What is the Hyperledger Fabric System Architecture?

Before beginning with our core **Hyperledger Fabric tutorial,** it is essential to learn about the underlying architecture. Having a firm knowledge of the structure will help you understand how the framework operates.

- **Transactions:** There are two types of transactions: deploy and invoke. The deploy transactions are used for creating chaincodes, which are programs run by the blockchain. On the other hand, invoke transactions are used to execute transactions in context with the previously deployed chaincodes.
- **Data structures:** Like any blockchain framework, the data structures used in Hyperledger Fabric are in key-value pairs. The chain codes manipulate the data with the put and get operations.
- **Ledger:** It is a sequence of all the successful and unsuccessful changes to the chain.
- **Nodes:** They are the communication agents or, in other words, entities of a network.
- **Client:** End-users or someone who acts on their behalf.
- **Peer:** They handle the states of the chain and ensure everything runs smoothly.

Now since we know the basic architecture, it's time to develop our first Hyperledger Fabric app.

Building and Launching the First Network

This step comprises installing prerequisites and sample components to initiate developing our app.

Building the Network

Once everything is installed, use the following command to move to the repository with the sample network.

```
cd fabric-samples/first-network
```

Next, use the byfn.sh script to run the network. Running this network will facilitate network communication and launch the essentials like containers, peers, chaincode, etc., that will help with the other process. After running the sample network, it's time to generate the artifacts.

Generating the Artifacts

Use the following code to generate the artifacts and then press "Y" when asked for confirmation to start filling in the necessary details.

```
.byfn.sh generate
```

Powering the Network

After generating the artifacts, you need to power the network. It will fire up all the components, including chaincode and containers, to allow new peers to join the network. Use the following code for powering the network:

```
./byfn.sh up
```

The default language for Hyperledger Fabric chaincode is Go. However, you can switch to using Node.js or Java by using the following commands respectively:

```
.byfn.sh up -l node
```

```
.byfn.sh up -l java
```

You can also opt for using multiple programming languages by using either of the following codes:

```
./byfn.sh up -o etcdraft
```

```
./byfn.sh up -o kafka
```

Launching the Network

Now we will be launching the network. But before that we have to:

- get our application's subdirectory
- make a local clone of the fabric-sample repo

Use the following command script to launch the network:

```
./startFabric.sh javascript
```

After launching the network, you will have everything right from orderers to peers to certificate authorities for developing your application. Since we are using the application's subdirectory, a smart contract will be auto-launched.

Installing Your First Application

Use the following code to install the dependencies required for installing and initiating the application.

```
npm install
```

Connecting With Certificate Authorities

Once the network is up and running, let’s start by communicating with the authorities who can allow us to add users to the chain. To do this, we need to undergo the Certificate Signing Request (CSR). Use the below command to contact the certificate authority and create a public, private, and x.509 certificate.

```
node enrollAdmin.js
```

Your information will now be held in the wallet. Similarly, to add any new user, you need to go through the same process and use the following code:

```
node reisterUser.js
```

This code will add the user, and the details will be held in the wallet.

Querying a Ledger

You can use read queries to query a ledger through the blockchain network and get the results. The application won’t execute any write queries as ledgers are read-only.

Use the below code to retrieve all the data from the ledger:

```
node query.js
```

Next, we need to establish a connection to convey the data. After the connection is made, create a gateway to allow the application to connect to the network. The following line of codes will help you do all this.

```
const { FileSystemWallet, Gateway } = require(‘fabric-network’);

const gateway = new Gateway();

await gateway.connect(ccp, { wallet, identity: ‘user1’ });

const contract = network.getContract(‘subdirectory_name’);
```

The CPP in the above code is used to access and verify user information. The last line of the code is connecting to a particular channel. It is essential as the entire blockchain network runs on multiple channels.

Understanding the Contract

Head to the chaincode/subdirecory_name/javascript/lib subdirectory to access the application’s smart contract. In the above directory, the subdirectory_name will be the name of your application’s subdirectory. Open the application_name.js file you get here in an editor. In this file, you will get the smart contract and all the transactions defined within the Contract class. Suppose your application is about cars, the transaction details will look like:

```
async queryCar(ctx, carNumber) {…}
```

Read: Why Blockchain is the Future

Updating the Ledger

The last part of this **Hyperledger Fabric tutorial for beginners** is to update the ledger. Considering the same car example, let's look at how you can add a new car in the log. Use the following line of code to add a new vehicle to the ledger.

```
await contract.submitTransaction('createCar', 'CAR15', 'Owner', 'Brand',
'CarModel', 'Color');
```

Using the above command will create a new car along with the mentioned attribute values. To send this new transaction to the ledger, write the below code:

```
node invoke.js
```

This will successfully update the ledger.

Conclusion

That brings us to the end of our **Hyperledger Fabric tutorial.** Blockchain technology and Hyperledger Fabric framework have the potential to revolutionize multiple industries across the world. This tutorial gave a brief idea of how the Hyperledger Fabric framework works and how to run your first transaction and app on it.