**Salesforce DC**

0:02  
And once again ping the plan Zira Division.

0:05  
I have been the

0:23  
Okay, can you just see my screen?

0:26  
Yeah, we can say it. Okay.

0:29  
Alright, so Salesforce, Duck Creek has 2 flows.

0:36  
So one flow is basically

0:39  
receiving information from sales force and then it is sending it to Duck Creek

0:48  
and then the second floor is receiving information from Salesforce and it is updating Dragon.

0:57  
Okay, so lets just go over the first floor Duck Creek sales force.

1:03  
So in meals, Sir, we have this component for Salesforce where we can connect directly like this is a platform events component. So on the sales force side,

1:14  
it is configured as a platform events where

1:18  
any update happens on like certain actions it will trigger an event and send a request or or publish an event to this platform event. And Mulesoft is kind of always listening to this platform event. So whenever meals of receives a request on this platform and it will try to

1:42  
process it.

1:44  
And what this one basically does is like from Let me go to Mulesoft logs.

1:54  
Hi, Chetan.

1:57  
Yes. Chetan, before going to directly jumping into the court, could you please show us like the repository now which one you have or took this one

2:09  
and the depository prospective?

2:12  
Yeah, I mean it. We will eventually review it, right?

2:42  
This is the repository,

2:46  
what is the brand should we choose like a DA word? Which one is a more stable

2:53  
UAT?

2:57  
Okay,

3:06  
say you can to choose like. I think staging a UAT should be fine. Both has you

3:13  
know you can also review like the last commit right? But eventually UAT will have the changes what is in staging. So you can choose staging or UAT.

3:23  
Okay, we will take a stage then.

3:28  
Okay,

3:33  
alright, so some announcements are going on on this service Chetan, because I see its been updated 5 days ago.

3:43  
Yeah, I think Muhammad is working on some changes for Canada.

3:47  
So I am not sure like so when are we planning to work on this?

3:55  
Do we know

3:59  
is 4th service?

4:01  
What is the 3rd one? Yeah, 3rd one. This is still it will take some time.

4:09  
Okay, so I mean, yeah, we will. We will have to coordinate with

4:13  
Mohammed and all. I think Mohammed mainly because he is the one who is going to work on this.

4:19  
I am not sure whether there any additional changes he is planning to do. Maybe not because I think he just mentioned just adding some currency code for Canada in the request.

4:34  
So we can, we can check with him, but I think we have time before we can even start working on this, right?

4:45  
Yeah,

4:46  
OK.

4:50  
Alright.

4:53  
So what we are doing here is like whenever there whatever payload comes from sales force, right. So let me just show you the payload how it looks like.

5:03  
So

5:05  
Salesforce will send like 2 type of request.

5:09  
If you see here, there is like a update Salesforce broker and then one would be create Salesforce broker and for both of them

5:18  
this payload is mostly similar

5:21  
except the action is different.

5:25  
Okay,

5:28  
so once we receive the request,

5:33  
we are just trying to capture the information

5:37  
and I think I am not printing the data dot payload. I think we are. So whatever comes from Salesforce is different. This is how it comes from Salesforce,

5:51  
right? So we are basically reading that payload from Salesforce and like capturing some variables from this payload

6:00  
and then we are doing some modification. We are removing some special characters like like sorry not special. I would like line separate right like that we are removing

6:13  
from the request we are receiving because when we send it to dakri there were some errors because of this line breaks. We are just kind of replacing with like an empty string

6:27  
and then in the payload there will be a this Duck Creek status as well. Based on that we have a choice where we are doing as if the Creek status is null or the Creek status is CREATE failed,

6:45  
then we are

6:48  
creating a request for CREATE sales force broker action.

6:53  
And then we are just creating this payload based on the data we have from sales force.

7:02  
Okay. And if the

7:05  
Duck Creek

7:08  
Pillur Duck Creek status is not null

7:12  
then we are sending the update Salesforce broker request.

7:17  
So if it is null or create failed,

7:20  
then we are sending

7:25  
this request create Salesforce if it is otherwise we are sending the update broker request.

7:36  
Okay. And then we are just like printing this payload and this flow references like this is for sending it to Duck Creek.

7:45  
If we come back here, this is sent to the Creek. So first I think we are calling is the Duck Creek endpoint.

7:54  
This is this is going to be a meal soft endpoint.

8:00  
Sorry main resources.

8:04  
So it's calling the Mulesoft endpoint but I think when when we are calling a Mulesoft endpoint from outside it will be an A Pi internal end point. It wont be this mule worker. This is only 4 when you are calling a Mulesoft app within Mulesoft itself.

8:21  
But if you are calling it from outside, we will use the API

8:26  
internal flow

8:28  
or API internal URL

8:31  
which we can share I think.

8:34  
And then there are some configuration for Salesforce like username, password and then the URL for the environment. And

8:42  
we also have configuration for the Dragon database that is for the second use case what we will review.

8:52  
So we will send the request to Takreek, we receive the response then we are transforming the response

9:02  
where we are capturing the status and the error

9:07  
in the

9:12  
from the response what the Creek sends.

9:15  
And then

9:18  
we have a choice

9:22  
after where we are saying if the status is success

9:25  
when we have to do like an operation in the Creek where we have to update certain field saying this broker was created in

9:36  
Duck Creek. So we are basically sending. We are

9:42  
creating this payload

9:46  
where we are sending the broker ID

9:50  
update timestamp as like current timestamp and then the update success

9:57  
and then we are calling this Salesforce component which is doing an upsert operation on this account object type based on this broker ID field. And then whatever payload we send this is the payload we need to set in order to do that offset operation. Now we have to figure out like if we are going with

10:24  
ECS then how are we going to have a Salesforce integration like are we going to build a sales force component which will have all the operations available for us. So thats something we will have to

10:37  
see as part of this

10:39  
and then we are just getting the response back

10:45  
from from sales force.

10:50  
Okay. And then

10:52  
here I think if it is

10:55  
38 failed or null

10:59  
the dakreek status is create failed or not then we are doing this.

11:05  
So okay in the success then we have another choice router

11:09  
where we have create failed

11:13  
Arnal

11:15  
where our status was CREATE failed or not then we are doing a update success. That means this is not a new broker

11:23  
if it is not. If it is a new broker

11:30  
then also we are doing the same like we are checking some condition. I mean I dont know why we have these 2 but there must be some reason we have done this

11:40  
update success and then we are sending the payload and here also update success in the payload and then here another we go back to the original choice where we have status as failure from the Duck Creek. Then we are we again have like some

12:00  
error messages we are parsing from the Dakreek response where Duck Creek will say create status will be CREATE failed or null and then the payload will have this response that OK the broker already exists. So in that case we are kind of setting the status as create failed as a duplicate because this we are trying to send a create request but the broker already exists.

12:26  
So we are saying create is failed because it is a duplicate. Technically from Salesforce side if it is a if the broker is already created in Dakri, they should always send the update request not the create request. But there have been some issues in the past in sales force where they were sending like

12:46  
create request for an existing broker because there was the sync did not happen or some some issue.

12:53  
So thats why we have this conditions here and then there is another condition

13:00  
also same thing like it just basically

13:04  
has create failed

13:08  
and then this message here it says create failed and

13:14  
different error code. I am not sure what is this.

13:23  
Yeah I am.

13:26  
I'm not sure why it is duplicated, but I am not sure if Mohammed is working on this. Is CREATE failed a Creek status

13:39  
Karnal and then not equal to OK, yeah. So that 2 things OK,

13:44  
this one says equals. The error is the

13:48  
reference already exists. Okay and this one says the broker does not exist. Then we are sending. We are setting the status as create failed

14:01  
and then default we are saying update failed.

14:06  
This is different condition. I think we need to capture

14:10  
and then same thing like there is an error handling we are doing

14:16  
on the different statuses from Duck Creek. We are just sending

14:21  
create or update failed to fails force.

14:27  
This is pretty simple workflow. Just like multiple conditions, we need to check

14:36  
put this any, any questions on this?

14:43  
No, no, no, Chetan. See, I wanted to know exactly the functionality, what it does actually how, how in the pure insurance, what is the contribution of this particular service.

14:57  
So I mentioned right like from from sales force.

15:03  
So whenever a new broker needs to be created in

15:08  
the Creek, it always comes through sales force. Like in the Creek, we are not creating brokers manually. So Salesforce is our platform for broker creation and management.

15:21  
So as part of a workflow in Salesforce, when they are in certain screens, when they

15:30  
take some action or click some button, as part of that

15:34  
we want to create or update a broker in Duck Creek,

15:39  
right. So that's the whole business case,

15:44  
creating and managing broker from Duck Creek to sorry, from Salesforce to Duck Creek.

15:50  
Sorry to ask like what is this? Dakri

15:54  
Creek is our billing system.

16:01  
So like if anyone is a purchasing insurance, so while the time of purchasing whatever the bill amount will get generate that will comes under the decree.

16:11  
Yeah, that is right. OK.

16:15  
It belongs to like it might be a house

16:20  
or vehicle or

16:23  
upon like what type of the insurance. He is the person,

16:29  
right?

16:31  
Yeah.

16:32  
So all the billing management happens through Duck Creek for for pure.

16:40  
Okay,

16:46  
thank you. Any any other questions on this?

16:57  
Nope. Okay. And then I will move it to the second floor for Salesforce drive. The the previous flow, there is no business logic or anything based on the event that we are just distributing based on the status that we will see based on the status you are just setting the pending. So basically what event Sir, whatever payload we are generating here, right, that is being sent to Duck Creek as a web service

17:21  
and then whatever response comes back from the Creek, we are just

17:27  
like parsing it and then based on the status we are just sending different statuses back to sales force.

17:38  
Okay,

17:41  
Now this is the Salesforce Dragonfloe. So what happens in this one is as I said like all the broker management happens in sales force and

17:53  
for each broker we have commissioned year which gets assigned. That means like lets say if broker A is writing a policy for us,

18:05  
so for broker A, the Commission, whatever policy is there right? Whatever premium we get right out of that like 10% of the premium we paid as a Commission to the broker

18:16  
and each broker has different Commission tiers

18:21  
and based on the tier, the percentage get defined. So this tier management is

18:28  
within Salesforce.

18:31  
So and then

18:34  
those tier assigned to those tiers have some percentages and based on that percentages

18:43  
in Dragon when we write a policy we basically

18:48  
fine, OK, what should be the Commission for the broker.

18:55  
Now that used to happen in Dragon before we onboarded Duck Creek.

19:00  
Now the Creek manages all the tier calculation everything, but we still want

19:08  
the tier information in Dragon for each broker.

19:12  
So this tier information comes from like this is manage in sales force. And we want to sync this tears in Dragon as an information on the on the UI. So what this flow does is it it runs on a scheduler. I think it runs every day at 10:00 PM

19:32  
Eastern Time. Chandragini ko mute or someone's talk. Yeah, okay.

19:38  
It runs 10:00 PM each and everyday. And what first step is does is

19:44  
yes, it reaches out to Dragon

19:47  
and it invokes this to procedure which is I think creating a user session

19:56  
because for every dragon call we are, we are kind of trying to have like a valid user session.

20:05  
So it just like a Stow procedure call whatever comes back from dragon, we store it in a variable,

20:12  
we will we will use it in later point. Now what this step does

20:19  
is

20:22  
basically it is getting

20:26  
all the broker information

20:29  
with the status active.

20:33  
This is the Salesforce query I think where this is the record type ID for broker and then we are getting a name broker ID, admitted, Commission tier

20:47  
and non. So we have like pure has like 2 different line of business like admitted and non admitted. And for each it is it has different tiers assigned and those tiers are shown. I can just show you

21:24  
see here assigned tier Salesforce standard one. So this is where it is basically showing on the UI and this is where

21:34  
this is what it updated that Muleshoflow updates.

21:39  
So it will receive the list of brokers. Then for each broker

21:45  
what we are doing is

21:49  
we are calling this store procedure

21:52  
and this store procedure basically will

21:55  
update the tier. So we are using this session ID here. We are using to call that so procedure and then we are passing the broker ID admitted Commission to your description and non admitted Commission to your description. And this to procedure basically updates

22:15  
at the broker level in Dragon.

22:18  
And if this process is completed successfully, then it sends out an success email.

22:25  
And yeah, right now it just goes to me and Meenakshi. It may need to update when you guys are working on it. Just let me know. I can reach out to Meenakshi and ask what it should be updated to. And if there is an error, if any error happens during that flow, it will send out a flow like I mean error email message.

22:47  
It's pretty simple, I mean

22:51  
reaching out to sales force, getting the list of brokers and then for each broker calling duck sorry call calling a Dragons to procedure

22:59  
and dragonstore procedure will do its thing. And then it once the flow is completed, send out an email.

23:10  
Okay, so

23:12  
well there this is the combination of a Pi and bad job, right?

23:18  
There is no API here.

23:21  
I mean, in the whole, yeah, this is just calling a Dragon database directly, not in this flow, but Oh yeah, yeah, the whole application, yes, that's right. Yes,

23:31  
again, like there is no end point here, right? There is no endpoint, you are exposing

23:38  
whatever is happening. Its like in the back end like this is a Salesforce platform event like which we are kind of subscribing to from from Mulesoft.

23:50  
OK as a as a as if like an MQ subscription, yeah, consider that as an M yeah similar to that like it is some sort of like you want this trailforeside.

24:05  
So I think for for AWS I am not sure like there is a way

24:12  
there is a

24:14  
we have done some POC for this right Bhanu like subscribing to Salesforce publish events from a WS.

24:22  
Have you used like event bridge for that or

24:29  
Udhar Bhanu?

24:35  
Yeah, sorry I was on mute. Yeah, we used even bridge and we are getting very much event bridge to kind of get the events from sales force and then kind of invoking either Lambda function or something right in in the function in AWS. Yeah, Lambda function. So we need to expose this as an endpoint right? For sure. Because for AWS to trigger, if you are deploying this as an ECS container,

25:02  
then we need to expose an info in so that we can trigger from the email range.

25:08  
Yeah, I mean, I am not sure if this is like does does this is pretty simple, right? Like should it be an ECS container or this can be done in Lambda? This can be done in Lambda

25:20  
because this is very simple and like yeah payload size is very small. The response time is like pretty small.

25:31  
How about the scheduler?

25:37  
We can have scheduler part like

25:40  
yeah that will I am not sure it will probably call like some. I think you most of the time this process like finishes like let me see, I do get an email right. I think it finishes

25:59  
10:00 PM right? Like I think within like 4 to 5 minutes,

26:04  
OK, you are talking about of this whole project, we dont want to go with the ECS

26:11  
if it is not needed, right? I am not sure like this is also like pretty simple flow, it finishes like 4 to 5 minutes everyday

26:22  
right? So

26:24  
I think this can be done in Lambda as well, right? Lambda is like what, 15 minutes limit?

26:32  
Yeah,

26:36  
I'm just looking at the history, right? I mean every

26:39  
everyday I see its finishing like a 10 or 4 kicks off at 10:00 and finish at 10:00 or 4.

26:51  
So, yeah,

26:53  
I'm not sure. I think this also once a day. Let's schedule a schedule for. Yeah, once a day. Yeah, if at least once a day. But the events can be n number of events, right? Yeah. This one, this one can be like anytime during the day. Yeah. So this could be event bridge calling Lambda again. This one also can be event bridge calling a Lambda function,

27:17  
both of them

27:20  
I think. In the Lambda no we need to create this event please and expose it call it. This all the functionality. I think

27:33  
you need to create an event bridge which will invoke a Lambda function.

27:38  
Yeah, yeah.

27:40  
I mean to say,

27:42  
yeah,

27:44  
OK.

27:47  
Any other questions

27:50  
that we were thinking about going with Node?

27:57  
Yeah, we can go with this for this. Yeah, this will be better for Node JS.

28:02  
Yeah, I would actually

28:06  
send this question to the received platform events, right their Mulesoft connector for this app. I would send an email to Salesforce team and get confirmation that we can follow the same way we are using that Amazon even Bridget a WS one bridge

28:23  
that project is called as SPN project. So we can ask Shruti from Salesforce team whether we can follow the same option for this. So let me create a group chat and then ask that question.

28:41  
Okay,

28:45  
alright. Any, I mean did you guys get a chance to review the Smartcom like any questions on Smartcom yet or not yet?

28:56  
Not yet.

28:58  
So OK, one question was OK for the smart COM

29:05  
is that we are storing a token in any cache and all for validation or moving forward. See for a smart community, for a smart COM we are generating a token right? With the help of the token, we are a processing and calling those API calls and all right. So the token validation and all we are a checking

29:25  
in this smart COM.

29:28  
This is not you are not generating token like new token for every request right? I think

29:34  
its you are not reaching out to any service to get that token. You are just generating

29:42  
or or

29:45  
what we saw in the code in in Mulesoft. We may not need to do that way like there might be easier way using Java and thats what we had before.

29:58  
We should probably look for the other option rather than like complicating things for like. I think if I remember could I told you right like we use Jersey client

30:10  
to connect to the smart COM web service. I am just trying to let me just go back and like try to find the code what we had before

30:22  
we moved to like

30:26  
Mulesoft.

30:33  
It was. It was simpler than it looks in Mulesoft

30:43  
OH

32:52  
web resource. I think this is what

32:58  
for using SO

33:07  
rupee Sixtyfour

33:11  
10,

33:14  
we are clear creating this client

33:20  
and we have this URL.

33:25  
Yeah, I think this is what we were

33:28  
doing. And then all these things like all the information we receive,

33:35  
API key and all, just trying to see where we are using it,

33:49  
Peggy said. Secret

33:57  
using it somewhere.

34:01  
You will have to review it and I will try to find it. But I think we can we can do it easier way

34:08  
right now. What I thought no Chetan with the help of I think OK, Okta Togan generation, right. With help of that we are getting a token and once the token will get there is no there is no Octa token associated with this when calling the Smartcom service.

34:26  
Okay

34:29  
it has its own

34:36  
way of calling it

34:39  
okay and and and okay so for those token right we have a expiry time and all and storing that particular token sum amount of that time under 1I say to all those things you dont need to do all those things. I think you are you are overthinking it.

34:55  
This has nothing to do with Octa or anything so

35:03  
I am trying to find. I think someone should have like a postman collection. Postman let me see Bhanu did we

35:11  
do we know like the direct are there like any postman collection stored for like direct call to smart com and we have something? I do have it, yeah.

35:23  
Yeah, I am just trying to find.

35:31  
We should probably just try to like, reuse that and mimic it instead of.

35:37

36:05  
Someone should maybe like Lib in or constant libin I am not sure. Constantin maybe, I am not sure

36:17  
Actions

36:20  
red

36:27  
it's a local

36:38  
I'll. I'll try to find

36:41  
the code and see if we can

36:45  
make it work,

36:47  
but basically it is using Oath to authentication

36:53  
so I would say follow how the Oath 2 usually works rather than

37:00  
trying to make it complicated

37:02  
in Java. OK, OK,

37:11  
okay.

37:16  
Anything else for today?

37:20  
No, thank you.

37:22  
Okay, thank you, guys.

37:25  
Yeah, thank you. Yeah, bye. Thank you very much.

37:28  
Goodbye.