NGCC-Digital Sales Fusion Integration Overview

OVERVIEW

Currently NGCC is integrated with Oracle digital using REST API for Customer Connect CRM,

Oracle Digital is getting migrated to Fusion CRM as Digital Sales. This will be replacement for Customer Connect,.

Fusion supports Computer Telephony Integration (CTI) partners to integrate their media toolbars with Oracle B2B Service. NGCC communicates with Digital sales(CRM) using MCA API's via Toolbar.

NGCC will integrate with Fusion Digital Sales for CRM Integration using MCA API's via Toolbar.

For Phase 1 ,

- Both agents and supervisors will use Digital Sales and WWE i.e. side by side usage of Digital Sales toolbar and Genesys WWE for agents and supervisors
- This will be CRM integration from NGCC using MCA API's which enable's their Computer Telephony Integration (CTI) partners to integrate their media toolbars with Oracle B2B Service.
- All Field Sales Reps Globally OUTBOUND CALLS ONLY
- Pilot Group of North America Oracle Digital Reps - Outbound Calls Only
- Pilot Group of North America BDCs - Outbound Calls Only
- Global Oracle Digital Reps - Outbound using CUSTOMER CONNECT
- Global BDCs - All Field Sales Reps Globally OUTBOUND CALLS ONLY

This includes below functionalities

- Caller ID and Click To Dial
- Call event stamping for Outbound call initiated from Digital sales

SOLUTION DETAILS

NGCC communicates with Digital sales(CRM) using MCA API's via Toolbar which will be embedded as iFrame in Digital sales.

1. Create new war file(**ngcc-modelx-fusion-ig-v1.war**) and deployed on Web Logic server.

   War file contains :

   - NGCCFusionToolbar.html i.e. Basic toolbar with phone icon and Agent Name and will be embedded as iFrame in Digital sales.
   - NGCCFusionMCAAPI.js i.e. JavaScript function to invoke or register MCA API's.
   - NGCCModelxFusionWWE.html i.e. Html page load for LOB ZZZ agent group configuration.
   - NGCCModelxFusionWWEEventHandler.js - JavaScript function for Genesys call events handling.
   - wwe-service-client-api.js(shared by Genesys) for Genesys event handler

2. Create new war file(**ngcc-modelx-fusion-event-stamping-ig-v1.war)** and deployed in web logic server

   - Exposes REST API (/sendcallevents)to write Genesys event details to Apache Kafka which is used as message Broker. This REST API will be invoked by WWE custom HTML NGCCModelxFusionWWEEventHandler.js
   - Exposes REST API(/receivecallevents) to read Genesys call event details from Apache Kafka . This will be invoked by Fusion Toolbar (NGCCFusionMCAAPI.js) so that Toolbar invokes MCA API to do call event stamping in Fusion

3. Create new war file (**ngcc-modelx-fusion-configuration.war)** and deployed in weblogic server.

   - Exposes REST API (/configuration)to fetch agent configuration for fusion as per the LOB requirement. This API will be invoked by (NGCCFusionMCAAPI.js) and also by NGCCModelxFusionWWEEventHandler.js to fetch configurations like set of disabled features on fusion, channelname, appclassifiction, channelType using agent email Address.
   - REST API reads the agent configuration like LOB and BU from RPT_LOB and RPT_BU using GWS API and does Database lookup to fetch the disabled feature list,channelname, channeltype and appClassification.

NGCC Tool Bar HTML Load and Init Functionality

NGCCFusionToolbar.html file gets loaded in Fusion application as an iFrame with MCA profile options configuration in Fusion settings to enable NGCC Toolbar.

1.On load of NGCCFusionToolbar.html, load the MCA Javascript dynamically. Sample Code For MCA API's

Note: use oraApiPath to find the MCA API's java script function src to invoke other MCA API's

2. After Page load, it invokes initAgentToolbar() java script method with below MCA API's one after the other.

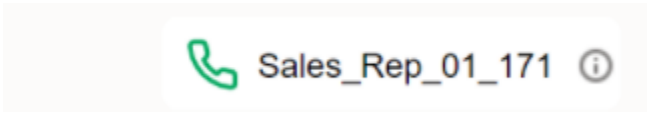
- In this method get the toolbar configuration using getConfiguration() MCA API which sets up toolbar.Call getConfiguration), which is mandatory method to get configuration information with config type as 'ALL' .
- Invoke REST API ngcc_webservice/fusionconfiguration/v1/configuration?agentemail=<salesRep@oracle.com> to fetch Fusion configuration.
- Invoke MCA API `disableFeature() to disable features which are retrieved with /configuration REST API . For example for outbound agents disable INBOUND_CALL,"TRANSFER_CALL" and "CONFERENCE_CALL".` Sample Code For MCA API's

- Invoke MCAP API readyForOperation(). This method notifies B2B Service that the toolbar is ready for operation. The toolbar is disabled by default and is enabled when this method is called with the readiness parameter set to TRUE. Sample Code For MCA API's

3. In Toolbar UI, it will have phone icon(in grey) with Not Signed in Text by default. Agent needs to click on this phone icon to make it active and available. Phone icon color changes to green and Not signed in text gets replaced with Agent Id as received in getConfiguration() method. This is required to before agent does any clicktodial from Fusion or to see screenpop for inbound call.

By default Toolbar UI looks as below

After click on Grey phone icon



When agent clicks on phone icon which is grey in toolbar, toolbar registers with B2B service for event listener.

The event listener API allows the toolbar register listeners to track events that are triggered by B2B Service. This allows B2B Service to initiate outbound interaction events and send updates to the toolbar.

Below are the set of Event Listener API to be invoked by Toolbar: Sample Code For MCA API's

agentStateEvent()	Notifies B2B Service of changes in a user's signed in or availability status for the specified channel.
onDataUpdated()	Registers a callback for customer data update events to be transmitted to the media toolbar with the updated information.
onOutgoingEvent()	Registers a toolbar-initiated callback for the start of an outgoing event triggered from B2B Service. On receive of this event, it needs to invoke NGCC REST API for Caller Id fetch and Clicktodial REST API here as mentioned in ClickToDial and CallerId flow. **Note: All outData object attribute of this method response to be used to be used to set inData object Attribute for newCommEvent(),startCommEvent(),closeCommEvent(). ie. `SVCMCA_ANI,SVCMCA_DISPLAY_NAME, SVCMCA_COMMUNICATION_DIRECTION,SVCMCA_CALL_ID,SVCMCA_CONTACT_ID,`** **`mcaOrigEventSource,mcaWindow`**
outboundCommError()	Notifies B2B Service that an error occurred during initiation of the outbound event.
onToolbarAgentCommand()	Registers a listener with B2B Service to provide agent control functionality.
onToolbarInteractionCommand()	Registers a listener with B2B Service to provide interaction control functionality.

4. Write javascript function in Toolbar Html js for toolbarNewCommEvent(),toolbarStartCommEvent(),toolbarCloseCommEvent() which is to be invoked as per the call event details fetched using REST API(/ngcc_webservice/fusioneventstamping/v1/receivecallevents?agentemail=<salesrep@oracle.com>&ngccuid=<12345qwer>&topic=<kafkatopic>

- toolbarNewCommEvent() function: It receives parameter like channel,appClassification,channelType,eventId,GENESYS_CALLUUID, timestamp ,eventType ,AgentExtension,NGCC_UserName,customernumber, commDirection ,callduration

It invokes MCA API newCommEvent() with below parameters:

```
svcMca.tlb.api.newCommEvent('PHONE', 'ORA_SALES',GENESYS_CALLUUID, inData, null, function (response) {},'ORA_SVC_PHONE');

i.e svcMca.tlb.api.newCommEvent(channel, appClassification,eventId, inData, null, function (response) {}, channelType);

construct inData as below:
```

```
inData.SVCMCA_ANI = <interaction id received in on OutGoingEvent()>
inData.SVCMCA_COMMUNICATION_DIRECTION=<SVCMCA_COMMUNICATION_DIRECTIONid received in on OutGoingEvent()>
inData.SVCMCA_CALL_ID = <SVCMCA_CALL_ID id received in on OutGoingEvent()>
inData.mcaOrigEventSource =<mcaOrigEventSource id received in on OutGoingEvent()>
inData.mcaWindow = <mcaWindow  received in on OutGoingEvent()>
inData.SVCMCA_CONTACT_ID = <SVCMCA_CONTACT_ID received in on OutGoingEvent()>
inData.SVCMCA_DISPLAY_NAME= <SVCMCA_DISPLAY_NAME received in on OutGoingEvent()>
```

```
On the return response, capture outData  inData Object attribute for startCommEvent() and closeCommEvent() .
```

Sample Code For MCA API's

In case if there is an error in call back response of newCommEvent(), log the error in console log.

- toolbarStartCommEvent() function:

It receives parameter like channel,appClassification,channelType,eventId,GENESYS_CALLUUID,timestamp ,eventType ,AgentExtension, NGCC_UserName,customernumber, commDirection,callduration

It invokes MCA API startCommEvent() with below parameters:

- ```
  svcMca.tlb.api.startCommEvent('PHONE', 'ORA_SALES', eventId, inData, function (response) {},'ORA_SVC_PHONE');
  svcMca.tlb.api.startCommEvent(channel, appClassification,eventId, inData, null, function (response) {},channelType);
  construct inData as below:
  ```

  ```
  inData.SVCMCA_ANI =<SVCMCA_ANI  received in on newCommEvent()>  inData.SVCMCA_CONTACT_NAME = <SVCMCA_CONTACT_NAME received in on newCommEvent()>  inData.eventId = <eventId received in on newCommEvent()>  inData.SVCMCA_INTERACTION_ID =<SVCMCA_INTERACTION_ID received in on newCommEvent()>
  inData.SVCMCA_COMMUNICATION_DIRECTION =<SVCMCA_COMMUNICATION_DIRECTION received in on newCommEvent()>
  inData.channel = channel;  inData.channelType = channelType;  inData.channelId = <channelId received in on newCommEvent()>  inData.SVCMCA_CALL_ID =<SVCMCA_CALL_ID received in on newCommEvent()>  inData.SVCMCA_CONTACT_ID =<SVCMCA_CONTACT_ID received in on newCommEvent()>  inData.mcaOrigEventSource =<mcaOrigEventSource received in on newCommEvent()>  inData.mcaWindow = <mcaWindow received in on newCommEvent()>
  ```

Sample Code For MCA API's

In case if there is an error in call back response of startCommEvent(), log the error in console log.

- toolbarCloseCommEvent() function:

It receives parameter like channel,appClassification,channelType,eventId,GENESYS_CALLUUID,timestamp ,eventType , AgentExtension,NGCC_UserName,customernumber, commDirection,callduration,reason

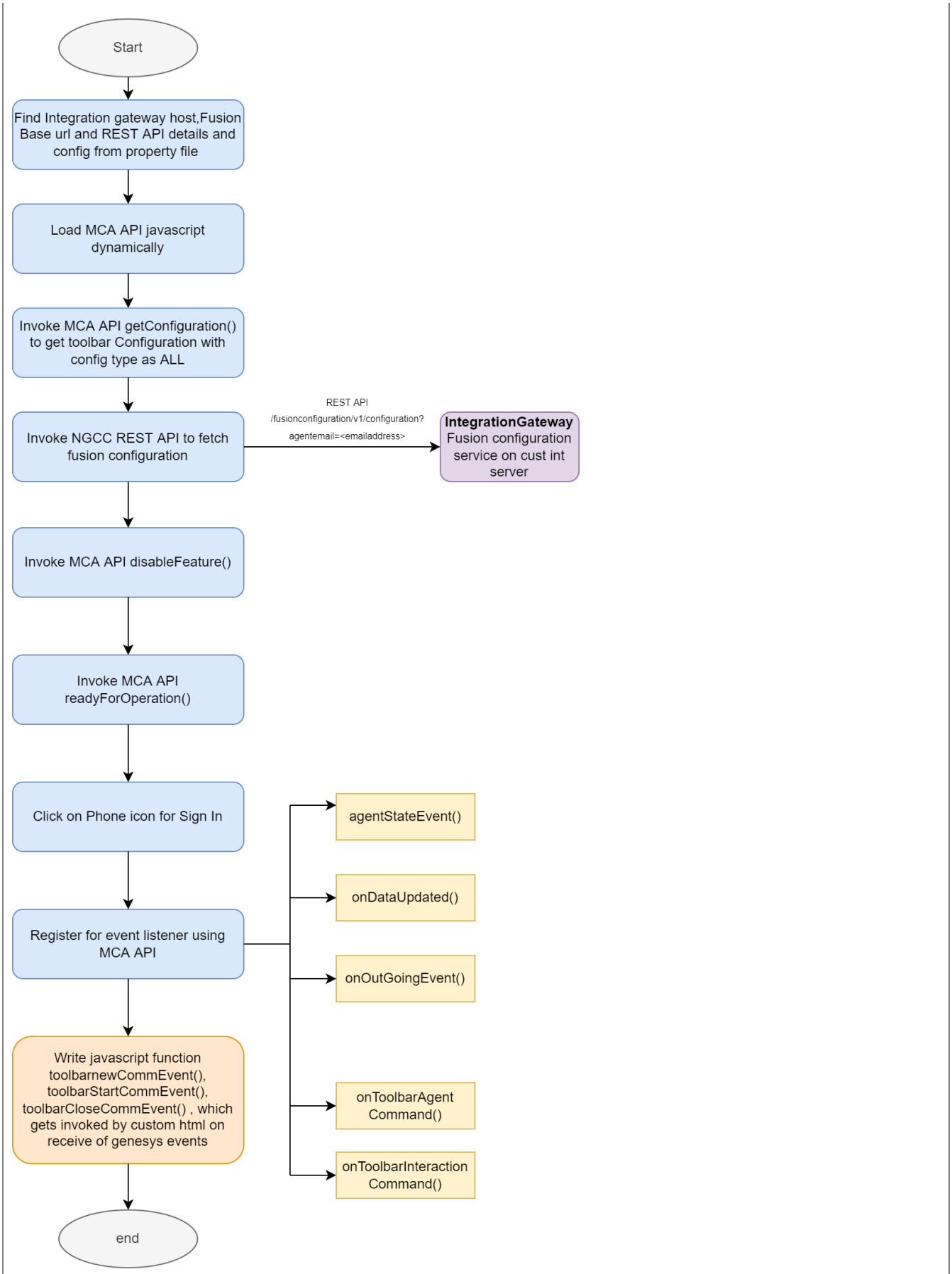
It invokes MCA API closeCommEvent() with below parameters:
  ```

```
svcMca.tlb.api.closeCommEvent('PHONE', 'ORA_SALES', eventId, inData, reason,, function (response){},'ORA_SVC_
PHONE');
svcMca.tlb.api.closeCommEvent('PHONE',  appClassification,eventId, inData, reason,, function (response){},
channelType);

construct inData as below:

 inData.SVCMCA_ANI =<SVCMCA_ANI  received in on startCommEvent()>
 inData.SVCMCA_CONTACT_NAME = <SVCMCA_CONTACT_NAME received in on startCommEvent()>
  inData.eventId = <eventId received in on startCommEvent()>
 inData.SVCMCA_INTERACTION_ID =<SVCMCA_INTERACTION_ID received in on startCommEvent()>
  inData.SVCMCA_COMMUNICATION_DIRECTION =<SVCMCA_COMMUNICATION_DIRECTION received in on startCommEvent()>
  inData.channel = channel;
 inData.channelType = channelType;
 inData.channelId = <channelId received in on startCommEvent()>
 inData.SVCMCA_CALL_ID =<SVCMCA_CALL_ID received in on startCommEvent()>
  inData.SVCMCA_CONTACT_ID =<SVCMCA_CONTACT_ID received in on startCommEvent()>
 inData.mcaOrigEventSource =<mcaOrigEventSource received in on startCommEvent()>
 inData.mcaWindow = <mcaWindow received in on startCommEvent()>
inData.SVCMCA_UI_TYPE_CD =<SVCMCA_UI_TYPE_CD received in on startCommEvent()>
SVCMCA_WRAPUP_ID= <SVCMCA_WRAPUP_IDreceived in on startCommEvent()>
Sample Code For MCA API's
```

High Level flow for Toolbar initialization:


Start
Find Integration gateway host,Fusion Base url and REST API details and config from property file
Load MCA API javascript dynamically
Invoke MCA API getConfiguration() to get toolbar Configuration with config type as ALL
Invoke NGCC REST API to fetch fusion configuration
REST API
/fusionconfiguration/v1/configuration?
agentemail=<emailaddress>
IntegrationGateway
Fusion configuration service on cust int server
Invoke MCA API disableFeature()
Invoke MCA API readyForOperation()
Click on Phone icon for Sign In
Register for event listener using MCA API
agentStateEvent()
onDataUpdated()
onOutGoingEvent()
onToolbarAgent Command()
onToolbarInteraction Command()
Write javascript function toolbarnewCommEvent(), toolbarStartCommEvent(), toolbarCloseCommEvent() , which gets invoked by custom html on receive of genesys events
end

Custom HTML for Genesys Event Handling

Custom HTML for Genesys Event Handling:

NGCCModelXFusionCustom.html is loaded in NGCC WWE as hidden page.

This WWE custom HTML receives Genesys call events with WWE service client API's provided by Genesys.

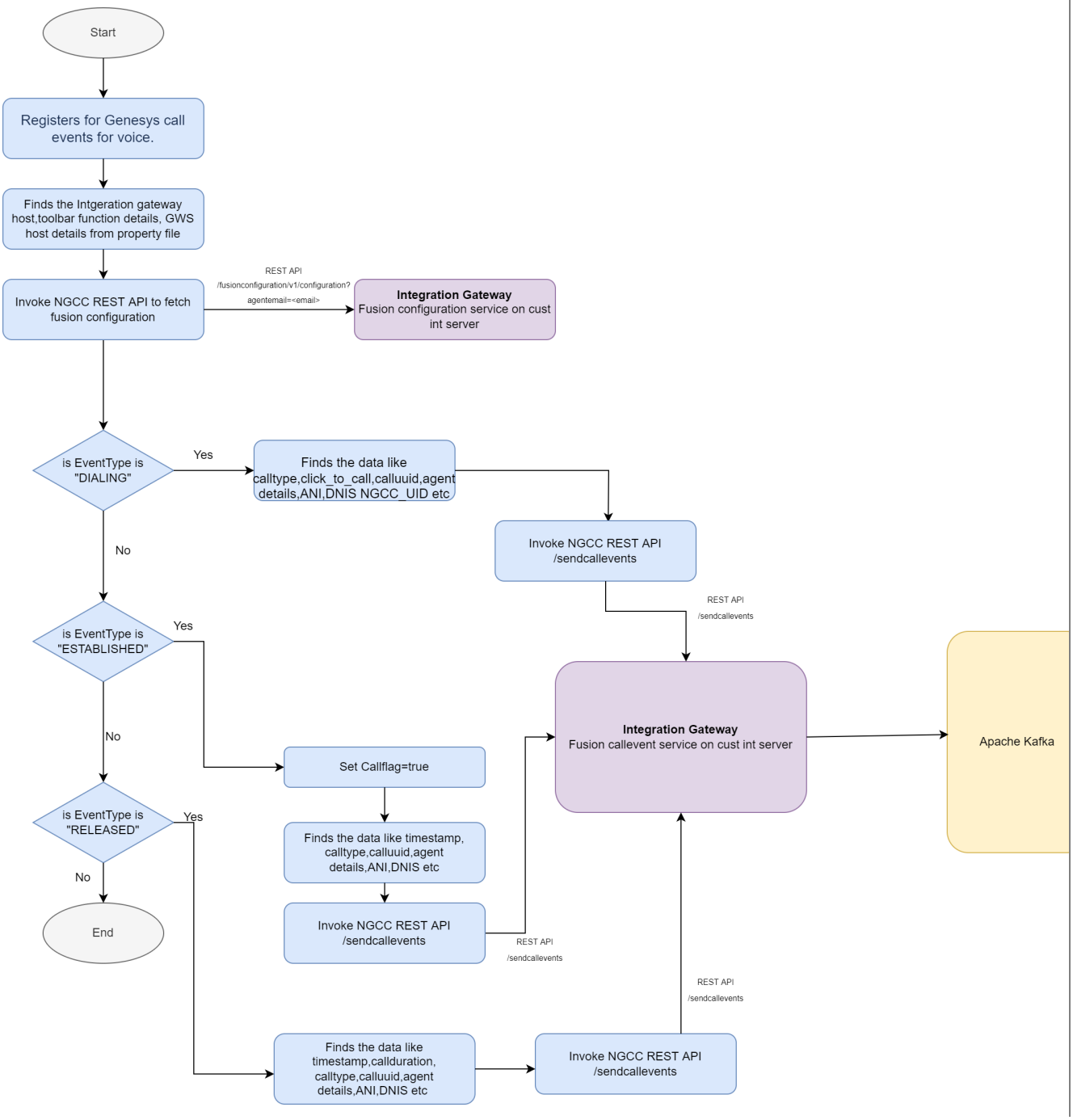
Handles events like "ESTABLISHED","RELEASED","DIALING" and invokes the NGCC REST API /sendcallevents to post call event details to Apache Kafka so that Toolbar fetch call events details using NGCC REST API /receivecallevents and Toolbar can communicates the events to Fusion.

- NGCCModelXFusionCustom.html loads in to WWE as hidden page when agent log in to WWE.
- NGCCModelXFusionCustom html registers for Genesys call events for voice.
- Finds the Integration gateway and Genesys Hostname mapping ,toolbar java script which needs to be invoked as per the WWE node side so that custom html can invoke appropriate Host name,java script for events.(property file)
- Invoke NGCC REST API ngcc_webservice/fusionconfiguration/v1/configuration?agentemail=<salesRep@oracle.com> to fetch Fusion configuration.
- On Receive of "**DIALING**" event(outbound call), it does following
    - Find NGCC Timestamp using new Date().getTime(), customernumber as message.data.interaction.dnis, GENESYS_CALLUUID as message.data.interaction.callUuid.
    - find click_to_call is true or false using message.data.interaction.userData.click_to_call;
    - if click to call is true find NGCC_Agentextension by message.data.interaction.userData. agentExtension, NGCC_UserName as message.data.interaction.userData.agentEmailId,eventId as message.data. interaction.userData.NGCC_UID
    - Construct call event details object with event type as DIALING, channel,appclassifictaion,channelnameas fetched in configuration fetch from NGCC REST API,,customer number, timestamp,comm direction, genesys calluuid, reason etc.
    - Invoke NGCC REST API /sendcallevents (post method) and passes callevents objects so that NGCC REST API /sendcallevents writes call event details to apache kafka.
- On Receive of **"ESTABLISHED"** event, it does the following
    - find NGCC_calltype using message.data.interaction.callType;
    - Set callflag=true
    - Find NGCC Timestamp using new Date().getTime(), NGCC_CALLUUID and eventId message.data.interaction.callUuid;
    - Find customernumber using message.data.interaction.userData.dnis,
    - find click_to_call is true or false using message.data.interaction.userData.click_to_call;
    - if click to call is true find NGCC_Agentextension by message.data.interaction.userData. agentExtension, NGCC_UserName as message.data.interaction.userData.agentEmailId and assign eventid as message.data.interaction.userData.NGCC_UID
    - if click to call is true and If NGCC_calltype is OUTBOUND,
    - Construct call event details object with event type as ESTABLISHED, channel,appclassifictaion,channelnameas fetched in configuration fetch from NGCC REST API,,customer number, timestamp,comm direction, genesys calluuid, reason etc.
    - Invoke NGCC REST API /sendcallevents (post method) and passes callevents objects so that NGCC REST API /sendcallevents writes call event details to apache Kafka.
- On receive of "**RELEASED**" event, it does the following
    - find callflag is true or false
    - find NGCC_calltype using message.data.interaction.callType;
    - Find NGCC Timestamp using new Date().getTime(), GENESYS_CALLUUID and eventidas message.data.interaction. callUuid;
    - If callflag is true
        - Find the callduration in milliseconds by finding time difference in current time and time when ESTABLISHED event is received
        - Find ANI using message.data.interaction.userData.ANI; and destination number using message.data. interaction.userData.dnis
        - Find NGCC_interaction as message.data.interaction.userData.interaction_id
        - Find the Agent Extension, agent DN and Agent User name as per the keys specified in WWE PARTY.
        - if calltype is OUTBOUND or INTERNAL, find whether click_to_call is true using message.data.interaction. userData.click_to_call;
            - if click to call is true , find NGCC_Agentextension=message.data.interaction.userData.agentExtension, NGCC_UserName=message.data.interaction.userData.agentEmailId,eventid as message.data. interaction.userData.NGCC_UID
            - Construct call event details object with event type as RELEASED, channel,appclassifictaion, channelnameas fetched in configuration fetch from NGCC REST API,,customer number, timestamp, comm direction, genesys calluuid, reason as WRAPUP etc.
            - Invoke NGCC REST API /sendcallevents (post method) and passes callevents objects so that NGCC REST API /sendcallevents writes call event details to apache Kafka.

- If callflag is false,
    - Find ANI using message.data.interaction.userData.ANI; and destination number using message.data.interaction.userData.dnis
    - Find the Agent Extension, agent DN and Agent User name as per the keys specified in WWE PARTY.
    - if calltype is OUTBOUND or INTERNAL, find whether click_to_call is true using message.data.interaction.userData.click_to_call;
        - if click to call is true , find NGCC_Agentextension=message.data.interaction.userData.agentExtension, NGCC_UserName=message.data.interaction.userData.agentEmailId,eventid as message.data.interaction.userData.NGCC_UID

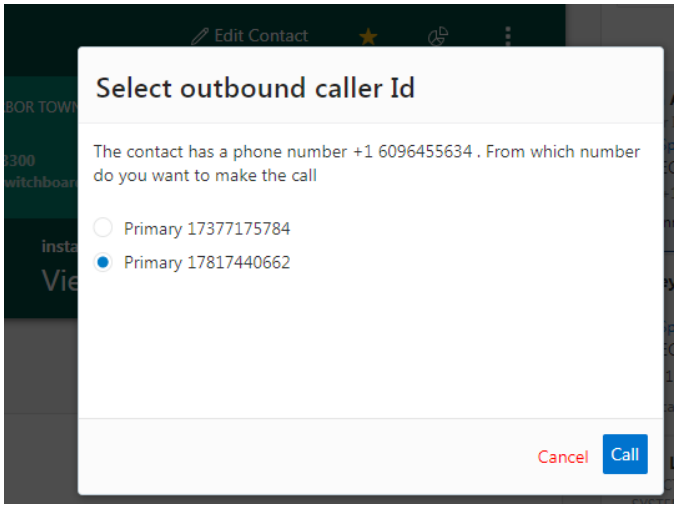
        - Construct call event details object with event type as RELEASED, channel,appclassifictaion,channelnameas fetched in configuration fetch from NGCC REST API,customer number, timestamp,comm direction, genesys calluuid, reason as MISSED ,topic name etc.
        - if click to call is true,Invoke NGCC REST API /sendcallevents (post method) and passes callevents objects so that NGCC REST API /sendcallevents writes call event details to apache Kafka.

NGCCModelXFusionCustom.html high level flow:



Caller ID and Click To Dial &Call event stamping

- Agent login to his/her Cisco Jabber account which is configured.
- Agent login to NGCC workspace WWE using SSO account and makes channel voice as Ready.
- Agent login to Digital sales and Clicks on Contacts.
- Clicks on Manage Presence icon on top left and it loads toolbar. Agent login to toolbar by clicking phone icon. Registers for MCA events.
- Agent Selects any contacts and clicks on Phone number.
- On click of phone number,Since Toolbar is already registered for onOutgoingEvent(),Fusion applications sends onOutgoingEvent() to toolbar using MCA Framework. Toolbar receives customer ANI, Customer Id, Contact name, Contact email with this event. ie. `SVCMCA_ANI`, `SVCMCA_EMAIL`, `SVCMCA_CONTACT_ID`, and `SVCMCA_CONTACT_NAME`. Note:`SVCMCA_ANI` is the customer phone number which is to be used in clicktodial pay load later.
    - In onOutgoingEvent() method of toolbar do below thing:
    - Invoke Caller Id REST API NGCC service to fetch callerid of an agent using agent email address which is received with getConfiguration() response. Refer API documentation .ClickToDial & CallerId Webservice API
    - On return response of fetch caller Id, show up all caller id's in selection window as below. In case agent is not configured in NGCC, it shows up error and dial fails. Incase callerId fetch fail, set outboundCallerId as 'n/a' for click to dial.
    - Incase caller id fetch fails, say that Primary phone number of contact is used to make outbound call and on click to call button, it invokes click to dial REST API.

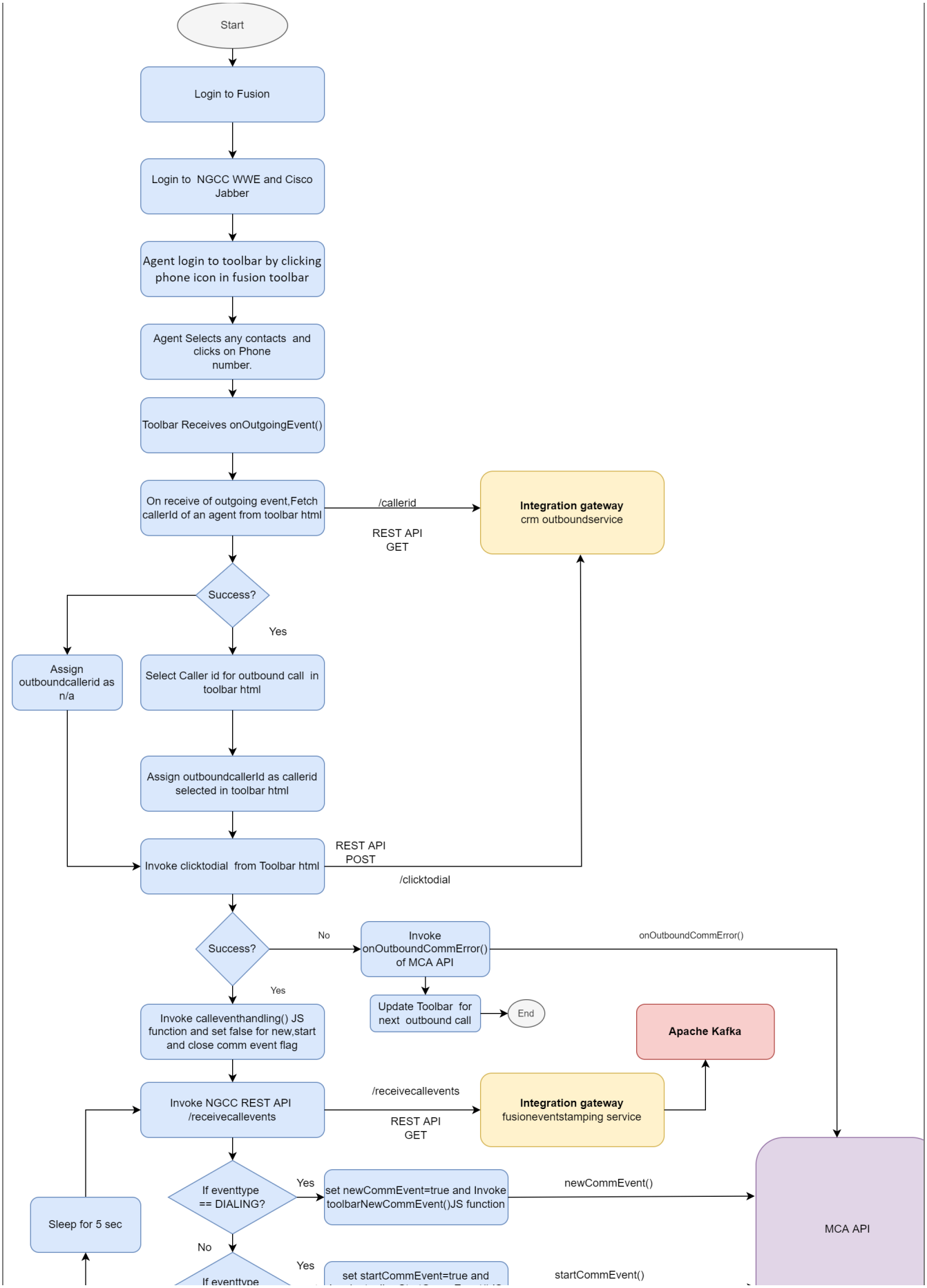


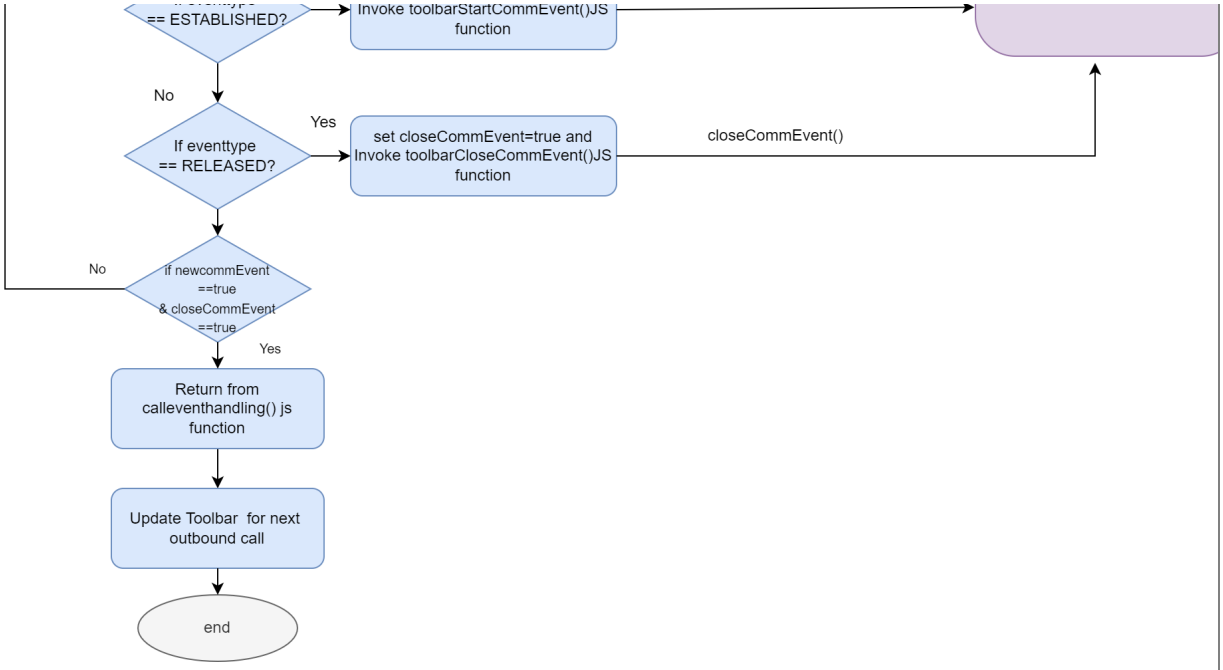
- If caller id fetch is success, Agent selects one of the caller id, On selection of caller id, clicks on dial button in that window and it invokes click to dial REST API of NGCC using ajax call. On UI ,show that call is in progress as below with Customer Name, ANI and Dialing...

- If Click to dial REST API is failed, show error response in Toolbar UI with error window. Invoke MCA API, outboundCommError() and update toolbar UI like green phone icon with agent name.Note: commUuid is nothing but NGCC UID returned in click to dial.
- If Click to dial REST API is success, it receives NGCC_UID from REST API response need to start capturing call events from Genesys. Invoke Javascript function calleventshandling().
- calleventhandling() javascript function ,Sets false for flag newCommEvent,startCommEvent,closeCommEvent
- Invokes NGCC REST API /receivecallevents using agent email address and NGCC_UID and topic name for every 5 sec till it gets calleventobject with DIALING,ESTABLISHED and RELEASED. **Note:** It is not necessary that it will get all 3 events for each outbound call. Sometimes if call is not Established between agent and customer, only DAILING and RELEASED events will be received.
- Response of /receivecallevents contains list of callevent objects. Checks eventtype in call event objects.
- If event type is DIALING, captures calluuid in response and pass this as event id and SVCMCA_COMMUNICATION_DIRECTION as ORA_SVC _OUTBOUND and pass this information to Toolbar by invoking toolbarNewCommEvent() js function so that toolbar invokes newCommEvent() of MCA framework. Set newCommEvent=true

- If eventtype is "ESTABLISHED" event it invokes toolbar javascript function toolbarStartCommEvent() with required parameter so that toolbar invokes startCommEvent() of MCA framework . In this case first leg of the call connects to agent and second leg of the call connects to customer.Set startCommEvent=true
- If eventtype is "RELEASED" event and it invokes toolbar javascript function toolbarCloseCommEvent() with required parameter so that toolbar invokes closeCommEvent() of MCA framework.Set closeCommEvent=true
- If value is true for all flag newCommEvent,startCommEvent,closeCommEvent,returns calleventhandling() java script function.
- Update toolbar UI like green phone icon with agent name

High Level Flow


Start
Login to Fusion
Login to NGCC WWE and Cisco Jabber
Agent login to toolbar by clicking phone icon in fusion toolbar
Agent Selects any contacts and clicks on Phone number.
Toolbar Receives onOutgoingEvent()
On receive of outgoing event,Fetch callerId of an agent from toolbar html
/callerid
REST API
GET
Integration gateway
crm outboundservice
Success?
Yes
Assign outboundcallerid as n/a
Select Caller id for outbound call in toolbar html
Assign outboundcallerId as callerid selected in toolbar html
Invoke clicktodial from Toolbar html
REST API
POST
/clicktodial
Success?
No
Invoke onOutboundCommError() of MCA API
onOutboundCommError()
Yes
Update Toolbar for next outbound call
End
Invoke calleventhandling() JS function and set false for new,start and close comm event flag
Apache Kafka
Invoke NGCC REST API /receivecallevents
/receivecallevents
REST API
GET
Integration gateway
fusioneventstamping service
If eventtype == DIALING?
Yes
set newCommEvent=true and Invoke toolbarNewCommEvent()JS function
newCommEvent()
MCA API
Sleep for 5 sec
No
Yes
set startCommEvent=true and
startCommEvent()



Fusion Event Stamping Web Service

Fusion Event Stamping webservice is used for call event stamping to Fusion. Apache kafka is used as Message broker and Genesys events are communicated to Fusion via Apache Kafka.

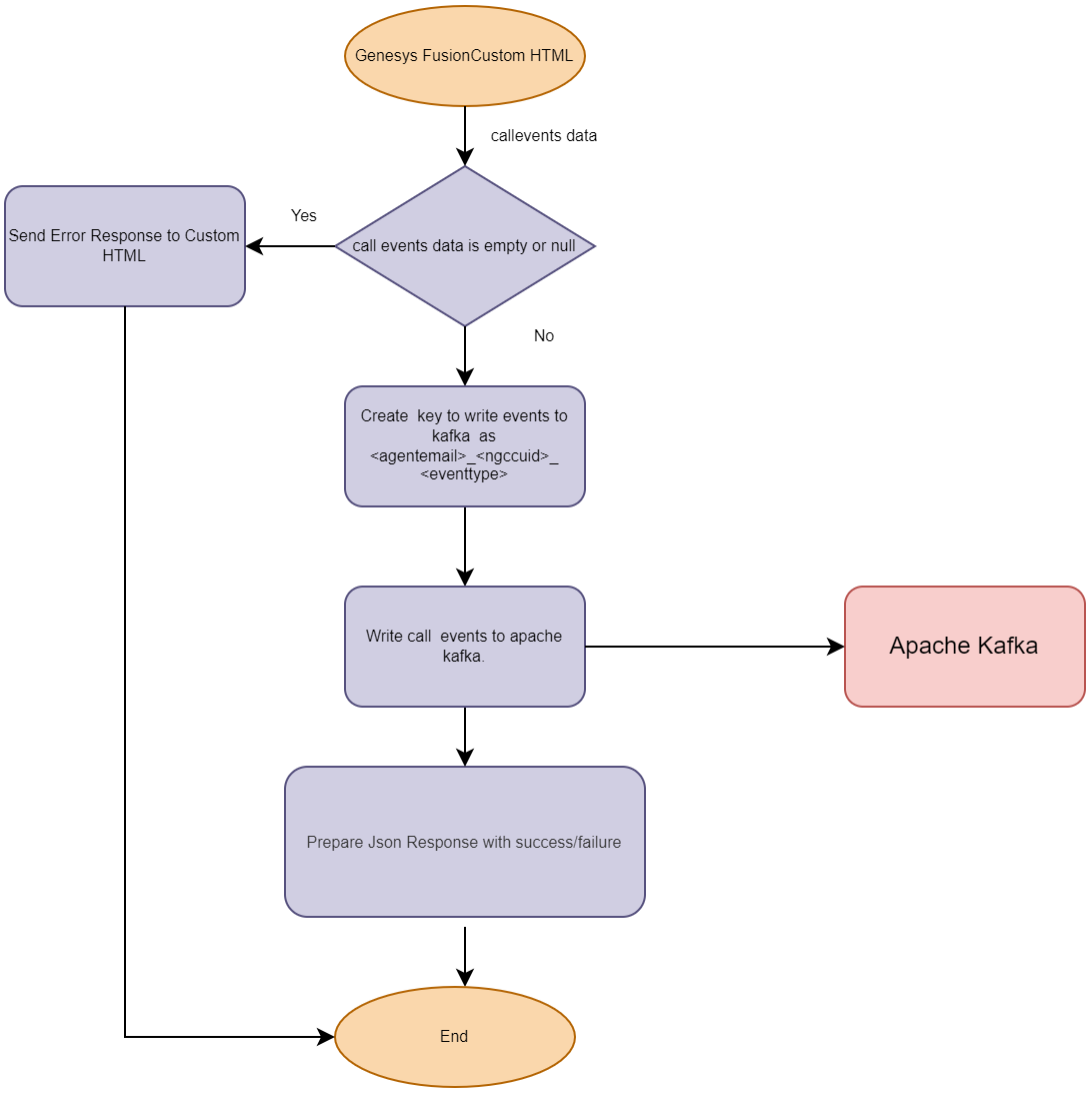
Genesys Fusion Custom HTML(NGCCModelxFusionWWEEventHandler.js) receives Genesys call events like DIALING, ESTABLISHED, RELEASED and it invokes event stamping webservice(/sendcallevents) API. /sendcallevents REST API accepts the call events request from Geneys Fusion Custom Html and writes call events to Kafka.

Toolbar(NGCCFusionToolbar.html) which is loaded as iFrame in Fusion, invokes /receivecallevents REST API on success of click to dial.

/receivecallevents API reads Genesys call event details from Apache Kafka and returns call event details to Toolbar. Fusion Toolbar (NGCCFusionMCAAPI.js) invokes MCA API to do call event stamping in Fusion.

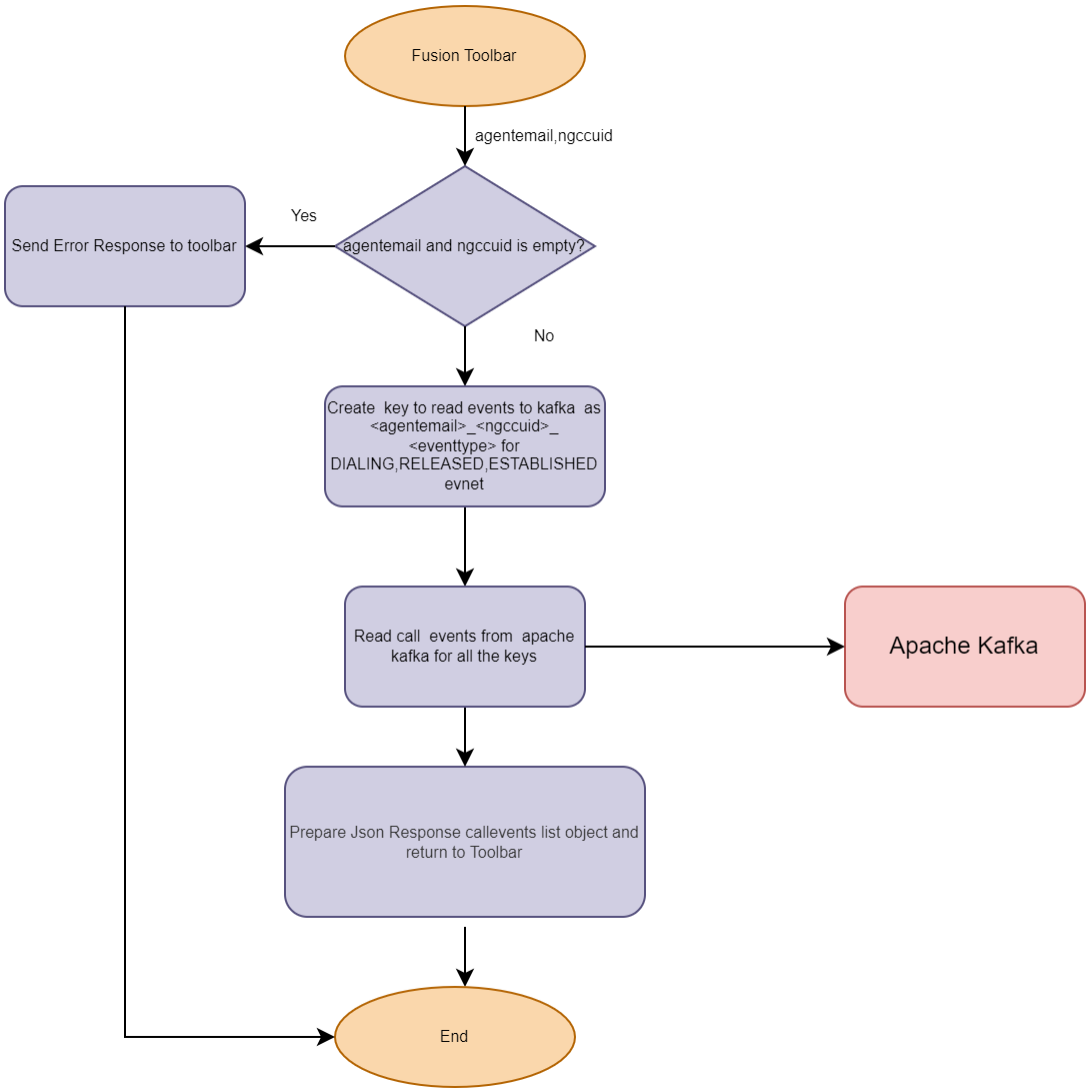
/sendcallevents flow:

- Integration gateway receives POST REST API /sendcallevents request from Genesys Fusion Custom HTML wth request body as Callevent details like eventtype,calluuid,ngccuid,callduartion,channeltype,appclassification,agentemail, etc.
- Integration gateway web service validates whether mandatory objects in callevent objects are empty or null. Mandatory attributes: agent email, calluuid, ngccuid, eventtype,topicname
- If mandatory attributes are empty or null,sends error response to Genesys Fusion Custom HTML.
- If data is not empty/null , creates key to write call event details to Apache Kafka using <agentemail>_<ngccuid>_<eventtype>
- Writes callevents to apache kafka on topic which is received in request body("OD_SALES_NAM_OUTBOUND") with key <agentemail>_<ngccuid>_<eventtype>.
- Sends Success/failure response to Genesys Fusion Custom Html as per the success or failure of call event write to Kafka.



/receivecallevents flow:

- Integration gateway receives GET REST API /receivecallevents request from Toolbar with request parameter as agentemail and ngccuid&topic eventtype,calluuid,ngccuid,callduartion,channeltype,appclassification,agentemail, etc.
- Integration gateway web service validates whether agentemail and ngccuid is empty or null.
- If input request parameters are empty or null, sends error response to Toolbar.
- If input data is not empty or null, creates key to read call event details from Apache Kafka using <agentemail>_<ngccuid>
- Reads callevents from apache kafka topic name received in request parameter("OD_SALES_NAM_OUTBOUND") with key <agentemail>_<ngccuid>
- Prepare Json Response with list of callevents object received from Kafka.
- Sends Success/failure response to Toolbar along with callvent details as per the success or failure of call event read from Kafka.



Apache Kafka details:

Topic name: OD_SALES_NAM_OUTBOUND for OD NAM outbound agents

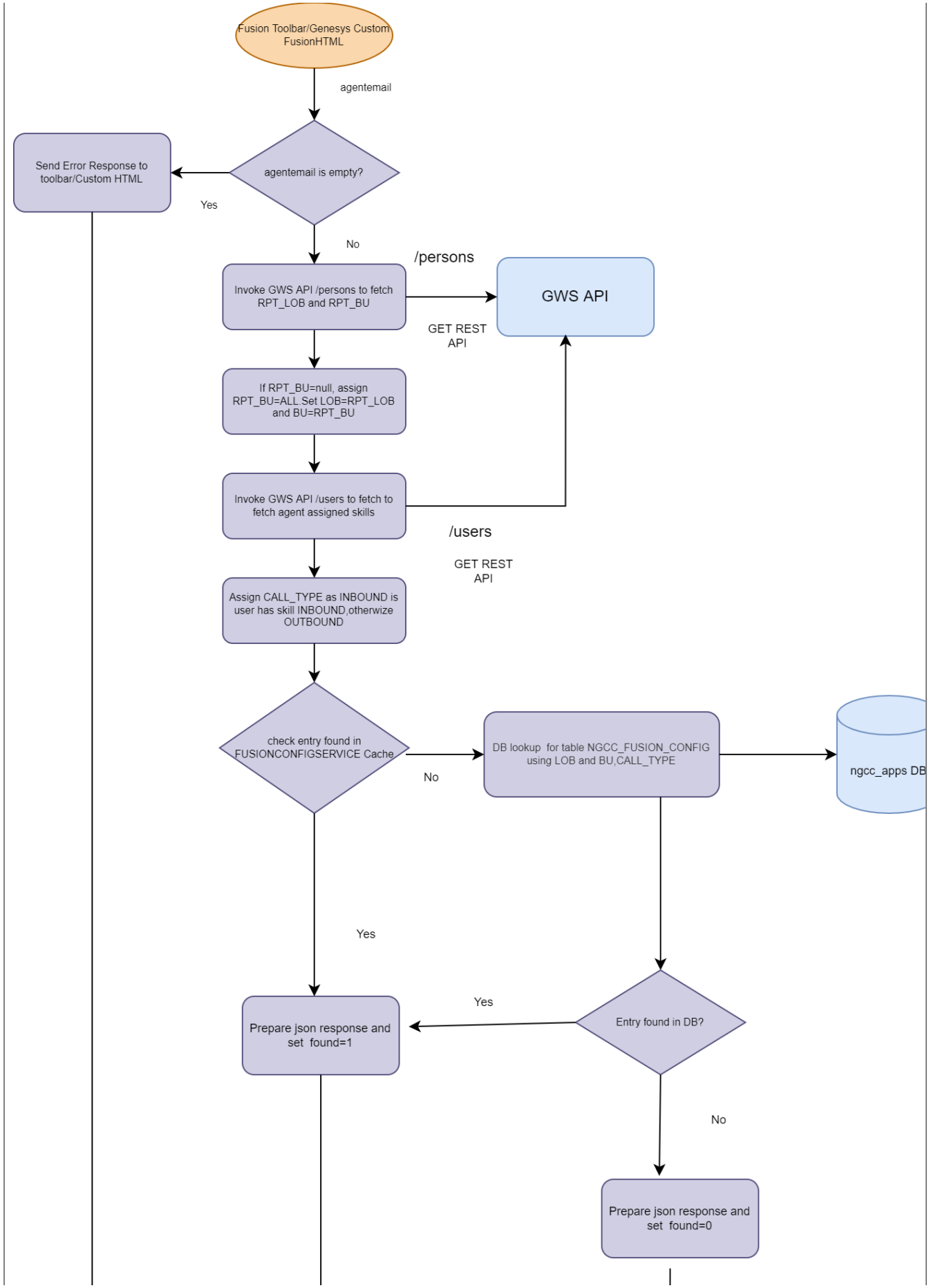
Key: <agentemail>_<ngccuid>_<eventtype>

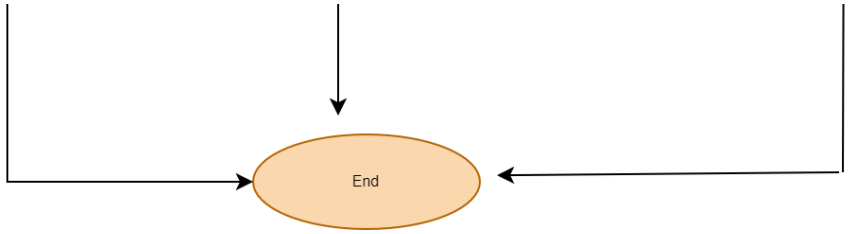
Fusion Configuration Webservice

Fusion Configuration webservice is used to fetch fusion configuration like disable feature list, LOB,BU,channelType,channlename, appclassification Id,kafka topic name, call type, maxwaitingtimeDialing event etc using agent email address.

NGCC Fusion Configuration webservice receives agent email address as request parameter.

- Integration gateway receives GET REST API /configuration request from Toolbar and also from Genesys Fusion Custom HTML with request parameter as agentemail.
- Integration gateway web service validates whether agentemail is empty or null.
- If agent email is empty or null, sends error response to Toolbar/Genesys Fusion Custom Html
- If agentemail is not empty or null, invokes GWS API(/persons) API to fetch RPT_LOB and RPT_BU of an agent. Also checks whether agent has INBOUND skill. Use GWS API like **Sample GWS API: https://sit-wwe.ngcc-v1-ash1-oragit.oraclecorp.com/api/v2/users?userName=nagaratna.hegde@oracle.com&subresources=skills** to check the same.
- If RPT_BU is empty or null, set RPT_BU=ALL. Assign LOB=RPT_LOB value and BU as RPT_BU value. If Agent has INBOUND skill, set CALL_TYPE=INBOUND, else CALL_TYPE as OUTBOUND.
- Does Database lookup(NGCC_FUSION_CONFIG) to fetch Fusion configuration using LOB and BU & CALL_TYPE.
- If configuration data found in database, set found=1 and prepare json response with Disable feature list, channel name,channeltype, appClassification,Kafka topic name etc.
- If Configuration data not found in database, set found=0 and prepare Json response with error details.
- Return Json Response to Toolbar/Genesys Fusion Custom WWE HTML.


Fusion Toolbar/Genesys Custom FusionHTML
agentemail
agentemail is empty?
Send Error Response to toolbar/Custom HTML
Yes
No
/persons
Invoke GWS API /persons to fetch RPT_LOB and RPT_BU
GWS API
GET REST API
If RPT_BU=null, assign RPT_BU=ALL.Set LOB=RPT_LOB and BU=RPT_BU
Invoke GWS API /users to fetch to fetch agent assigned skills
/users
GET REST API
Assign CALL_TYPE as INBOUND is user has skill INBOUND,otherwize OUTBOUND
check entry found in FUSIONCONFIGSERVICE Cache
No
DB lookup for table NGCC_FUSION_CONFIG using LOB and BU,CALL_TYPE
ngcc_apps DB
Yes
Yes
Entry found in DB?
Prepare json response and set found=1
No
Prepare json response and set found=0



NGCC_FUSION_CONFIG TABLE

Column Name	Data Type	Constraints
RECORD_ID	NUMBER	PK,SEQUENCE
LOB	VARCHAR2 (30)	NOT NULL,PK
BU	VARCHAR2 (30)	NOT NULL,PK
DISABLE_FEATURE_LIST	VARCHAR2 (1000)	
CHANNEL_TYPE	VARCHAR2 (30)	NOT NULL
CHANNEL	VARCHAR2 (30)	NOT NULL
APP_CLASSIFICATION_ID	VARCHAR2 (30)	NOT NULL
KAFKA_TOPIC	VARCHAR2 (30)	NOT NULL
EVENT_POLLING_INTERVAL	NUMBER	NOT NULL
CALL_TYPE	**VARCHAR2 (30)**	**PK**
MAX_WAITING_TIME_DIALING	**NUMBER**	**NOT NULL**
CREATE_DATE	TIMESTAMP	
LAST_UPDATE_DATE	TIMESTAMP	
LAST_UPDATE_BY	VARCHAR2 (100)	NOT NULL

Genesys Configuration

To be added only for agents who uses Fusion .

WWE Cluster Configuration

Section=App-**NGCCModelXFusionCustom**

url=https://dev-cust-int.ngcc-v1-phx1-oragit.oraclecorp.com/ngcc_webservice/fusion/v1/**NGCCModelXFusionCustom.html**?userName=$Agent.UserName$

mode=HIDDEN

label=fusion

ZZZ Agent Group configuration for Modelx

[interaction-workspace]

workspace.web-content=App-**NGCCModelXFusionCustom**

Skills for Fusion OD NAM Outbound: OD_SALES,NAM,NEXT_GEN_SALES

Fusion and NGCC Host Mapping

ENV	Fusion URL	NGCC Toolbar for Fusion	NGCC WWE URL	User Details	Cisco Jabber
DEV	**https://eeho-dev4.fa.us2.oraclecloud.com/crmUI/faces/FuseWelcome**	https://dev-cust-int.ngcc-v1-phx1-oragit.oraclecorp.com/ngcc_webservice/fusion/v1/NGCCFusionToolbar.html	https://dev-wwe.ngcc-v1-phx1-oragit.oraclecorp.com/ui/ad/v1/index.html?authType=saml	fusion-oal-sales-rep-110_ww@oracle.com Rel8_sr110	13033346061
SIT	https://efip-dev3.fa.us6.oraclecloud.com**/crmUI/faces/FuseWelcome**	https://ngcc-sit-integrations.oraclecorp.com/ngcc_webservice/fusion/v1/NGCCFusionToolbar.html	https://cc-sit-wwe.oraclecorp.com/ui/ad/v1/index.html?authType=saml	fusion-oal-sales-rep-110_ww@oracle.com Rel8_sr110	13033346061
PROD	https://eeho.fa.us2.oraclecloud.com**/crmUI/faces/FuseWelcome**	https://ngcc-integrations.oraclecorp.com/ngcc_webservice/fusion/v1/NGCCFusionToolbar.html	https://ngcc-wwe.oraclecorp.com/ui/ad/v1/index.html?authType=saml		

Additional Notes

- **B2B REST API Services:**

  B2B service : https://docs.oracle.com/en/cloud/saas/sales/22a/faaps/Use_JWT_Token_for_Authorization.html

  Using JWT received in getConfiguration() with config type as FA_TOKEN, need to make REST calls to Fusion to get the agent's configured email address, use the agentId field received from getConfiguration as the Username, then query the resourceUser query for the email address for this user. E.g.:

  Sample B2B REST API

  {{base_url}}/crmRestApi/resources/11.13.18.05/resourceUsers?q=Username="mike.rabatin"&fields=ResourceEmail

  This returns the agent email address :

  "items": [

  {

  "ResourceEmail": mike.rabatin@oracle.com

  }