DECENTRALIZED ENERGY WITH HYPERLEDGER COMPOSER

MARCH 7, 2018

IBM CODE TECH TALK


>> MARC-ARTHUR PIERRE LOUIS: Good afternoon. This is Marc-Arthur Pierre Louis, your moderator for this series of tech talks, on about technology and about IBM code patterns and about open source projects. Today we have another tech talk on the Blockchain technologies. It's going to be an interesting one, where you run a application that lets you see how you can dole out energy between people who have solar panels and other means of energy, and are able to either sell it to other consumers or sell it back to the utility company. That will show you how that is done.

We have on speaking our friend Raheel Zubairy, software engineer that is focused on developing patterns for the emerging technologies team. He is going to be showcasing Blockchain, that is one of his strong points, strong suits.

He is interested in [inaudible] been working with the energy industry for the last five years. Raheel will take you through this interesting presentation. I can't wait to hear him going through it.

>> RAHEEL ZUBAIRY: Thank you, and welcome everyone to this tech talk on decentralized energy with Hyperledger Composer and IBM code pattern. The agenda for the talk is that we will talk first about the application, we will talk about the purpose of Blockchain for this application, and we will discuss the architecture of the application on how everything is connected. Then I'll give a demo of the application and how you can perform the transactions through the application and use the transactions on the Blockchain ledger. After that we will go through steps on how you can create such a application. We will start with steps on how you can create a business network using Hyperledger Composer. We will be talking about files which are used by Hyperledger Composer, composer playground and how you can create your business network.

We will go through steps on how you can deploy this network to a instance of the Hyperledger Fabric. We will show how you can install the composer on the fabric instance and start your
business network on the fabric instance.

Next we will talk about how you can create a angular app to interact with the network. We will look at yeoman generator which can create a skeleton angular app to interact with the network and how you can enhance that to create a full-fledged angular application to interact with Blockchain network. Then we will discuss some additional Hyperledger Composer components which are special case in this code pattern and finally we will talk about how you can extend this pattern to different ways.

Let's get started. First, we will take a look at the code pattern, which you can find at developer.IBM.com/go/patterns where you can find great patterns and under Blockchain you can find this code pattern called decentralized energy with Hyperledger Composer.

I want to start the discussion with the purpose of Blockchain which, code like IBM Blockchain is primarily to bring in different entities and different people together with a great level of trust in the transactions that they do among each other.

To me that is a core of Blockchain. Second, the purpose of Blockchain is the aspect of the shared ledger. Instead of every entity having to keep their own records, Blockchain provides a way where you can have a shared distributed ledger on which each participant can view and record their transactions and access. This application tries to showcase those expose surfaces.

Let's discuss about the application. Residents are producing energy through solar panels or other means, that you can sell their excess energy to their neighbors or to the utility companies, and then you will have utility on the network you can buy energy from. If you are interested in energy sustainability or climate change, you would certainly like this application, this code pattern and great example of how Blockchain can transform our world.

I want to mention coming from oil and gas industry, oil and gas industry leaders, there is this misconception that the world will run out of oil and gas, is that the [inaudible] stone age ended because we found better technologies, better ways to live. The same is true with, there is lots of positives in the technology [inaudible] great infrastructures for creating energy and other means of energy.

(faint and muffled audio).

Talking about the application, what this application shows is a network where you have residents who can sell or buy energy for coins at a predetermined rate. Then you have utility companies on the network, can also buy and sell energy for predetermined rates, exchange energy for points. You can exchange cash for coins. When I developed this, this was before
the [inaudible] the idea was participants will have balances where the residents having energy coins and cash and then they can create the purpose of banks was primarily to exchange these coins for some monitoring, cash that would be currency [inaudible]

We will see the application in a little bit. Before we look at the application, I want to go through the architecture and flow of this application. Here you have an admin. interacting with Angular Web UI which interacts with composer REST server which is on top of the Hyperledger Fabric.

One thing which, feedback I've got about adding admin. here, with Hyperledger offerings, it's a permission network and I mention it as a Blockchain as a service. When I was working on this code pattern, I envisioned government to be the admin. here, providing a decentralized network to residents and utility companies and creating infrastructure, where energy can be used and exchanged on the network.

Here you have Angular Web UI is being accessed by admin. who make requests through the composer rest server and which interacts with the Hyperledger Fabric and through this composer rest server you can retrieve this database and display your data on the Angular Web, and you can showcase the ledger.

Interesting code pattern list instructions, we will be going through the instructions. The repo will have the code for the entire application and the network, and it will have [inaudible]

Now let's look at the application. This is the application which is generates, created through this code pattern. You have step 1 is create and update participants. Step 2 is transactions. Let's take a look. Right now I created two residents. In their account they have coin balance, energy value, energy units, cash balance, cash currency.

Carlos has [inaudible] exchange energy, then you can perform a transaction between these two for coins. Before we jump into that, I want to mention that we have banks on the network which you can add a bank here. It shows their coin balance, cash balance and cash currency. You have utility companies and with their coin balance, energy value and energy units. Let's go ahead make a transaction.

We will send ten kilowatt hours from Carlos to John. We should see balance update. We will see the transaction on our ledger.

We will do R1, R2, exchange ten kilowatts. (mumbling).

Transaction ID, this is billing period. The idea would be that this would be, this transaction would be occurring every two time intervals. If you look at the balances and you see that this incremented by ten, here again same thing.
The application shows the Blockchain transactions here. One thing which composer provides is the transaction type. You can have a system transaction where you are adding participants and issuing identity when you create it. You have namespace for your network which shows the transactions you perform.

We will be looking into the file to see what these transactions do. This is recording all transactions on your Blockchain ledger.

One more transaction, to show Carlos has 990 coins he wants to exchange for cash balance. He can make a transaction with the bank. He will get cash for coins. This one bank on the network, go ahead and pick that. 5 cash value transaction. Then balance updated here. Carlos, he has more cash now, he create coins for cash here. Again on Blockchain we should see this cash to coins added.

This is the application. Now we will talk about the steps in creating Blockchain network, steps into creating Angular application. Now I'd like to introduce Hyperledger Composer. Hyperledger Composer is an extensive tool set of framework that makes developing Blockchain applications easier, in a straightforward way. One thing I like about it is in addition to making applications easier it keeps the purpose of Blockchain in front of you throughout.

When you are designing, you always have transactions, who the participants are. The composer consists primarily of, when you are designing your business network you want to start with your model file. This is the model.

You are happy with it. You can define participants. You can define assets. Assets could be tangible, intangible, it could be actual items or it could be different account values. You will have participants which are entity or organizations which are part of the network. Then you have transactions. Transactions will tell you, you are going to interact with your Blockchain network, how you want to update your assets and your participants.

The logic of the transaction is defined in the transaction functions. This is a JavaScript file. A lot of people are comfortable with JavaScript. It's a great way to define logic of your transaction. Then you have access control which define which certain, what access participants have. You can say participants have access to transaction 1, but not transaction 2. Those are defined through access control. Query definition is where you can define queries.

Starting with Hyperledger Composer I recommend you start with one file, script file, access control usually you can start with
default, like access [inaudible] once your application progresses, you can [inaudible] queries on the Blockchain database.

This will create a business network archive file, which can be run on the Hyperledger Fabric instance or on composer as well. We will take a look at composer playground. You can go to composer.playground.net and great place to start with composer. You can go to deploy business network. I recommend looking at these going through some of the samples. They have great samples, and demonstrates what Blockchain networks are, use cases and you can get familiar with it. Second [inaudible] composer documentation, great resource. You have in addition to the concepts you have great tutorials on developing applications, you have references on all composer. I'll be touching on a few which really helped me in creating the application.

This is what I would recommend, starting composer project, start playground. We are going to look at before looking at decentralized network I'm going to share this network to demonstrate what the files are going to look like and discuss some of the syntax around it.

This is [inaudible] start with a namespace. You define your asset, participants and transactions. At this point we are going to not talk about an event, another transaction, primarily these three I'll go through. Here we have an asset called commodity. You want to have your asset and participants identified by one of the attributes. If you have object limited background, this will come easily to you, straightforward, you have already properties that you can define. Then you can define relation between asset and participants over here as well.

Here, it seems like when you use this arrow, this trader is the owner of this commodity. What this is doing, you have asset called commodity. You have a participant trader. You have a transaction. What you are mainly doing is saying that through this transaction, change the owner of the commodity from one trader to the next trader.

Talking some of the syntax again, circles, its own syntax, you want to identify some ID, if you are doing users, you can always identify by like E-mail, phone number, and then you can perform transactions.

Let's quickly go through. You have this great feature of test, you can test transactions. Before we go there, I want to show the logic, JavaScript file where we are going to define this transaction.

Here you have the same function, whatever the transactions you list on your modified list will be defined in your logic
JavaScript file. We will focus on the straight commodity. What the function is doing, create commodity.owner equals dot new owner. You are passing the parameters to here and saying what is the commodity that is passed, you want to change the owner of the commodity to [inaudible] this provides participant registry. You can go through and update your asset registry and here you are updating commodity asset to say that the owner is the new owner.

This is very basic starting place for composer. Start with these three asset participant transaction and you can test the network. Here I have created participants. I'll create new participant. What this shows here, I want to highlight is JSON. (mumbling).

T3 and T4, then here you have commodity, T3 is the owner of this commodity. What you can do is your transaction, I want to do trade. This send tax wants the namespace and you tell it what commodity you are sending and who the owner is. You want the new owner to be T4. Do submit. And you see the transaction added to our ledger. We go through our commodity and see the owner changed to T4. It's very basic. You can create, whenever I see this I want to create transactions to update the quantity. You can start here and go through there.

I hope this gave a good overview of what, how composer develops networks through composer. Now we will look at the decentralizing file. Here we have our participant call resident. For resident we are giving attributes identified by resident I.D. That is first name, last name and signing a coins asset, cash asset and energy asset to the participants. You have bank participant on the network, bank ID, name and you are giving coin asset and associated with that. Similarly for [inaudible]

Next, defining my assets. For my assets I'm telling what primarily you want to ID, the value [inaudible] owner ID, owner entity. Similarly for energy we are adding units. The idea was we can add more units for like in different places and same thing with cash. You are adding, so network could be used in different places.

Then you have your transaction. You have two transactions, energy to coins and cash to coins. Energy to coins again you are getting energy rate, a value. You are passing the assets that you want the coins, where you want the coins to increase, the asset, where you want the coins to decrease [inaudible] this is creating between residents or create between resident and utility company. We would expect the producers coin and producer energy, their assets and consumers coin increase and consumer energy [inaudible] similarly for cash to coins, where you are transacting with the bank, asset of the banks and the
resident.

Next look at the transaction here, kind of straightforward. We are finding out how much the value of the coins will change. Then we are going to say that for increase to happen, do addition of coin change. We are updating accounts here and updating the value of energy assets here. This provides asset registry where you can go and update your coin assets and we are updating our energy assets. We are using promises you can add update all.

Similarly, the network coming here it's straightforward. One thing you want to mention, this is where you can add more logic to it. Creating application, you have decision-making where you perform actions on the application side, or perform actions on your transactions, within your transactions, so that is how you design your application. Next I want to share the provisions file. On the application, we can't really see it but I added it for demonstration purposes, was that you can do something with residents to access only their profile. We would update, they can have create. Then certain places, predominantly provide -- my idea was that only the assets be updated but the participants should not be [inaudible] some of the application update the asset here. Something you can come here and play around with. You have queries. The queries, about that you can use sequel style statements here.

For this one, select all resident. We will see later on, I want to ensure the utility companies does not have access to banks [inaudible] banks can only [inaudible] those are two added permissions.

This is the core of it, going to start with [inaudible]

(sorry, audio quality is very poor, very muffled and faint).

Now we are going to talk about deploying network to Hyperledger Fabric instance. Here we go to repo. I have already [inaudible] I want to mention that in the first part [inaudible] this will take five minutes. Then you can get your Hyperledger images. Start fabric. This will create Docker containers. We are going straightforward. You create a peer admin. card, interact with it.

When you are starting a fabric, it creates all the fabric, it takes care of all the fabric concerns. Here you have peer admin. Next what we want to do is create our business network archive file. Install, and I make a note that you have this composer archive create command which creates a file.

Here is composer archive create. You can use this command by itself. [inaudible]

(mumbling).

To create the new file, once this [inaudible] deploy to our
Once you get this (static) it means you are successful creating [inaudible] (distorted audio).

Installing composer runtime. This requires [inaudible] (sorry, audio quality has deteriorated, I can't understand).

There are a few [inaudible] next our composer [inaudible] start the network, we provide admin. and [inaudible].

We are using admin. [inaudible] specifying [inaudible] for participants, that process, admin. can issue cards for participants. Then they can access [inaudible] network is running.

Next we are ready to [inaudible] your business network archive has successfully [inaudible] next we can go ahead and [inaudible] Angular app.

This will take a minute or two, install dependencies [inaudible] while it's doing that, we will start our discussion around Angular apps that we have. Angular apps [inaudible] (sorry, audio is extremely muffled and faint, I can't understand).

(lots of static and whirring noises).

Every stage we saw in the application [inaudible] get more familiar [inaudible] interact with [inaudible] you can say update resident. Load all. [inaudible] (if he could use a microphone, that would be great).

Start working on [inaudible] (I'm sorry, I can't understand anything).

Also include here, once you have [inaudible] you can start by [inaudible] before we do that, create a [inaudible] now the process, going to create assets first and then go ahead and create [inaudible] assign assets to [inaudible] some functionality you can control. [inaudible] update asset, name and ID which you can do from here. Similarly [inaudible] you can start [inaudible] you can start over here, applications, show [inaudible] you can make a post command. On our Angular site [inaudible] this is where [inaudible] it's making [inaudible] add, update. Here, where you can create [inaudible] then compare both apps.

[inaudible]

Angular app, composer ... we want to connect ... [inaudible] this will create [inaudible] once you have your network, Angular apps [inaudible] generate. This will provide us with the data service, have function, then you can update interaction through data.service file. [inaudible] two functions, transactions [inaudible]
Looking at [inaudible]
Looking at [inaudible]
Connected to [inaudible]
This is Angular app, coins [inaudible] you can take that, you can start [inaudible]
Additional components I want to talk about, you can create a test file. [inaudible] with this highlighted, you can use composer to create test cases. Here I'm using test case, can create a full-fledged application. Configure admin., configure [inaudible] you can create the whole [inaudible] I'm going to be setting up my entire network and then performing [inaudible] coin assets [inaudible] more familiar with composer [inaudible] update based on transactions.
[inaudible]
You can create test. Here what I was doing, I'm checking to ensure that all my transaction are working as expected. I get my assets [inaudible] you can work, the next thing I show here is, utility company transaction, bank transaction, and then, what I want to show [inaudible] access to all coins, should have access to everything, and [inaudible] query functions. [inaudible] actually using queries. The queries are straightforward [inaudible] you can explore more.
[inaudible] shows all the [inaudible] and you can do that. [inaudible]
I want to mention in terms of historian, here you can access, you can perform queries for database and see how all the transactions that you have on the ledger [inaudible] finally, extending the pattern. One thing [inaudible] I wanted to share [inaudible] provide you with a very straightforward steps to network to IBM. It goes through steps, adding to BlueMix, you can set up a cluster, and eventually create [inaudible] to use the cluster and then [inaudible] for Blockchain. You can install [inaudible] on to this cluster.
Provide some great scripts here [inaudible] Blockchain [inaudible] I was able to deploy my network on to IBM Blockchain. Then you can further application as well which we talk about. Once you deploy, you get [inaudible] interact network through this [inaudible] interact with network through this, expose your composer to [inaudible] next steps that I did was [inaudible] updated this configuration file to have IP address, updated the JSON file, to run it at public IP address. Then added [inaudible] configure the port so that you get the correct [inaudible] I was able to run the app, so this application is running in the Cloud with the network running in the Cloud.
Once you have the network created [inaudible] to the Cloud,
cluster, you can [inaudible] you can create specific permissions and participant access. Where you can do that, create cards for different participants, how access the application, you can create [inaudible] those participants, you can integrate [inaudible] realtime transaction [inaudible] it's all manual, but the idea behind it was that this would be like realtime, after a certain [inaudible] it would update account balances on the ledger on the realtime basis.

At this point, I end my presentation. If there is any questions ... 
>> Are there any questions in the chat? Are there any questions in the chat we should address.
>> Laura has been addressing the questions in the chat.
>> I was on mute. I'll go back to a earlier question which I had not [inaudible] (background noise).

Possible that the Angular app can integrate directly with composer through some API JDK instead of the composer rest server? He clarified, to further saying some kind of JDK like Node.js using HOSSDK, I don't know if you were aware of any use case like that.

>> RAHEEL ZUBAIRY: I'm not aware of use case particularly, but you can use composer SDK to design apps, and this would be not only Angular application it could be any Web interface application.

>> There is some discussion around, I was starting to answer it, maybe you could expound upon it further about using Blockchain internally, and then specifically around compliance and that ties into contracts and security policies. How can you apply Blockchain to that?

>> RAHEEL ZUBAIRY: For compliance, I think Blockchain can be great way where you can record all your transactions on the ledger, and then you can provide the regulators with a read access where they can look at all the transactions and they can query those transactions to ensure that the transactions amongst entities were compliant. That would be my understanding. But again it would depend on the particular use case as well, on what type of transactions you want compliant but Blockchain would be a great resource for any compliance related use cases. Then you can provide read access to the auditor or the regulator, and they are not changing the network. They are not making any change except only thing they were doing is they can view the data and network can ensure that it's compliant.

>> Okay. Someone goes into, maybe you can go more detail how a block is verified on the chain. Is this IBM handling? To let people know, Hyperledger is part of the open source foundation. IBM is a contributor to it. But IBM does not own Hyperledger per se. We help in the development process. But Raheel if you
want to expound further on talking about how blocks are verified.

>> RAHEEL ZUBAIRY: Yes. Kind of mention around when composer that you have your script file, and the JavaScript where you place your logic for each transaction, this is where you would have logic behind the verifications on ensuring that you are making the correct transactions. I kind of touched upon this, energy [inaudible] performing a lot of [inaudible] on the application side and not on in my JavaScript, again how your architecture application as well. But you would want to have checks, you can place the checks in the Blockchain smart contract in the JavaScript file.

>> I'm curious on your opinion, I responded but I'm curious on your point of view in terms of someone asked about the real value for Blockchain. I know in my experience a lot of people debate between using databases versus Blockchain. I'd be curious to hear your response is the true value.

>> RAHEEL ZUBAIRY: My view, any place you have multiple entities and you want to have a greater level of trust between the entities and the transaction, that is where you would want Blockchain. Instead of, there are any cases where you have two sets of recordkeeping going on, you can bring them through Blockchain ledger and that would be, in my opinion, the greatest use case for Blockchain, is bringing entities together to ledger and creates more accountability, transparency, compliance, it's great, you can select your controls.

>> Someone posted a question and is it possible to transfer assets between channels?

>> RAHEEL ZUBAIRY: Channels would be again, that would be more on the Hyperledger Fabric side, if you are creating different channels, but again for composer, the way it's going to be is through permissions file. You can say send participant has access to certain assets or they don't have access to certain assets, so that is where your permissions define what permission level a certain participant has.

>> I don't see any other questions in the queue. Let me finish up quick, just posted something, talking about -- (mumbling).

Okay. I can respond to this. I don't see any other questions in the queue. I don't know if you want to close.

>> KATHY GHANEEI: Okay. Marc-Arthur has computer problems. He is not able to speak. I'll go ahead and close out. Thank you so much, Raheel, for this presentation. Thank you, Laura. We will post the recording to the YouTube channel as soon as this ends and we will post the chat transcript soon after. With that, we will end the meeting. Thanks, everyone.

>> RAHEEL ZUBAIRY: Thank you.
>> Bye.
>> Thank you.
>> Good-bye.

(end of call at 12:57 p.m. CST)