af89f75:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

Figure: Transaction invocation



```
aditya@aditya-Nitro-AN515-51:~/fabric_files/fabric-samples/Vehicle_Tracking_System/network$ docker exec -it cli /bin/bash
root@9d288af69f75:/opt/gopath/src/github.com/hyperledger/fabric/peer# peer chaincode invoke -o orderer.centralized-vehicle.com:7050 -C registrationchannel -n regnet -
c '{"Args":["org.centralized-vehicle.regnet:approveUser","Utkarsh","1234"]}'
2023-05-08 17:29:04.328 UTC [chaincodeCmd] chaincodeInvokeOrquery -> INFO 001 Chaincode invoke successful. result: status:200 payload:"{\"{\"name\":\"Utkarsh\",\"phon
e\":\"9452940191\",\"id\":\"1234\",\"state\":\"PENDING\",\"createdBy\":\"x509::/C=IN/ST=UP/L=Kanpur/OU=client/CN=Admin@public.centralized-vehicle.com::/C=IN/ST=UP/L=K
anpur/O=public.centralized-vehicle.com/CN=ca.public.centralized-vehicle.com\",\"createdAt\":\"2023-05-08T17:28:52.800Z\",\"updatedAt\":\"2023-05-08T17:29:04.323Z\",\"
status\":\"APPROVED\",\"money\":0,\"updatedBy\":\"x509::/C=IN/ST=UP/L=Kanpur/OU=client/CN=Admin@rto.centralized-vehicle.com::/C=IN/ST=UP/L=Kanpur/O=rto.centralized-ve
hicle.com/CN=ca.rto.centralized-vehicle.com\"}"
root@9d288af09f75:/opt/gopath/src/github.com/hyperledger/fabric/peer#
```

Figure: Successful invocation of transactions

Pros and Cons of using hyperledger Fabric

- **Pros:**
  - Fabric architecture allows to add plugins for the identity management and consensus algorithm.
  - Confidentiality and security of data can be achieved through MSP.
  - Performance is optimized, since mining is not required.
  - Creation of a private channel for only a few participants among a large blockchain network.
- **Cons:**
  - The architecture of hyperledger fabric is quite complex.
  - It is not a network fault tolerant.
  - Limited database support.

Attack Models

- **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.
- **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.Validating peers can use the fetch function to retrieve blocks from the ordering system. If the blockchain network has a high number of peers, the OS may be overwhelmed by constant fetch requests.

Conclusion

- Hyperledger fabric is a promising blockchain framework comes with policies, smart contracts and provision of secure identities.
- Enable the Vehicle registration scenario interoperable among multiple users. organizartions
- A promising framework for private and closed blockchain scenarios
- Provide reliable and secure solution in managing vehicle records

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.

References:

- Different Types of Hyperledger Technologies. https://medium.com/@vinshublockcluster/different-types-of-hyperledger-technologies-929a67c98c32.
- “Download fabric samples procedure.” In: url:https://hyperledgerfabric . readthedocs . io / en / release - 2 . 2 / testnetwork . html before-you-begin
- Ledger. https://hyperledger-fabric.readthedocs.io/en/release2.2/ledger/ledger.html.
- “Prerequisites before start test network of hyperledger fabric.” In: url: https : / / hyperledger - fabric . readthedocs . io / en / release - 2 . 2 / prereqs.htmlprerequisites.
- Smart Contracts and Chaincode. https://hyperledger-fabric.readthedocs. io/en/release-2.2/smartcontract/smartcontract.html.
- Ripping the Fabric: Attacks and Mitigations on Hyperledger Fabric: https://www.researchgate.net/publication/337276742RippingtheFabricAttacksandMitigatio

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

THANK YOU