we have this component for Salesforce where we can connect directly like this is a platform events component. So on the sales force side, it is configured as a platform events where 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 process it.And what this one basically does is like from Let me go to Mulesoft logs.

whenever there whatever payload comes from sales force, right. So let me just show you the payload how it looks like.  
Salesforce will send like 2 type of request.  
If you see here, there is like a update Salesforce broker and then one would be create Salesforce broker and for both of them this payload is mostly similar except the action is different.

so once we receive the request, we are just trying to capture the information  
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, right? So we are basically reading that payload from Salesforce and like capturing some variables from this payload  
and then we are doing some modification. We are removing some special characters like sorry not special. I would like line separate right like that we are removing  
from the request we are receiving because when we send it to DuckCreek there were some errors because of this line breaks. We are just kind of replacing with like an empty string  
and then in the payload there will be this DuckCreek status as well. Based on that we have a choice where we are doing as if the duckCreek status is null or the DuckCreek status is CREATE failed,  
then we are creating a request for CREATE salesforce broker action. And then we are just creating this payload based on the data we have from sales force. And if the DuckCreek  
Payload DuckCreek status is not null then we are sending the update Salesforce broker request.  
So if it is null or create failed, then we are sending this request create Salesforce if it is otherwise we are sending the update broker request. And then we are just like printing this payload and this flow references like this is for sending it to DuckCreek.  
If we come back here, this is sent to the Creek. So first I think we are calling is the Duck Creek endpoint.  
So it's calling the MuleSoft endpoint but I think 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.  
But if you are calling it from outside, we will use the API internal flow or API internal URL which we can share I think. And then there are some configurations for Salesforce like username, password and then the URL for the environment. And we also have configuration for the Dragon database that is for the second use case what we will review.  
So we will send the request to DuckCreek, we receive the response then we are transforming the response. where we are capturing the status and the error in the from the response what the Creek sends. And then

we have a choice after where we are saying if the duckcreek status is success when we have to do like an operation in the Creek where we have to update certain field saying this broker was created in Duck Creek. So we are basically sending. We are creating this payload where we are sending the broker ID update timestamp as like current timestamp and then the update success   
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  
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  
see as part of this and then we are just getting the response back from from sales force.  
 And then here I think if it is failed or null the duckcreek status is create failed or not then we are doing this.  
So okay in the success then we have another choice router where we have create failed where our status was CREATE failed or not then we are doing a update success. That means this is not a new broker if it is not. If it is a new broker 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 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 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. 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 create request for an existing broker because there was the sync did not happen or some some issue. So thats why we have this conditions here and then there is another condition also same thing like it just basically has create failed and then this message here it says create failed and different error code This is different condition. I think we need to capture and then same thing like there is an error handling we are doing on the different statuses from Duck Creek. We are just sending   
create or update failed to fails force Just like multiple conditions, we need to check  
So whenever a new broker needs to be created in 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. So as part of a workflow in Salesforce, when they are in certain screens, when they  
take some action or click some button, as part of that we want to create or update a broker in Duck Creek, right. So that's the whole business case, creating and managing broker from Duck Creek to sorry, from Salesforce to Duck Creek.