24 September 2023

CS61065 Theory and Applications of Blockchain

Assignment 2: Hyperledger Fabric

Date of submission: October 10, 2023 EOD

To be done groups of maximum 4 members.

Part A — 20 marks

Deploy a blockchain network with two organizations on the Hyperledger Fabric platform to support private book-keeping for the organizations. This part involves the usage of the concept of private data on the Hyperledger Fabric network.

Reference:
https://hyperledger-fabric.readthedocs.io/en/latest/private-data/private-data.html
https://hyperledger-fabric.readthedocs.io/en/release-2.5/private-data-arch.html

Note: If the data is not kept private the evaluation will be done for 70% of the marks

Data to be stored on the blockchain:

- Account balance for each organization (**Public data**)
- Inventory for each organization storing the following details for each item: Item name, number of items and the price for each item (**Private data**)

The chain-code should support the following functions: **10 marks**

- Add money to account
  `async AddBalance()`
  - send the details as transient data
  - Add balance to the respective account of the client's organization
- Add item to inventory
  `async AddItem()`
  - send the details as transient data
  - store the details in the private data for the org
- Get balance
  `async GetBalance()`
  - read private data for the org and return the details
- Get item details
  `async GetItem()`
  - read private data for the org and return the details

Other requirements:

- One organization should not be able to access/figure out the inventory information of the other organization

Create a terminal based application for each organization to connect to the network and support the following: **10 marks**

- Take input command as “ADD_MONEY” or “ADD_ITEM” or “QUERY_BALANCE” or “GET_ITEM”
- Take appropriate arguments if needed and invoke the appropriate smart contract.

Part B 30 marks

Support trade of goods on the blockchain network. Involves the usage of events and events listeners and is a continuation of the previous part.

Reference:
https://github.com/hyperledger/fabric-samples/tree/main/asset-transfer-events
https://hyperledger.github.io/fabric-sdk-node/release-2.2/module-fabric-network.Network.html
https://hyperledger.github.io/fabric-sdk-node/release-2.2/module-fabric-network.Contract.html

Data to be stored on the chain:

- Marketplace: A collection of all the items which the organizations want to sell. This is **public data**. Organizations can invoke smart contracts to add to the marketplace from their respective inventories.

Chain code should support all the functionalities from the previous part and the following additional functionalities:

- Add item to marketplace 5 marks

```
async AddToMarket(item, price)
    - check private data of the organization to ensure that the item is present in the inventory
    - add to marketplace and remove from the inventory
```

- Buy an item from marketplace 5 marks

```
async BuyFromMarket(item)
    - check private data to ensure if sufficient balance to buy the item
    - Deduct from the balance of the buyer and add to the balance of the seller
    - Remove item from the marketplace
```

- Get all items on the marketplace 5 marks

```
async GetItemsInMarket()
    - Return all the items in the marketplace along with their price
```

Application should support all the functionalities from the previous part and the following additional functionalities: 5 marks

- The user should be able to enlist an item to the marketplace using ENLIST_ITEM command
- The user should be able to see all the items in the marketplace using ALL_ITEMS command

Additionally, 10 marks

- Maintain a local list of required items for each client application. The user must be able to add items to the list using the WISHLIST command
- Listen to the event in case there is a new item added to the marketplace. If the item is present in the list of required items, trigger a smart contract to buy the item.

Submission:

Submit a zip file containing the chain code and application code for both the parts in separate folders along with any other config files required to deploy your code and a README. Clearly mention the group details and instructions on how to run your code. The zip file should be named <Roll_no.>_Assignment_2.zip