DAFTAR PUSTAKA

[1] Anak Agung Gde Agung, Rini Handayani, “Blockhain for smartgrid,” Elsavier Inc, 2020.

[2] Imran Bashir, 2017. "Mastering Blockchain Distributed ledgers, decentralization and smart contracts explained," Packt Publishing Ltd, Birmingham,United Kingdom.

[3] Elad Elrom. *"The Blockchain Developer,"* Apess : New York.USA.2019.

[4] Chris Dannen. 2017 *"Introducing Ethereum and Solidity Foundation of Cryptocurreny and Blockchain programmer for beginner,"* Apress : Brooklyn, New York, USA.

[5] Ayman Esmat, Martijn de Vos, Yashar Ghiassi-Farrokhfal, Peter Palensky and Dikc Epema ."A novel decentralized platform for peer to peer energy trading market with blockchain technology,"Elsavier Inc.2021.

[6] Gavin Zheng, Longxiang Gao, Liqun Huang and Jian Guang. "Ethereum Smart Contract Development in Solidity," Springer Nature Singapore Pte Ltd.2021.

[7] Xiaoqiong Xu, Athanasios V. Vasilakos, Gang Sun, Long Luo, Huilong Cao, Hongfang Yu*," Latency performance modeling and analysis for hyperledger fabric blockchain network"*, Elsavier Inc, 2021.

[8] Dr.Perry Xiao, " Practical Java Programming for IoT, AI, and Blockchain," John Wiley & Sons, Inc, Indianapolis, Indiana,Canada, 2019.

[9] Sever Spânulescu, "ESP32 programming for the Internet of Things Second edition," 2020.

[10] Neil Cameron, "Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices," Edinburgh, UK, 2021.

[11] Nishith Pathak and Anurag Bhandari, " IoT, AI, and Blockchain for .NET," Apress, India, 2018.

[12] Mohammad Ali Saberi, Mehdi Adda, Hamid Mcheick. *" system based on Blockchain, IPFS and ABAC Break-Glass Conceptual Model for Distributed EHR management system based on Blockchain, IPFS and ABAC,"* Elsavier Inc.2021.

[13] Ritesh Modi. " Solidity Programming Essentials," Packt Publishing, 2018.

[14] Fatimah Hussain Al-Naji , Rachid Zagrouba, *" CAB-IoT: Continuous authentication architecture based on Blockchain for internet of things,"* Elsavier Inc, 2020.

[15] Muhammad Usmarr, Usman Qamar, "Secure Electronic Medical Records

Storage and Sharing Using Blockchain Technology,"Elsavier Inc, 2019

[16] Alexandru-Gabriel Cristea, Lenuta Alboaie, Andrei Panu, Vlad Radulescu, "Offline but still connected with IPFS based communication," Elsavier Inc, 2020.

[17] M. H. Rashid and H. M. Rashid. *" Design and implementation of an open-Source IoT and blockchain-based peer-to-peer energy trading platform using ESP32-S2, Node-Red and, MQTT protocol,"* Boca Raton: CRC Press.2006.

[18] Jonathan Oxer and Hugh Blemings, " Practical Arduino Cool Projects for Open Source Hardware,"Apress, 2009.

[19] https://firebase.google.com/docs/database, diakses pada maret 2021.

[20] Rodrigo da Rosa Righi, Antonio Marcos Alberti, *" Blockchain Trechnology For Industry 4.0:Secure, Distributed, and Trusted Industry Envirotment,"* Springer,2020.

[21] Houshyar Honar Pajooh, Mohammad A. Rashid, Fakhrul Alam and Serge Demidenko, “Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed”, Sensors 2022, 22, 4868.https://doi.org/10.3390/s22134868.

[22] https://developer.mozilla.org/en-US/docs/Glossary/JavaScript, diakses pada maret 2021.

[23] https://socket.io/, diakses pada maret 2021.

[24] Ulla Kirch-Prinz, Peter Prinz,” A Complete Guide to Programming in C++”, jones and bartlett publishers, Massachusetts 2002.

[25] Mohamed Fezari, Ali Al Dahoud,” Integrated Development Environment "IDE" For Arduino”, https://www.researchgate.net/publication/328615543

[26] Akshit J. Dhruv, Reema Patel and Nishant Doshi,” Python: The Most Advanced Programming Language for Computer Science Applications”, DOI: 10.5220/0010307900003051 In Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020), pages 292-299 ISBN: 978-989-758-501-2.

[27] Li You and Hui Sun, “Research and Design of Docker Technology Based Authority Management System”, Hindawi Computational Intelligence and Neuroscience Volume 2022, Article ID 5325694, 8 pages https://doi.org/10.1155/2022/5325694.

[28] Houshyar Honar Pajooh, Mohammad A. Rashid, Fakhrul Alam and Serge Demidenko,“Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed,” Sensors 2022, 22, 4868.https://doi.org/10.3390/s22134868

LAMPIRAN SPESIFIKASI CPU

System Information

Time of this report: 8/2/2022, 09:47:12

Machine name: **********

Machine Id: {4B34A097-EDBE-4C5E-ACA7-F596F2D8EEA7}

Operating System: Windows 11 Home 64-bit (10.0, Build 22621) (22621.ni_release.220506-1250)

Language: English (Regional Setting: English)

System Manufacturer: ASUSTeK COMPUTER INC.

System Model: AFA506IV_FA506IV

BIOS: FA506IV.319 (type: UEFI)

Processor: AMD Ryzen 7 4800H with Radeon Graphics (16 CPUs), ~2.9GHz

Memory: 16384MB RAM

Available OS Memory: 15790MB RAM

Page File: 12067MB used, 7306MB available

Windows Dir: C:\WINDOWS

DirectX Version: DirectX 12

DX Setup Parameters: Not found

User DPI Setting: 120 DPI (125 percent)

System DPI Setting: 120 DPI (125 percent)

DWM DPI Scaling: UnKnown

Miracast: Available, with HDCP

Microsoft Graphics Hybrid: Supported

DirectX Database Version: 1.4.4

DxDiag Version: 10.00.22621.0001 64bit Unicode

LAMPIRAN CODE

```
# CREATE CERTIFICATE
createcertificatesForOrg1() {
  echo
  echo "Enroll the CA admin"
  echo
  mkdir -p ../crypto-config/peerOrganizations/organization1/
  export FABRIC_CA_CLIENT_HOME=${PWD}/../crypto-
config/peerOrganizations/organization1/


  fabric-ca-client enroll -u https://admin:adminpw@localhost:1001 --
caname ca-organization1 --tls.certfiles ${PWD}/fabric-ca/org1/tls-
cert.pem


  echo 'NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/localhost-1001-ca-organization1.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/localhost-1001-ca-organization1.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/localhost-1001-ca-organization1.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/localhost-1001-ca-organization1.pem
    OrganizationalUnitIdentifier: orderer' >${PWD}/../crypto-
config/peerOrganizations/organization1/msp/config.yaml

  echo
  echo "Register peer0"
  echo
  fabric-ca-client register --caname ca-organization1 --id.name
peer0 --id.secret peer0pw --id.type peer --tls.certfiles
${PWD}/fabric-ca/org1/tls-cert.pem

  echo
  echo "Register user"
  echo
```

```
  fabric-ca-client register --caname ca-organization1 --id.name
user1 --id.secret user1pw --id.type client --tls.certfiles
${PWD}/fabric-ca/org1/tls-cert.pem

  echo
  echo "Register the org admin"
  echo
  fabric-ca-client register --caname ca-organization1 --id.name
org1admin --id.secret org1adminpw --id.type admin --tls.certfiles
${PWD}/fabric-ca/org1/tls-cert.pem

  mkdir -p ../crypto-config/peerOrganizations/organization1/peers

  # ---------------------------------------------------------------
------------------
  #  Peer 0
  mkdir -p ../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1

  echo
  echo "## Generate the peer0 msp"
  echo
  fabric-ca-client enroll -u https://peer0:peer0pw@localhost:1001 --
caname ca-organization1 -M ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/msp
--csr.hosts peer0.organization1 --tls.certfiles ${PWD}/fabric-
ca/org1/tls-cert.pem

  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/msp/config.yaml
${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/msp
/config.yaml

  echo
  echo "## Generate the peer0-tls certificates"
  echo
  fabric-ca-client enroll -u https://peer0:peer0pw@localhost:1001 --
caname ca-organization1 -M ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
--enrollment.profile tls --csr.hosts peer0.organization1 --csr.hosts
localhost --tls.certfiles ${PWD}/fabric-ca/org1/tls-cert.pem

  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
```

```
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/ca.crt
  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/signcerts/* ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/server.crt
  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/keystore/* ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/server.key

  mkdir ${PWD}/../crypto-
config/peerOrganizations/organization1/msp/tlscacerts
  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization1/msp/tlscacerts/ca.crt

  mkdir ${PWD}/../crypto-
config/peerOrganizations/organization1/tlsca
  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization1/tlsca/tlsca-organization1-
cert.pem

  mkdir ${PWD}/../crypto-config/peerOrganizations/organization1/ca
  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/msp
/cacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization1/ca/ca-organization1-cert.pem

  # ------------------------------------------------------------------
----------------------------------

  mkdir -p ../crypto-config/peerOrganizations/organization1/users
  mkdir -p ../crypto-
config/peerOrganizations/organization1/users/User1@organization1

  echo
  echo "## Generate the user msp"
  echo
```

```
  fabric-ca-client enroll -u https://user1:user1pw@localhost:1001 --
caname ca-organization1 -M ${PWD}/../crypto-
config/peerOrganizations/organization1/users/User1@organization1/msp
--tls.certfiles ${PWD}/fabric-ca/org1/tls-cert.pem

  mkdir -p ../crypto-
config/peerOrganizations/organization1/users/Admin@organization1

  echo
  echo "## Generate the org admin msp"
  echo
  fabric-ca-client enroll -u
https://org1admin:org1adminpw@localhost:1001 --caname ca-
organization1 -M ${PWD}/../crypto-
config/peerOrganizations/organization1/users/Admin@organization1/msp
--tls.certfiles ${PWD}/fabric-ca/org1/tls-cert.pem

  cp ${PWD}/../crypto-
config/peerOrganizations/organization1/msp/config.yaml
${PWD}/../crypto-
config/peerOrganizations/organization1/users/Admin@organization1/msp
/config.yaml

}

# createcertificatesForOrg1

createCertificatesForOrg2() {
  echo
  echo "Enroll the CA admin"
  echo
  mkdir -p /../crypto-config/peerOrganizations/organization2/

  export FABRIC_CA_CLIENT_HOME=${PWD}/../crypto-
config/peerOrganizations/organization2/


  fabric-ca-client enroll -u https://admin:adminpw@localhost:1002 --
caname ca-organization2 --tls.certfiles ${PWD}/fabric-ca/org2/tls-
cert.pem


  echo 'NodeOUs:
  Enable: true
  ClientOUIdentifier:
```

```
    Certificate: cacerts/localhost-1002-ca-organization2.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/localhost-1002-ca-organization2.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/localhost-1002-ca-organization2.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/localhost-1002-ca-organization2.pem
    OrganizationalUnitIdentifier: orderer' >${PWD}/../crypto-
config/peerOrganizations/organization2/msp/config.yaml

  echo
  echo "Register peer0"
  echo

  fabric-ca-client register --caname ca-organization2 --id.name
peer0 --id.secret peer0pw --id.type peer --tls.certfiles
${PWD}/fabric-ca/org2/tls-cert.pem


  echo
  echo "Register user"
  echo

  fabric-ca-client register --caname ca-organization2 --id.name
user1 --id.secret user1pw --id.type client --tls.certfiles
${PWD}/fabric-ca/org2/tls-cert.pem


  echo
  echo "Register the org admin"
  echo

  fabric-ca-client register --caname ca-organization2 --id.name
org2admin --id.secret org2adminpw --id.type admin --tls.certfiles
${PWD}/fabric-ca/org2/tls-cert.pem


  mkdir -p ../crypto-config/peerOrganizations/organization2/peers
  mkdir -p ../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2

  # ---------------------------------------------------------------
```

```
  # Peer 0
  echo
  echo "## Generate the peer0 msp"
  echo

  fabric-ca-client enroll -u https://peer0:peer0pw@localhost:1002 --
caname ca-organization2 -M ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/msp
--csr.hosts peer0.organization2 --tls.certfiles ${PWD}/fabric-
ca/org2/tls-cert.pem


  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/msp/config.yaml
${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/msp
/config.yaml

  echo
  echo "## Generate the peer0-tls certificates"
  echo

  fabric-ca-client enroll -u https://peer0:peer0pw@localhost:1002 --
caname ca-organization2 -M ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
--enrollment.profile tls --csr.hosts peer0.organization2 --csr.hosts
localhost --tls.certfiles ${PWD}/fabric-ca/org2/tls-cert.pem


  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/ca.crt
  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/signcerts/* ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/server.crt
  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/keystore/* ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/server.key
```

```
  mkdir ${PWD}/../crypto-
config/peerOrganizations/organization2/msp/tlscacerts
  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization2/msp/tlscacerts/ca.crt

  mkdir ${PWD}/../crypto-
config/peerOrganizations/organization2/tlsca
  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization2/tlsca/tlsca-organization2-
cert.pem

  mkdir ${PWD}/../crypto-config/peerOrganizations/organization2/ca
  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/msp
/cacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization2/ca/ca-organization2-cert.pem

  # ------------------------------------------------------------
----------------

  mkdir -p ../crypto-config/peerOrganizations/organization2/users
  mkdir -p ../crypto-
config/peerOrganizations/organization2/users/User1@organization2

  echo
  echo "## Generate the user msp"
  echo

  fabric-ca-client enroll -u https://user1:user1pw@localhost:1002 --
caname ca-organization2 -M ${PWD}/../crypto-
config/peerOrganizations/organization2/users/User1@organization2/msp
--tls.certfiles ${PWD}/fabric-ca/org2/tls-cert.pem


  mkdir -p ../crypto-
config/peerOrganizations/organization2/users/Admin@organization2

  echo
  echo "## Generate the org admin msp"
  echo
```

```
  fabric-ca-client enroll -u
https://org2admin:org2adminpw@localhost:1002 --caname ca-
organization2 -M ${PWD}/../crypto-
config/peerOrganizations/organization2/users/Admin@organization2/msp
--tls.certfiles ${PWD}/fabric-ca/org2/tls-cert.pem


  cp ${PWD}/../crypto-
config/peerOrganizations/organization2/msp/config.yaml
${PWD}/../crypto-
config/peerOrganizations/organization2/users/Admin@organization2/msp
/config.yaml

}

# createCertificateForOrg2

createCertificatesForOrg3() {
  echo
  echo "Enroll the CA admin"
  echo
  mkdir -p ../crypto-config/peerOrganizations/organization3/

  export FABRIC_CA_CLIENT_HOME=${PWD}/../crypto-
config/peerOrganizations/organization3/


  fabric-ca-client enroll -u https://admin:adminpw@localhost:1003 --
caname ca-organization3 --tls.certfiles ${PWD}/fabric-ca/org3/tls-
cert.pem


  echo 'NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/localhost-1003-ca-organization3.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/localhost-1003-ca-organization3.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/localhost-1003-ca-organization3.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/localhost-1003-ca-organization3.pem
```

```
    OrganizationalUnitIdentifier: orderer' >${PWD}/../crypto-
config/peerOrganizations/organization3/msp/config.yaml

  echo
  echo "Register peer0"
  echo

  fabric-ca-client register --caname ca-organization3 --id.name
peer0 --id.secret peer0pw --id.type peer --tls.certfiles
${PWD}/fabric-ca/org3/tls-cert.pem


  echo
  echo "Register user"
  echo

  fabric-ca-client register --caname ca-organization3 --id.name
user1 --id.secret user1pw --id.type client --tls.certfiles
${PWD}/fabric-ca/org3/tls-cert.pem


  echo
  echo "Register the org admin"
  echo

  fabric-ca-client register --caname ca-organization3 --id.name
org3admin --id.secret org3adminpw --id.type admin --tls.certfiles
${PWD}/fabric-ca/org3/tls-cert.pem


  mkdir -p ../crypto-config/peerOrganizations/organization3/peers
  mkdir -p ../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3

  # ---------------------------------------------------------------
  # Peer 0
  echo
  echo "## Generate the peer0 msp"
  echo

  fabric-ca-client enroll -u https://peer0:peer0pw@localhost:1003 --
caname ca-organization3 -M ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/msp
--csr.hosts peer0.organization3 --tls.certfiles ${PWD}/fabric-
ca/org3/tls-cert.pem
```

```
  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/msp/config.yaml
${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/msp
/config.yaml

  echo
  echo "## Generate the peer0-tls certificates"
  echo

  fabric-ca-client enroll -u https://peer0:peer0pw@localhost:1003 --
caname ca-organization3 -M ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
--enrollment.profile tls --csr.hosts peer0.organization3 --csr.hosts
localhost --tls.certfiles ${PWD}/fabric-ca/org3/tls-cert.pem


  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/ca.crt
  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/signcerts/* ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/server.crt
  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/keystore/* ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/server.key

  mkdir ${PWD}/../crypto-
config/peerOrganizations/organization3/msp/tlscacerts
  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization3/msp/tlscacerts/ca.crt

  mkdir ${PWD}/../crypto-
config/peerOrganizations/organization3/tlsca
```

```
  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/tlscacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization3/tlsca/tlsca-organization3-
cert.pem

  mkdir ${PWD}/../crypto-config/peerOrganizations/organization3/ca
  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/msp
/cacerts/* ${PWD}/../crypto-
config/peerOrganizations/organization3/ca/ca-organization3-cert.pem

  # ------------------------------------------------------------------
----------------

  mkdir -p ../crypto-config/peerOrganizations/organization3/users
  mkdir -p ../crypto-
config/peerOrganizations/organization3/users/User1@organization3

  echo
  echo "## Generate the user msp"
  echo

  fabric-ca-client enroll -u https://user1:user1pw@localhost:1003 --
caname ca-organization3 -M ${PWD}/../crypto-
config/peerOrganizations/organization3/users/User1@organization3/msp
--tls.certfiles ${PWD}/fabric-ca/org3/tls-cert.pem


  mkdir -p ../crypto-
config/peerOrganizations/organization3/users/Admin@organization3

  echo
  echo "## Generate the org admin msp"
  echo

  fabric-ca-client enroll -u
https://org3admin:org3adminpw@localhost:1003 --caname ca-
organization3 -M ${PWD}/../crypto-
config/peerOrganizations/organization3/users/Admin@organization3/msp
--tls.certfiles ${PWD}/fabric-ca/org3/tls-cert.pem


  cp ${PWD}/../crypto-
config/peerOrganizations/organization3/msp/config.yaml
```

```
${PWD}/../crypto-
config/peerOrganizations/organization3/users/Admin@organization3/msp
/config.yaml

}

createCretificatesForOrderer() {
  echo
  echo "Enroll the CA admin"
  echo
  mkdir -p ../crypto-config/ordererOrganizations/example.com

  export FABRIC_CA_CLIENT_HOME=${PWD}/../crypto-
config/ordererOrganizations/example.com


  fabric-ca-client enroll -u https://admin:adminpw@localhost:9054 --
caname ca-orderer --tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-
cert.pem


  echo 'NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/localhost-9054-ca-orderer.pem
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/localhost-9054-ca-orderer.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/localhost-9054-ca-orderer.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/localhost-9054-ca-orderer.pem
    OrganizationalUnitIdentifier: orderer' >${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/config.yaml

  echo
  echo "Register orderer"
  echo

  fabric-ca-client register --caname ca-orderer --id.name orderer --
id.secret ordererpw --id.type orderer --tls.certfiles ${PWD}/fabric-
ca/ordererOrg/tls-cert.pem
```

```
  echo
  echo "Register orderer2"
  echo

  fabric-ca-client register --caname ca-orderer --id.name orderer2 -
-id.secret ordererpw --id.type orderer --tls.certfiles
${PWD}/fabric-ca/ordererOrg/tls-cert.pem


  echo
  echo "Register orderer3"
  echo

  fabric-ca-client register --caname ca-orderer --id.name orderer3 -
-id.secret ordererpw --id.type orderer --tls.certfiles
${PWD}/fabric-ca/ordererOrg/tls-cert.pem


  echo
  echo "Register the orderer admin"
  echo

  fabric-ca-client register --caname ca-orderer --id.name
ordererAdmin --id.secret ordererAdminpw --id.type admin --
tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-cert.pem


  mkdir -p ../crypto-
config/ordererOrganizations/example.com/orderers
  # mkdir -p ../crypto-
config/ordererOrganizations/example.com/orderers/example.com

  # ---------------------------------------------------------------
-----------
  #  Orderer

  mkdir -p ../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com

  echo
  echo "## Generate the orderer msp"
  echo
```

```
  fabric-ca-client enroll -u
https://orderer:ordererpw@localhost:9054 --caname ca-orderer -M
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp --csr.hosts orderer.example.com --csr.hosts localhost --
tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-cert.pem


  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/config.yaml
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp/config.yaml

  echo
  echo "## Generate the orderer-tls certificates"
  echo

  fabric-ca-client enroll -u
https://orderer:ordererpw@localhost:9054 --caname ca-orderer -M
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls --enrollment.profile tls --csr.hosts orderer.example.com --
csr.hosts localhost --tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-
cert.pem


  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/ca.crt
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/signcerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/server.crt
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/keystore/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/server.key
```

```
  mkdir ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp/tlscacerts
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp/tlscacerts/tlsca.example.com-cert.pem

  mkdir ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/tlscacerts
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example
.com-cert.pem

  # ---------------------------------------------------------------
-------
  #  Orderer 2

  mkdir -p ../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m

  echo
  echo "## Generate the orderer msp"
  echo

  fabric-ca-client enroll -u
https://orderer2:ordererpw@localhost:9054 --caname ca-orderer -M
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/msp --csr.hosts orderer2.example.com --csr.hosts localhost --
tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-cert.pem


  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/config.yaml
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/msp/config.yaml

  echo
  echo "## Generate the orderer-tls certificates"
```

```
  echo

  fabric-ca-client enroll -u
https://orderer2:ordererpw@localhost:9054 --caname ca-orderer -M
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls --enrollment.profile tls --csr.hosts orderer2.example.com --
csr.hosts localhost --tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-
cert.pem


  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/ca.crt
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/signcerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/server.crt
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/keystore/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/server.key

  mkdir ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/msp/tlscacerts
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/msp/tlscacerts/tlsca.example.com-cert.pem

  # mkdir ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/tlscacerts
  # cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer2.example.co
m/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example
.com-cert.pem
```

```
  # ---------------------------------------------------------------
-----------
  #  Orderer 3
  mkdir -p ../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m

  echo
  echo "## Generate the orderer msp"
  echo

  fabric-ca-client enroll -u
https://orderer3:ordererpw@localhost:9054 --caname ca-orderer -M
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/msp --csr.hosts orderer3.example.com --csr.hosts localhost --
tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-cert.pem


  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/config.yaml
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/msp/config.yaml

  echo
  echo "## Generate the orderer-tls certificates"
  echo

  fabric-ca-client enroll -u
https://orderer3:ordererpw@localhost:9054 --caname ca-orderer -M
${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls --enrollment.profile tls --csr.hosts orderer3.example.com --
csr.hosts localhost --tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-
cert.pem


  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/ca.crt
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
```

```
m/tls/signcerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/server.crt
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/keystore/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/server.key

  mkdir ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/msp/tlscacerts
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/msp/tlscacerts/tlsca.example.com-cert.pem

  # mkdir ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/tlscacerts
  # cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/orderers/orderer3.example.co
m/tls/tlscacerts/* ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/tlscacerts/tlsca.example
.com-cert.pem

  # ----------------------------------------------------------------
-----------

  mkdir -p ../crypto-config/ordererOrganizations/example.com/users
  mkdir -p ../crypto-
config/ordererOrganizations/example.com/users/Admin@example.com

  echo
  echo "## Generate the admin msp"
  echo

  fabric-ca-client enroll -u
https://ordererAdmin:ordererAdminpw@localhost:9054 --caname ca-
orderer -M ${PWD}/../crypto-
config/ordererOrganizations/example.com/users/Admin@example.com/msp
--tls.certfiles ${PWD}/fabric-ca/ordererOrg/tls-cert.pem
```

```
  cp ${PWD}/../crypto-
config/ordererOrganizations/example.com/msp/config.yaml
${PWD}/../crypto-
config/ordererOrganizations/example.com/users/Admin@example.com/msp/
config.yaml

}



# sudo rm -rf ../crypto-config/*
# sudo rm -rf fabric-ca/*
createcertificatesForOrg1
createCertificatesForOrg2
createCertificatesForOrg3

createCretificatesForOrderer


# DOCKER-COMPOSE CERTIFICATE
version: '2'

networks:
  test:

services:

  ca_org1:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-organization1
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=1001
    ports:
      - "1001:1001"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ./fabric-ca/org1:/etc/hyperledger/fabric-ca-server
    container_name: ca-organization1
    hostname: ca-organization1
    networks:
      - test

  ca_org2:
```

```
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-organization2
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=1002
    ports:
      - "1002:1002"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ./fabric-ca/org2:/etc/hyperledger/fabric-ca-server
    container_name: ca-organization2
    hostname: ca-organization2
    networks:
      - test

  ca_org3:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-organization3
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=1003
    ports:
      - "1003:1003"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ./fabric-ca/org3:/etc/hyperledger/fabric-ca-server
    container_name: ca-organization3
    hostname: ca-organization3
    networks:
      - test

  ca_orderer:
    image: hyperledger/fabric-ca
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-orderer
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=9054
    ports:
      - "9054:9054"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ./fabric-ca/ordererOrg:/etc/hyperledger/fabric-ca-server
```

```
    container_name: ca_orderer
    networks:
      - test


# DEPLOY CHAINCODE
export CORE_PEER_TLS_ENABLED=true
export ORDERER_CA=${PWD}/artifacts/channel/crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp/tlscacerts/tlsca.example.com-cert.pem
export PEER0_ORG1_CA=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/ca.crt
export PEER0_ORG2_CA=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/ca.crt
export PEER0_ORG3_CA=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/ca.crt
export FABRIC_CFG_PATH=${PWD}/artifacts/channel/config/

export CHANNEL_NAME=smartgrid

setGlobalsForOrderer() {
    export CORE_PEER_LOCALMSPID="OrdererMSP"
    export
CORE_PEER_TLS_ROOTCERT_FILE=${PWD}/artifacts/channel/crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp/tlscacerts/tlsca.example.com-cert.pem
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/ordererOrganizations/example.com/users/Admin@example.com/msp

}

setGlobalsForPeer0Org1() {
    export CORE_PEER_LOCALMSPID="Org1MSP"
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG1_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization1/users/Admin@organization1/msp
    export CORE_PEER_ADDRESS=localhost:7051
}

setGlobalsForOrg1() {
    export CORE_PEER_LOCALMSPID="Org1MSP"
```

```
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG1_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization1/users/User1@organization1/msp
    export CORE_PEER_ADDRESS=localhost:7051
}

setGlobalsForPeer0Org2() {
    export CORE_PEER_LOCALMSPID="Org2MSP"
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG2_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization2/users/Admin@organization2/msp
    export CORE_PEER_ADDRESS=localhost:9051

}

setGlobalsForPeer0Org3(){
    export CORE_PEER_LOCALMSPID="Org3MSP"
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG3_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization3/users/Admin@organization3/msp
    export CORE_PEER_ADDRESS=localhost:11051

}

presetup() {
    echo Vendoring Go dependencies ...
    pushd ./artifacts/smartcontract/go
    GO111MODULE=on go mod vendor
    popd
    echo Finished vendoring Go dependencies
}
# presetup

CHANNEL_NAME="smartgrid"
CC_RUNTIME_LANGUAGE="golang"
VERSION="1"
SEQUENCE="1"
CC_SRC_PATH="./artifacts/smartcontract/go"
CC_NAME="smartcontract"

packageChaincode() {
    rm -rf ${CC_NAME}.tar.gz
    setGlobalsForPeer0Org1
    peer lifecycle chaincode package ${CC_NAME}.tar.gz \
        --path ${CC_SRC_PATH} --lang ${CC_RUNTIME_LANGUAGE} \
```

```
        --label ${CC_NAME}_${VERSION}
    echo "===================== Chaincode is packaged
===================== "
}
# packageChaincode

installChaincode() {
    setGlobalsForPeer0Org1
    peer lifecycle chaincode install ${CC_NAME}.tar.gz
    echo "===================== Chaincode is installed on peer0.org1
===================== "

    setGlobalsForPeer0Org2
    peer lifecycle chaincode install ${CC_NAME}.tar.gz
    echo "===================== Chaincode is installed on peer0.org2
===================== "

    setGlobalsForPeer0Org3
    peer lifecycle chaincode install ${CC_NAME}.tar.gz
    echo "===================== Chaincode is installed on peer0.org3
===================== "
}

# installChaincode

queryInstalled() {
    setGlobalsForPeer0Org1
    peer lifecycle chaincode queryinstalled >&log.txt
    cat log.txt
    PACKAGE_ID=$(sed -n "/${CC_NAME}_${VERSION}/{s/^Package ID: //;
s/, Label:.*$//; p;}" log.txt)
    echo PackageID is ${PACKAGE_ID}
    echo "===================== Query installed successful on
peer0.org1 on channel ===================== "
}

# queryInstalled

# --collections-config ./artifacts/private-
data/collections_config.json \
#         --signature-policy "OR('Org1MSP.member','Org2MSP.member')"
\

approveForMyOrg1() {
    setGlobalsForPeer0Org1
```

```
    # set -x
    peer lifecycle chaincode approveformyorg -o localhost:7050 \
        --ordererTLSHostnameOverride orderer.example.com --tls \
        --cafile $ORDERER_CA --channelID $CHANNEL_NAME --name
${CC_NAME} --version ${VERSION} \
        --init-required --package-id ${PACKAGE_ID} \
        --sequence ${SEQUENCE}
    # set +x

    echo "===================== chaincode approved from org 1
===================== "

}
# queryInstalled
# approveForMyOrg1

# --signature-policy "OR ('Org1MSP.member')"
# --peerAddresses localhost:7051 --tlsRootCertFiles $PEER0_ORG1_CA -
-peerAddresses localhost:9051 --tlsRootCertFiles $PEER0_ORG2_CA
# --peerAddresses peer0.organization1:7051 --tlsRootCertFiles
$PEER0_ORG1_CA --peerAddresses peer0.organization2:9051 --
tlsRootCertFiles $PEER0_ORG2_CA
#--channel-config-policy Channel/Application/Admins
# --signature-policy "OR ('Org1MSP.peer','Org2MSP.peer')"

checkCommitReadyness() {
    setGlobalsForPeer0Org1
    peer lifecycle chaincode checkcommitreadiness \
        --channelID $CHANNEL_NAME --name ${CC_NAME} --version
${VERSION} \
        --sequence ${VERSION} --output json --init-required
    echo "===================== checking commit readyness from org 1
===================== "
}

# checkCommitReadyness

approveForMyOrg2() {
    setGlobalsForPeer0Org2

    peer lifecycle chaincode approveformyorg -o localhost:7050 \
        --ordererTLSHostnameOverride orderer.example.com --tls
$CORE_PEER_TLS_ENABLED \
        --cafile $ORDERER_CA --channelID $CHANNEL_NAME --name
${CC_NAME} \
```

```
        --version ${VERSION} --init-required --package-id
${PACKAGE_ID} \
        --sequence ${SEQUENCE}

    echo "===================== chaincode approved from org 2
===================== "
}

# queryInstalled
# approveForMyOrg2

checkCommitReadyness() {

    setGlobalsForPeer0Org2
    peer lifecycle chaincode checkcommitreadiness --channelID
$CHANNEL_NAME \
        --peerAddresses localhost:9051 --tlsRootCertFiles
$PEER0_ORG2_CA \
        --name ${CC_NAME} --version ${VERSION} --sequence ${VERSION}
--output json --init-required
    echo "===================== checking commit readyness from org 1
===================== "
}

# checkCommitReadyness

approveForMyOrg3() {
    setGlobalsForPeer0Org3

    peer lifecycle chaincode approveformyorg -o localhost:7050 \
        --ordererTLSHostnameOverride orderer.example.com --tls
$CORE_PEER_TLS_ENABLED \
        --cafile $ORDERER_CA --channelID $CHANNEL_NAME --name
${CC_NAME} \
        --version ${VERSION} --init-required --package-id
${PACKAGE_ID} \
        --sequence ${SEQUENCE}

    echo "===================== chaincode approved from org 2
===================== "
}

# queryInstalled
# approveForMyOrg3
```

```
checkCommitReadyness() {

    setGlobalsForPeer0Org3
    peer lifecycle chaincode checkcommitreadiness --channelID
$CHANNEL_NAME \
        --peerAddresses localhost:11051 --tlsRootCertFiles
$PEER0_ORG3_CA \
        --name ${CC_NAME} --version ${VERSION} --sequence ${VERSION}
--output json --init-required
    echo "==================== checking commit readyness from org 1
==================== "
}

# checkCommitReadyness

commitChaincodeDefination() {
    setGlobalsForPeer0Org1
    peer lifecycle chaincode commit -o localhost:7050 --
ordererTLSHostnameOverride orderer.example.com \
        --tls $CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA \
        --channelID $CHANNEL_NAME --name ${CC_NAME} \
        --peerAddresses localhost:7051 --tlsRootCertFiles
$PEER0_ORG1_CA \
        --peerAddresses localhost:9051 --tlsRootCertFiles
$PEER0_ORG2_CA \
        --peerAddresses localhost:11051 --tlsRootCertFiles
$PEER0_ORG3_CA \
        --version ${VERSION} --sequence ${SEQUENCE} --init-required

}

# commitChaincodeDefination

queryCommitted() {
    setGlobalsForPeer0Org1
    peer lifecycle chaincode querycommitted --channelID
$CHANNEL_NAME --name ${CC_NAME}

}

# queryCommitted

chaincodeInvokeInit() {
    setGlobalsForPeer0Org1
    peer chaincode invoke -o localhost:7050 \
```

```
        --ordererTLSHostnameOverride orderer.example.com \
        --tls $CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA \
        -C $CHANNEL_NAME -n ${CC_NAME} \
        --peerAddresses localhost:7051 --tlsRootCertFiles
$PEER0_ORG1_CA \
        --peerAddresses localhost:9051 --tlsRootCertFiles
$PEER0_ORG2_CA \
        --peerAddresses localhost:11051 --tlsRootCertFiles
$PEER0_ORG3_CA \
        --isInit -c '{"Args":[]}'

}

# chaincodeInvokeInit

chaincodeInvoke() {
    setGlobalsForPeer0Org1

    # Create Car
    peer chaincode invoke -o localhost:7050 \
        --ordererTLSHostnameOverride orderer.example.com \
        --tls $CORE_PEER_TLS_ENABLED \
        --cafile $ORDERER_CA \
        -C $CHANNEL_NAME -n ${CC_NAME}  \
        --peerAddresses localhost:7051 --tlsRootCertFiles
$PEER0_ORG1_CA \
        --peerAddresses localhost:9051 --tlsRootCertFiles
$PEER0_ORG2_CA \
         --peerAddresses localhost:11051 --tlsRootCertFiles
$PEER0_ORG3_CA \
         -c '{"function": "CreateData","Args":["PV-0", "1kwh",
"testing", "User0","2023"]}'

}

# chaincodeInvoke

chaincodeInvokeDeleteAsset() {
    setGlobalsForPeer0Org1

    # Create Car
    peer chaincode invoke -o localhost:7050 \
        --ordererTLSHostnameOverride orderer.example.com \
        --tls $CORE_PEER_TLS_ENABLED \
        --cafile $ORDERER_CA \
```

```
        -C $CHANNEL_NAME -n ${CC_NAME}  \
        --peerAddresses localhost:7051 --tlsRootCertFiles
$PEER0_ORG1_CA \
        --peerAddresses localhost:9051 --tlsRootCertFiles
$PEER0_ORG2_CA \
        --peerAddresses localhost:11051 --tlsRootCertFiles
$PEER0_ORG3_CA \
        -c '{"function": "DeleteDataById","Args":["PV-0"]}'

}

# chaincodeInvokeDeleteAsset

chaincodeQuery() {
    setGlobalsForPeer0Org1
    # setGlobalsForOrg1

      # Query all cars
    peer chaincode query -C $CHANNEL_NAME -n ${CC_NAME} -c
'{"Args":["QueryAllDatas"]}'

    # Query Car by Id
    peer chaincode query -C $CHANNEL_NAME -n ${CC_NAME} -c
'{"function": "QueryData","Args":["PV-0"]}'
}

# chaincodeQuery

# Run this function if you add any new dependency in chaincode
presetup

packageChaincode
installChaincode
queryInstalled
approveForMyOrg1
checkCommitReadyness
approveForMyOrg2
checkCommitReadyness
approveForMyOrg3
checkCommitReadyness
commitChaincodeDefination
queryCommitted
chaincodeInvokeInit
sleep 5
chaincodeInvoke
```

```
sleep 3
chaincodeQuery

# CREATE CHANNEL
export CORE_PEER_TLS_ENABLED=true
export ORDERER_CA=${PWD}/artifacts/channel/crypto-
config/ordererOrganizations/example.com/orderers/orderer.example.com
/msp/tlscacerts/tlsca.example.com-cert.pem
export PEER0_ORG1_CA=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization1/peers/peer0.organization1/tls
/ca.crt
export PEER0_ORG2_CA=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization2/peers/peer0.organization2/tls
/ca.crt
export PEER0_ORG3_CA=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization3/peers/peer0.organization3/tls
/ca.crt
export FABRIC_CFG_PATH=${PWD}/artifacts/channel/config/

export CHANNEL_NAME=smartgrid

setGlobalsForPeer0Org1(){
    export CORE_PEER_LOCALMSPID="Org1MSP"
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG1_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization1/users/Admin@organization1/msp
    export CORE_PEER_ADDRESS=localhost:7051
}

setGlobalsForPeer0Org2(){
    export CORE_PEER_LOCALMSPID="Org2MSP"
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG2_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization2/users/Admin@organization2/msp
    export CORE_PEER_ADDRESS=localhost:9051

}

setGlobalsForPeer0Org3(){
    export CORE_PEER_LOCALMSPID="Org3MSP"
    export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0_ORG3_CA
    export CORE_PEER_MSPCONFIGPATH=${PWD}/artifacts/channel/crypto-
config/peerOrganizations/organization3/users/Admin@organization3/msp
    export CORE_PEER_ADDRESS=localhost:11051
```

```
}

createChannel(){
    rm -rf ./channel-artifacts/*
    setGlobalsForPeer0Org1

    peer channel create -o localhost:7050 -c $CHANNEL_NAME \
    --ordererTLSHostnameOverride orderer.example.com \
    -f ./artifacts/channel/${CHANNEL_NAME}.tx --outputBlock
./channel-artifacts/${CHANNEL_NAME}.block \
    --tls $CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA
}

removeOldCrypto(){
    rm -rf ./api-1.4/crypto/*
    rm -rf ./api-1.4/fabric-client-kv-org1/*
    rm -rf ./api-2.0/org1-wallet/*
    rm -rf ./api-2.0/org2-wallet/*
}


joinChannel(){
    setGlobalsForPeer0Org1
    peer channel join -b ./channel-artifacts/$CHANNEL_NAME.block


    setGlobalsForPeer0Org2
    peer channel join -b ./channel-artifacts/$CHANNEL_NAME.block

    setGlobalsForPeer0Org3
    peer channel join -b ./channel-artifacts/$CHANNEL_NAME.block

}

updateAnchorPeers(){
    setGlobalsForPeer0Org1
    peer channel update -o localhost:7050 --
ordererTLSHostnameOverride orderer.example.com -c $CHANNEL_NAME -f
./artifacts/channel/${CORE_PEER_LOCALMSPID}anchors.tx --tls
$CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA

    setGlobalsForPeer0Org2
    peer channel update -o localhost:7050 --
ordererTLSHostnameOverride orderer.example.com -c $CHANNEL_NAME -f
```

```
./artifacts/channel/${CORE_PEER_LOCALMSPID}anchors.tx --tls
$CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA

    setGlobalsForPeer0Org3
    peer channel update -o localhost:7050 --
ordererTLSHostnameOverride orderer.example.com -c $CHANNEL_NAME -f
./artifacts/channel/${CORE_PEER_LOCALMSPID}anchors.tx --tls
$CORE_PEER_TLS_ENABLED --cafile $ORDERER_CA

}

removeOldCrypto

createChannel
joinChannel
updateAnchorPeers
```