ELECTRONICS SUPPLY CHAIN MANAGEMENT

- **About The Project**

This Project is about maintaining the transparency between customer and E-com websites to ensure same product delivered to the end user or not by maintaining the ledger of Appliance or product data between Manufacturers and E-com websites and Delivery partners

This Project will be useful for tracking appliance or product life cycle between all participant to avoid gap between E-com websites and customers

- **Project Flow**

Manufacturer Will Release the product and add barcode to the product will contain below information of product

* Product Name

* Product Model Number

*MRP

* Date Of Manufacturing

* Manufacturer Name

The information added by the manufacturer gets stored on the block chain, providing transparency to the supply chain to other stakeholders. Manufacturer will send product to E-commerce company with the help of Hash ID stored on the block chain, E-Commerce company can verify the origin of Product after collecting them from the logistics service Providers E-Commerce Company validate the received Product and sign the transaction digitally, which is then added to the block chain. E-Commerce Company can get the product details like product type, model number in block

When Customer orders the product through E-Commerce website based on product details the order has to handover to Product Order logistics company for delivery with order and product details

* E-commerce can get the Product/Order details which can be traced back to know its origin, using the hash ID saved on the block chain

* Suppose any illegal product delivery happens and “product order” company to deliver counterfeit product with fake Order ID to Customer. In that case, the transaction is considered invalid because of the false information added to the Order.

* Also, unauthorized individuals cannot carry out transactions in the Electronics supply chain ecosystem without a valid private key

* Customer can validate the product received by scan the bar code received in product cover with E-commerce app

* Customer can ensure ordered product and delivered product both are matching or not

* The hash ID linked to the Barcode would fetch information from the blockchain for product quality to avoid false product E-commerce companies, in turn, send approved products to manufacturers.

* After that, the finished goods are delivered to a central warehouse, separated into packages, and distributed to local warehouses

* The repackaged Products will be handled by logistics to deliver to Ecommerce warehouses in different cities, based on the Customer order E commerce company will hand over product to “Product orders”

* Product order company will deliver the product to customer, customer/Ecommerce and track product based on deliver details and product details in website

- **Project Contents**

This Project will contain the 3 peers and Orderer will connect through a secured network layer with Fabric-ca certificate authentication

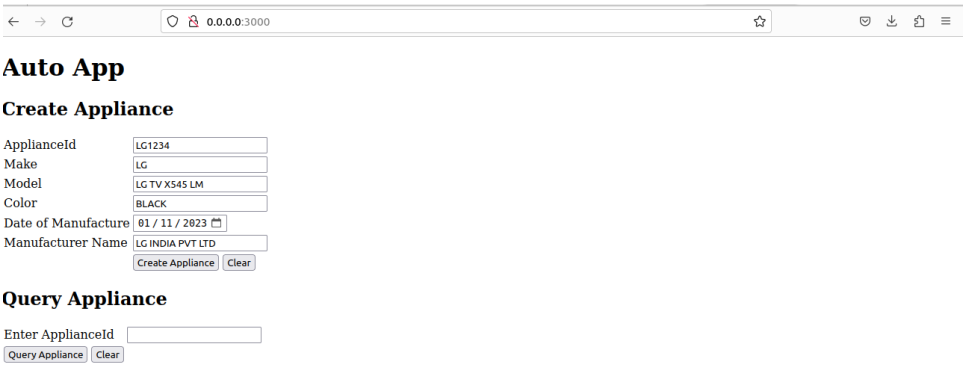
Step1: Up the network and enroll the peers and orderer with secured communication by generating the ca TLS signer certificates to exchange secure connection and data exchange

Step2: Deploy the chain code by invoking the business data

Step3 : insert product details and query from other peers

Step4: invoke the different operations of data

Step5: Create the Client app to enroll the data to register the product

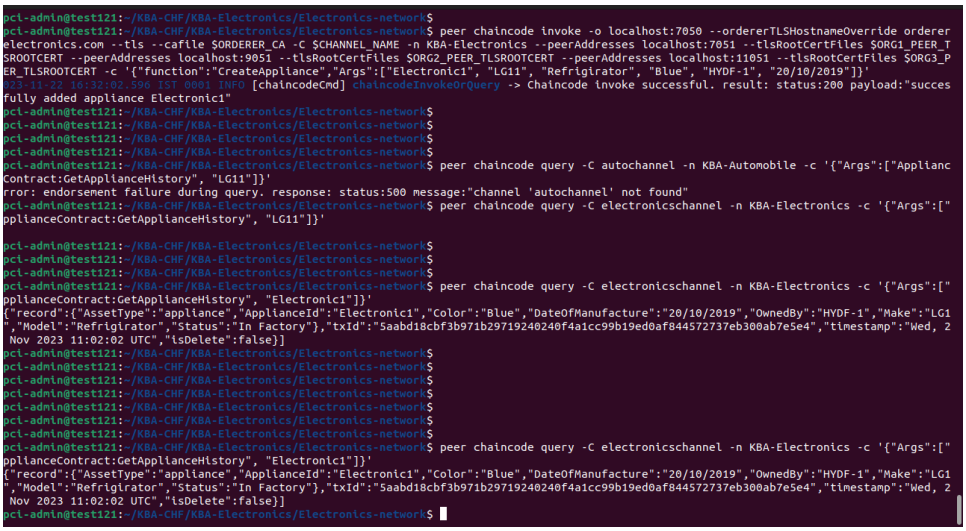


Step6: we can perform and verify the data exchange between client and server

```
rwxrwxr-x  2 npci-admin npci-admin   4096 Nov 21 15:34  Client/
rw-rw-r--  1 npci-admin npci-admin  18330 Nov 21 10:38  commandsElectronicsCommands.txt
rw-rw-r--  1 npci-admin npci-admin    431 Nov 10 11:56  Commands.txt
rw-rw-r--  1 npci-admin npci-admin   4966 Nov 23 09:11  CreateOrderDetails
rwxrwxr-x  7 npci-admin npci-admin   4096 Nov 22 09:40  Electronics-network/
rwxrwxr-x  2 npci-admin npci-admin   4096 Nov 10 11:56  Events/
rwxrwxr-x  3 npci-admin npci-admin   4096 Nov 10 11:56  Gin-App/
rw-rw-r--  1 npci-admin npci-admin 256464 Nov 10 11:56  Project_Document_Template.pdf
rw-rw-r--  1 npci-admin npci-admin   3290 Nov 10 11:56  ProjectGuidelines.md
rwxrwxr-x  4 npci-admin npci-admin   4096 Nov 10 11:56  Simple-Dapp/
pci-admin@test121:~/KBA-CHF/KBA-Electronics$ cd Simple-Dapp/
pci-admin@test121:~/KBA-CHF/KBA-Electronics/Simple-Dapp$ go run .
GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

GIN-debug] GET    /public/*filepath         --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (3 handlers)
GIN-debug] HEAD   /public/*filepath         --> github.com/gin-gonic/gin.(*RouterGroup).createStaticHandler.func1 (3 handlers)
GIN-debug] Loaded HTML Templates (2):
       -
       - index.html

GIN-debug] GET    /                         --> main.main.func1 (3 handlers)
GIN-debug] POST   /api/apliance             --> main.main.func2 (3 handlers)
GIN-debug] GET    /api/apliance/:id         --> main.main.func3 (3 handlers)
GIN-debug] GET    /get-event                --> main.main.func4 (3 handlers)
GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
lease check https://pkg.go.dev/github.com/gin-gonic/gin#readme-don-t-trust-all-proxies for details.
GIN-debug] Listening and serving HTTP on 0.0.0.0:3000

** Start Chaincode event listening
GIN] 2023/11/23 - 09:52:51 | 200 |     249.201µs |       127.0.0.1 | GET      "/"
GIN] 2023/11/23 - 09:52:51 | 404 |     708.161µs |       127.0.0.1 | GET      "/public/styles/style.css"
GIN] 2023/11/23 - 09:52:51 | 304 |     124.685µs |       127.0.0.1 | GET      "/public/scripts/index.js"
GIN] 2023/11/23 - 09:52:51 | 404 |         623ns |       127.0.0.1 | GET      "/favicon.ico"
```

```
** Start Chaincode event listening
eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"Refrigirator"}

eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"SS121"}

Csignal: interrupt
pci-admin@test121:~/KBA-CHF/KBA-Electronics/Events$ go run .

** Start Chaincode event listening
eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"ss1"}

eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"LG1"}

eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"HR1"}

eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"One1"}

Csignal: interrupt
pci-admin@test121:~/KBA-CHF/KBA-Electronics/Events$ go run .

** Start Chaincode event listening
Csignal: interrupt
pci-admin@test121:~/KBA-CHF/KBA-Electronics/Events$ go run .

** Start Chaincode event listening
eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"JBL1"}

eceived Chaincode Event: CreateAppliance
Data: {"Type":"Appliance creation","Model":"Refrigirator"}
```

We can invoke multiple operation to track the product data history and sort of data, pagination data and history and all order and product details



- **Functions Used:**

Various functions used to insert and retrieve data based on organization requirement (we can find business contract data in contracts folder)

CreateOrder()

ReadOrder()

GetPrivateData()

DeleteOrder()

Range

GetStateByRange(startkey,endkey)

History

GetHistoryForKey(ID)

Pagination

GetQueryResultWithPagination(QueryString,pageSize,bookmark)

Matching Orders

GetPrivateDataQueryResult(collectionName,querystring)

Match order and appliances

GetPrivateData(collectionName,ID)