



# “The” Design Challenge

Your Cisco Team

2024



Oops! Let me try to restore it!



# What are the 5 steps?



Now the real challenges 😊

## Region Originating Call Legs



**Sioe-Bing Khouw (EMEA)**

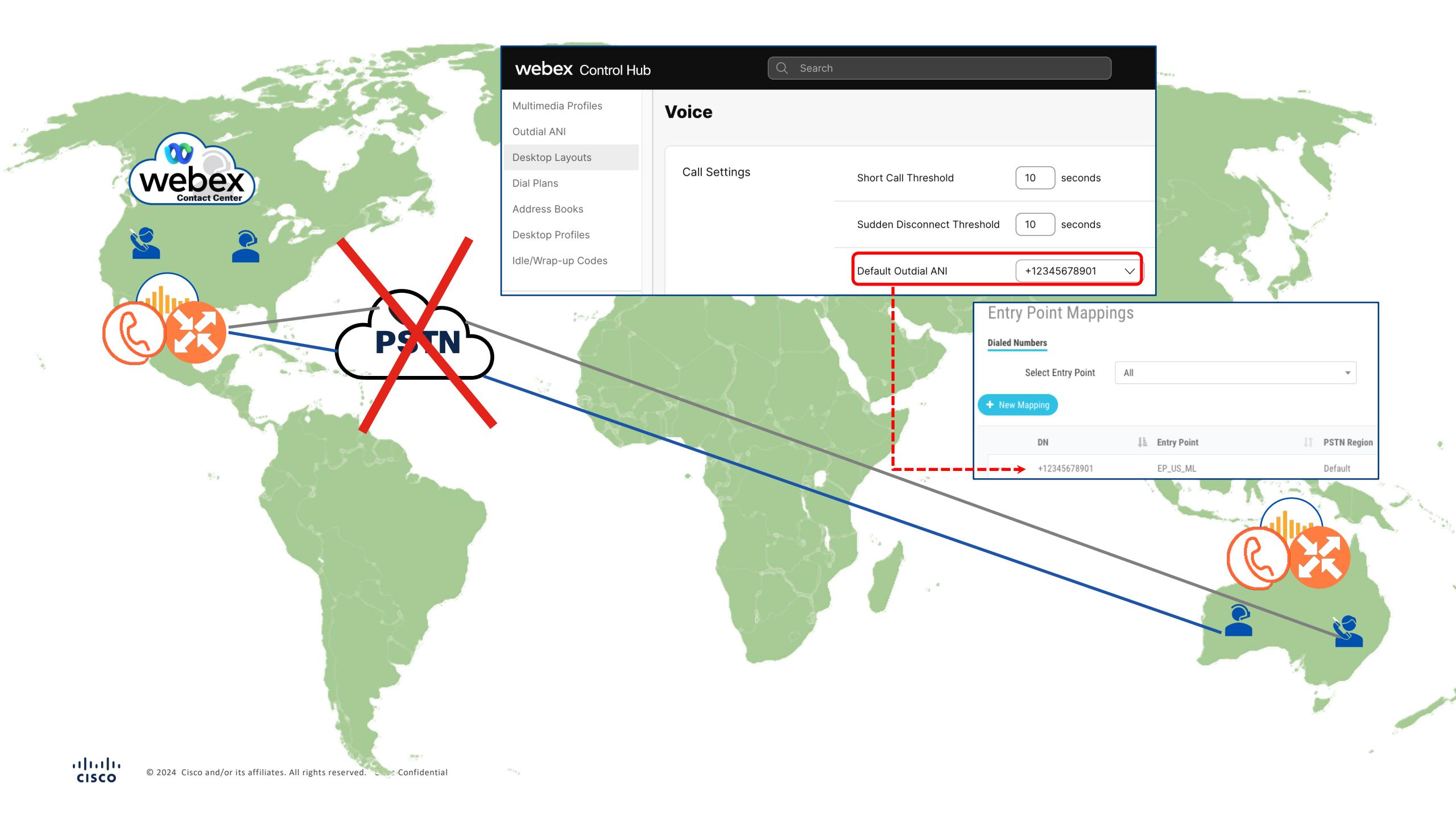
# PSTN Region Originating Call Legs

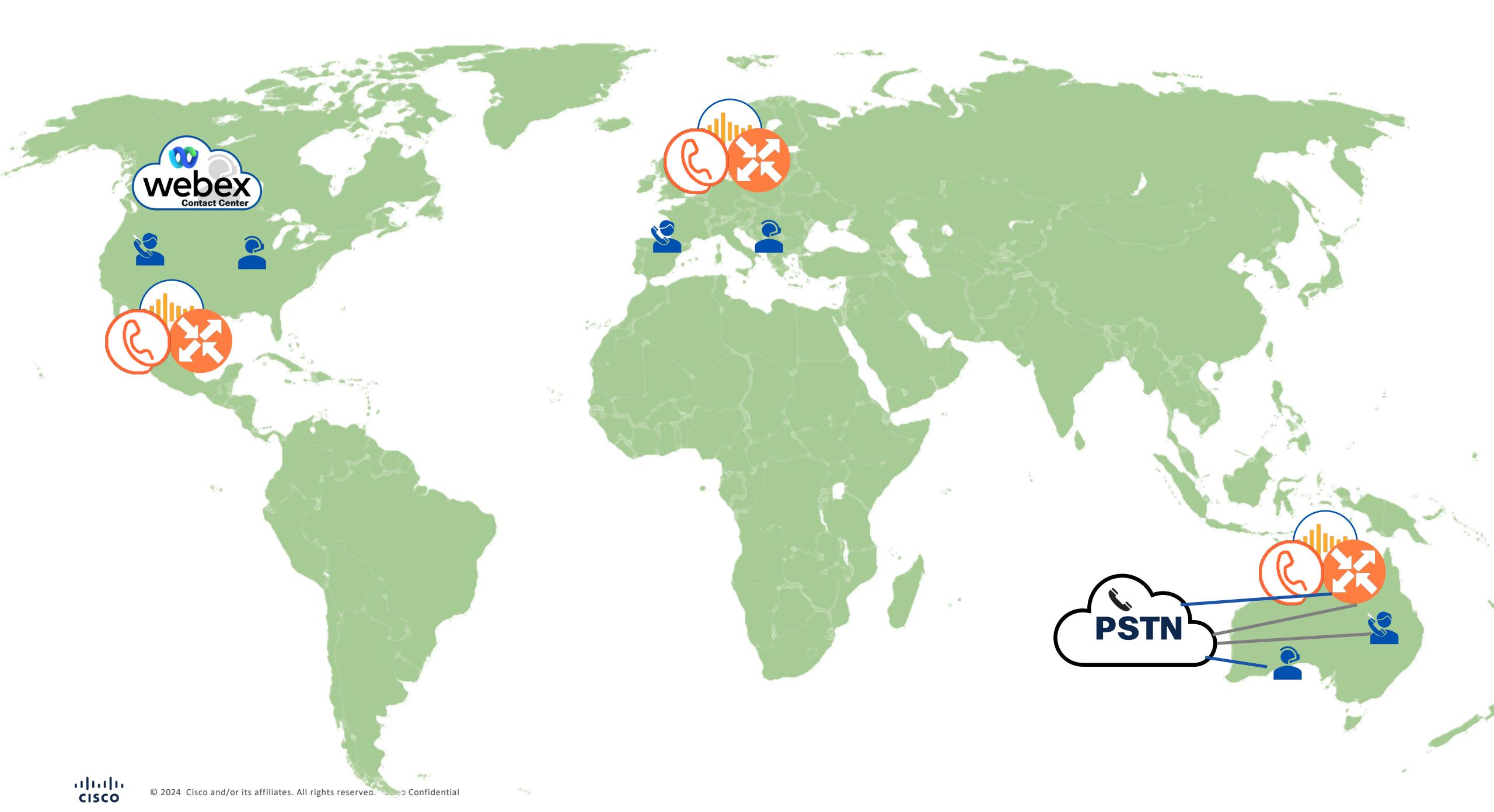
Customer ACME based out of the US > App Center in US.

Agents based in US, Europe and Australia.

Australian Agents making Domestic Outdial calls from Australia:

- Both Called Party Leg and Agent Leg should Originate from Australia
- Different ANI's should be presented on the Called Party and Agent Device
- In case the Agent selects an ANI from the Drop Down (Outdial ANI), this ANI should be presented on the Called Party Device.

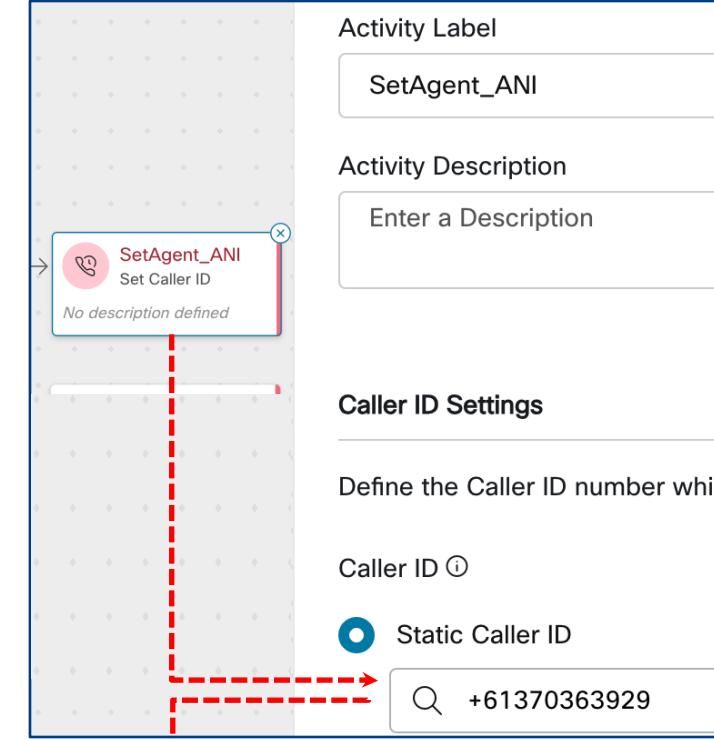
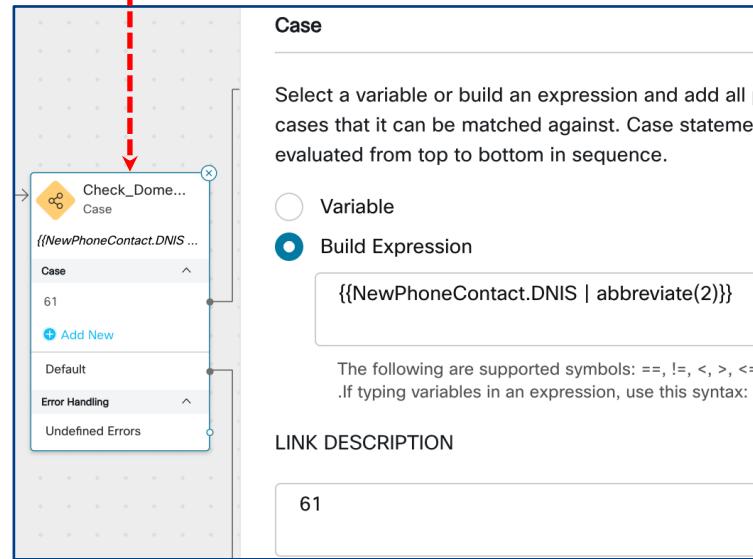
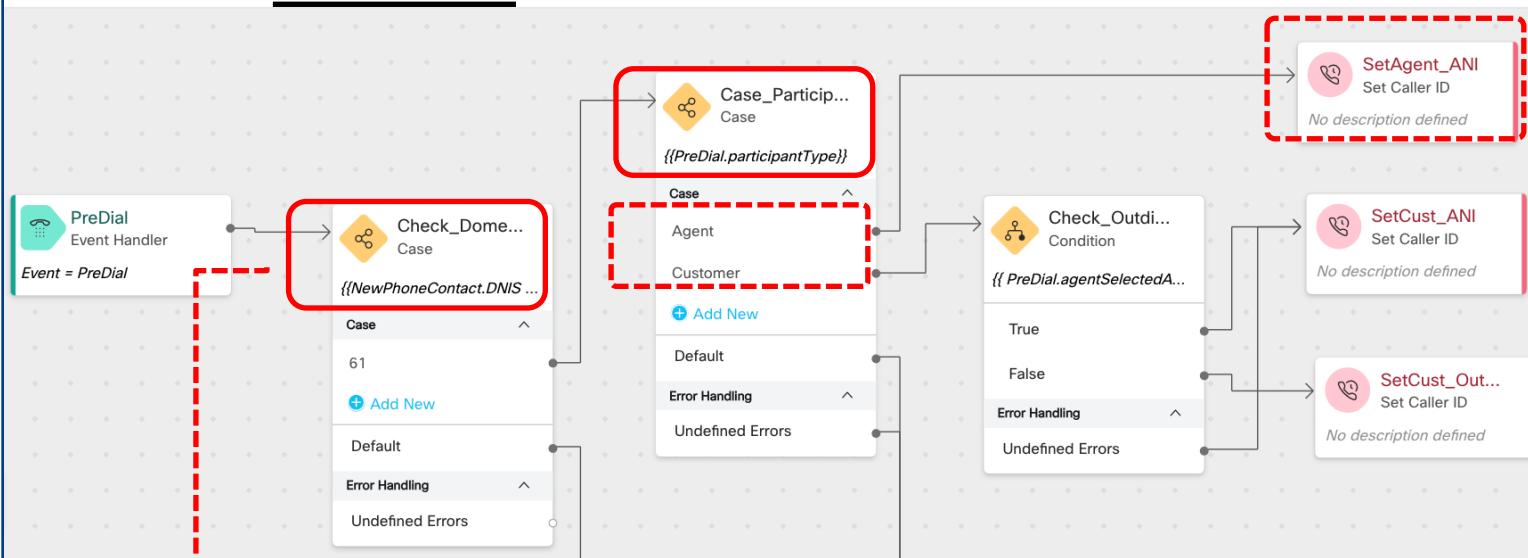




Main Flow

Event Flows

Help



## Entry Point Mappings

### Dialed Numbers

Select Entry Point

All

+ New Mapping

DN	Entry Point
... +61370363929	EP_AU_ML

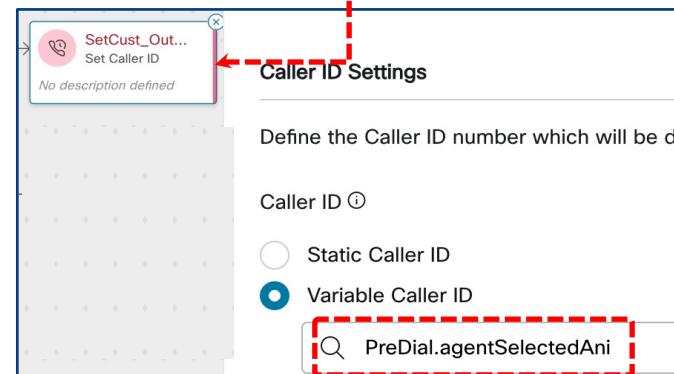
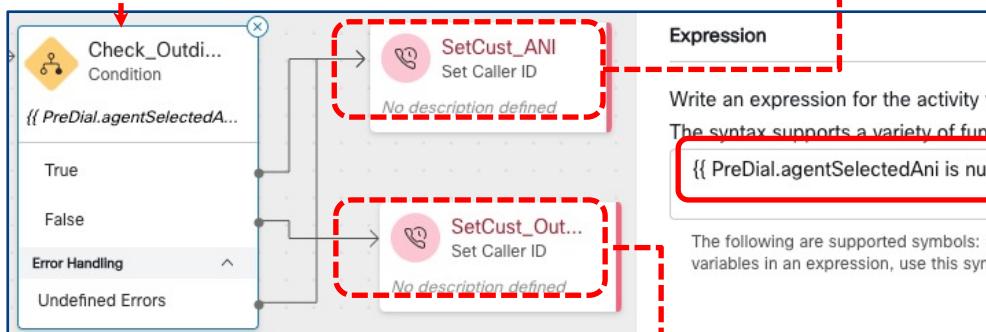
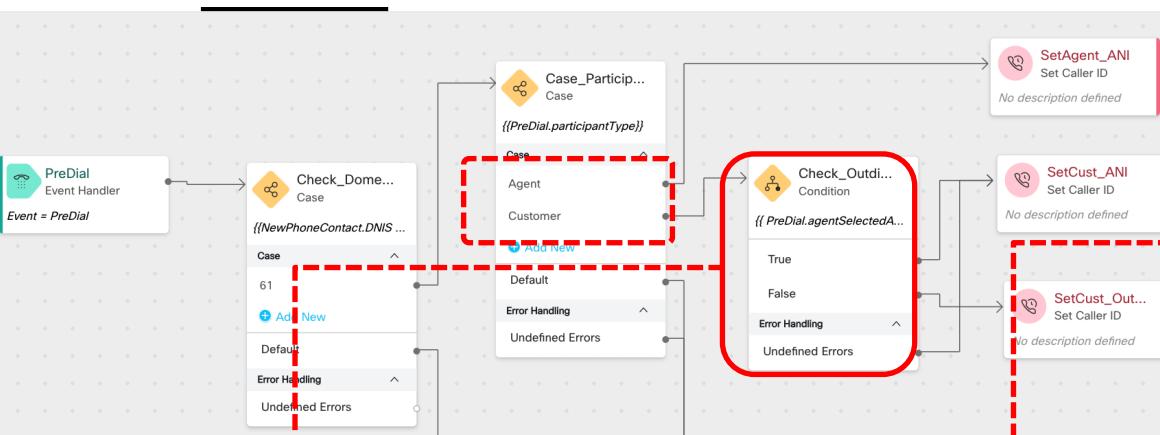
PSTN Region  
Australia

Decides upon Region

Main Flow

Event Flows

Help



Outdial ANI

### Caller ID Settings

Define the Caller ID number which will be displayed.

Caller ID (Static Caller ID)

Static Caller ID

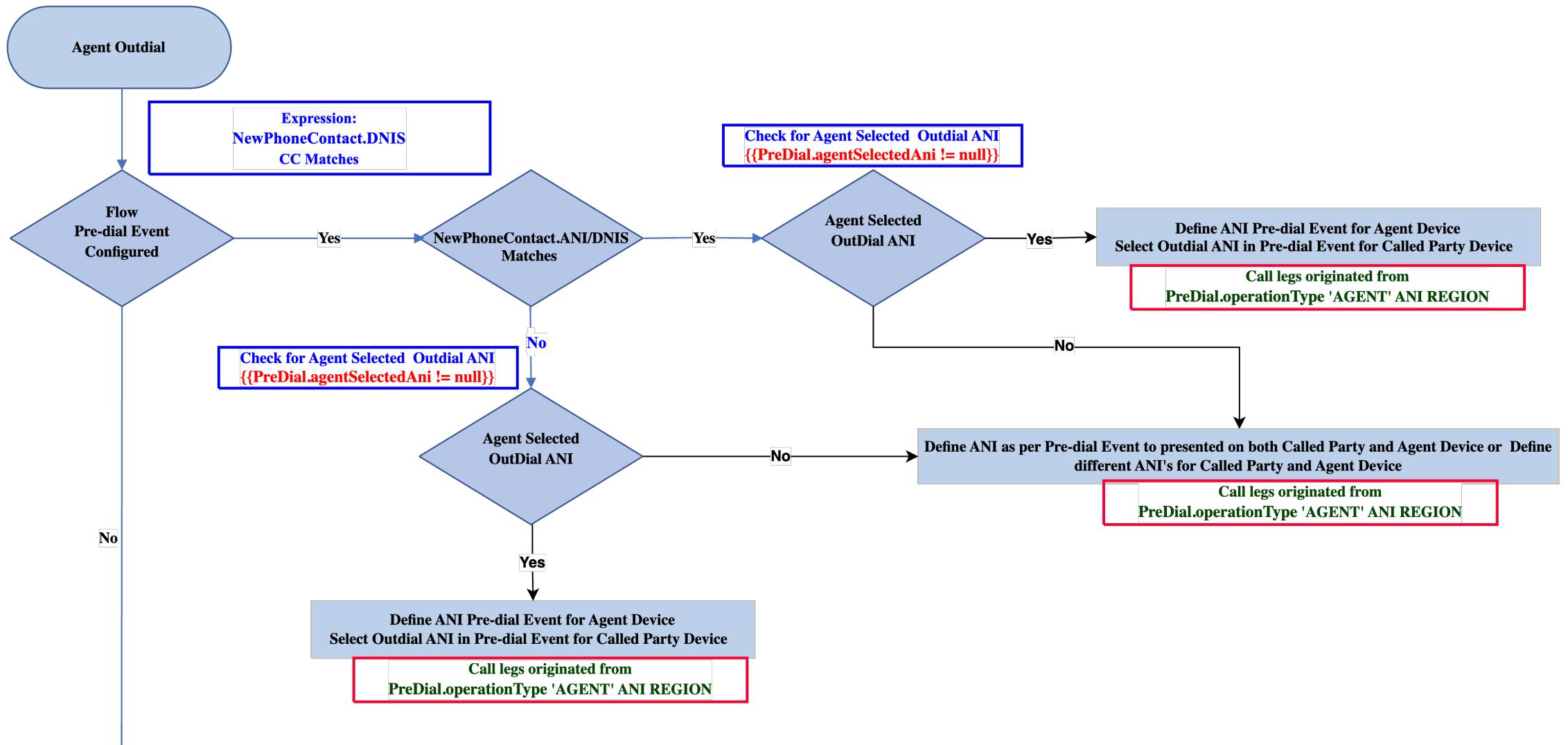
+61234567890

Dialed Numbers	Select Entry Point	All
<input type="button" value="+ New Mapping"/>		
+61234567890	Entry Point	EP_AU_ML

PSTN Region: Australia



## WebexCC Outdial



# Outbound & Skill groups



**Kevin Simson**

# Creating an Outbound Call with Skills

- Use Case:
  - Create an outbound call to be assigned only to agents with specific skills
  - Could be used to extend web callback/scheduled callback
- Features used:
  - Flow Variable
  - Case Statement (optional)
- APIs used:
  - Create Task API
    - Execute Flow option

# Creating an Outbound Call with Skills

- Create a flow with a string variable named lang with no default value
- Use a case statement to read the lang variable and send to the appropriate queue contact node
- Configure the queue contact node to set the skill of to call based on the value of lang
  - This can be a different queue node (as shown) or using a text skill set to equal {{lang}}
- Create an Inbound Entry Point and map the flow
  - Note the ID
  - Do not map a number to the Entry Point

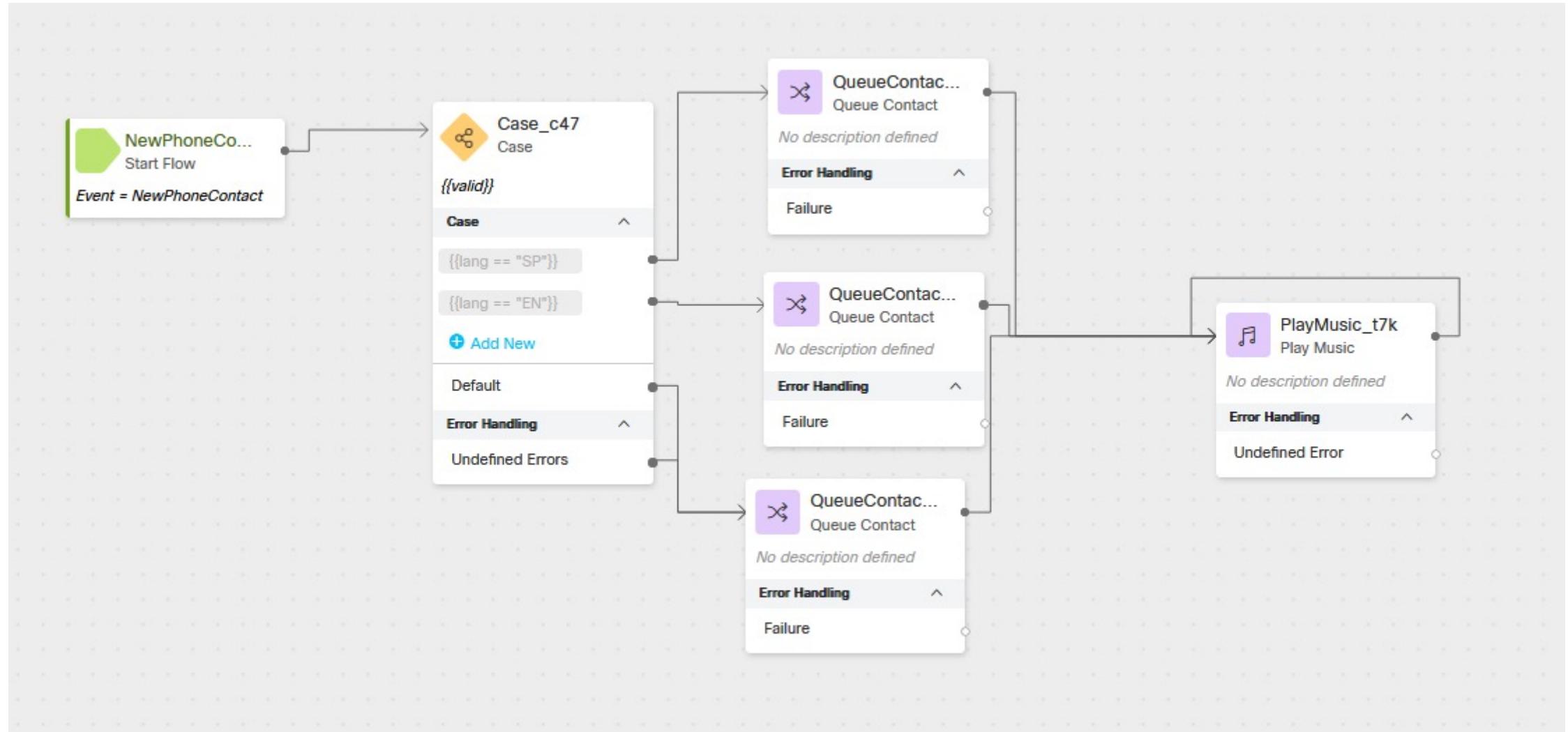
# Creating an Outbound Call with Skills

## ob\_demo\_sk

ID: 8b43bfcc-29c1-43b4-afb4-b482a6846e93 · Last Modified: March 06, 2024 22:51 PM

Entry Point	Name *	ob_demo_sk
	Description	Type here
	Channel Type *	Inbound Telephony
Referenced by	There are no references available. <a href="#">Reference list</a>	
Entry Point Settings	Service Level Threshold ⓘ *	60 Seconds
	Timezone (Business Hours only) *	America/New_York
	Routing Flow	kevsimps_ob_skill_demo
	Version Label	Latest
	Music on Hold	defaultmusic_on_hold.wav

# Creating an Outbound Call with Skills



# Creating an Outbound Call with Skills

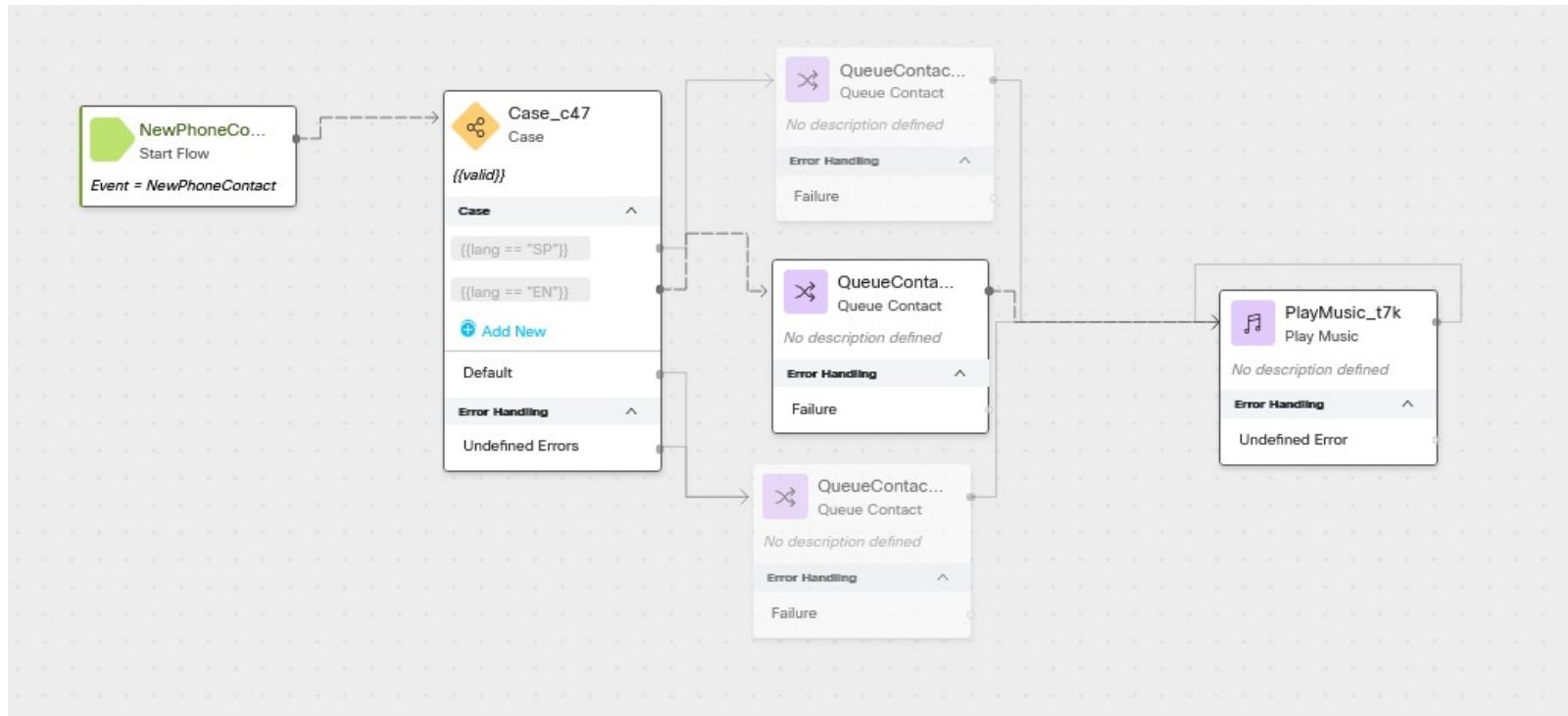
- Use the TASK API to create a new task
- Note that we are passing the lang value of EN in the attributes
- This value will be set to the flow variable of the same name

Request Body

```
{  
  "destination": "6103665851",  
  "entryPointId": "8b43bfcc-29c1-43b4-afb4-b482a6846e93",  
  "attributes": {"lang": "EN"},  
  "outboundType": "EXECUTE_FLOW",  
  "mediaType": "telephony"  
}
```

# Creating an Outbound Call with Skills

- Lang value of EN was true
- The contact was queued with the skill EN



# Emergency message Recording



**Krishna**

# Emergency message Recording

Consider a scenario where there is an emergency and supervisor must record a **special emergency** message to the callers.

# Use case – Play emergency message.

The image shows a screenshot of a REST API documentation interface, likely from a tool like Swagger or Postman. It displays four main sections: 'Create Audio File', 'Sample Code', 'Try Out', 'Update Global Variables', and 'Request Body'.

- Create Audio File:** A POST request to `/organization/{orgid}/audio-file`.
  - Path Parameters:** `orgid` (uuid). Description: Organization ID to be used for this operation. The specified security token must have permission to interact with the organization. Example: "b94a28f3-8b27-43aa-9794-2b9a846050e2".
  - Request Body:** `audioFile` (binary).
- Sample Code:** curl command for creating an audio file.

```
curl --request POST \
--url https://api.wxcc-us1.cisco.com/organization/b94a28f3-8b27-43aa-9794-2b9a846050e2 \
--header 'Accept: */*' \
--header 'Authorization: Bearer YOUR_TOKEN' \
--header 'Content-Type: multipart/form-data' \
--form audioFile= \
--form blobId= \
--form contentType=
```
- Try Out:** A placeholder for running the curl command.
- Update Global Variables:** A PUT request to `/organization/{orgid}/cad-variable/{id}`.
  - Path Parameters:** `orgid` (uuid). Description: Organization ID to be used for this operation. The specified security token must have permission to interact with the organization. Example: "b94a28f3-8b27-43aa-9794-2b9a846050e2".
  - Request Body:** JSON object with fields:
    - `agentViewable`: false
    - `active`: false
    - `agentEditable`: false
    - `name`: ""
    - `reportable`: false
    - `organizationId`: ""
- Request Body:** A large JSON object representing the global variable update payload.

# Use case – Play emergency message.

The screenshot displays two main sections of the Cisco Solution Assurance ESR GT interface:

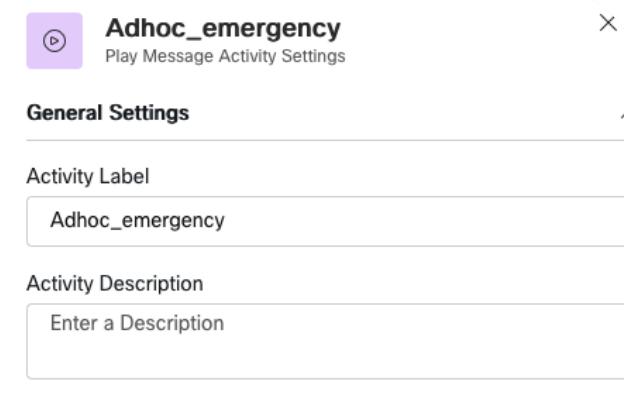
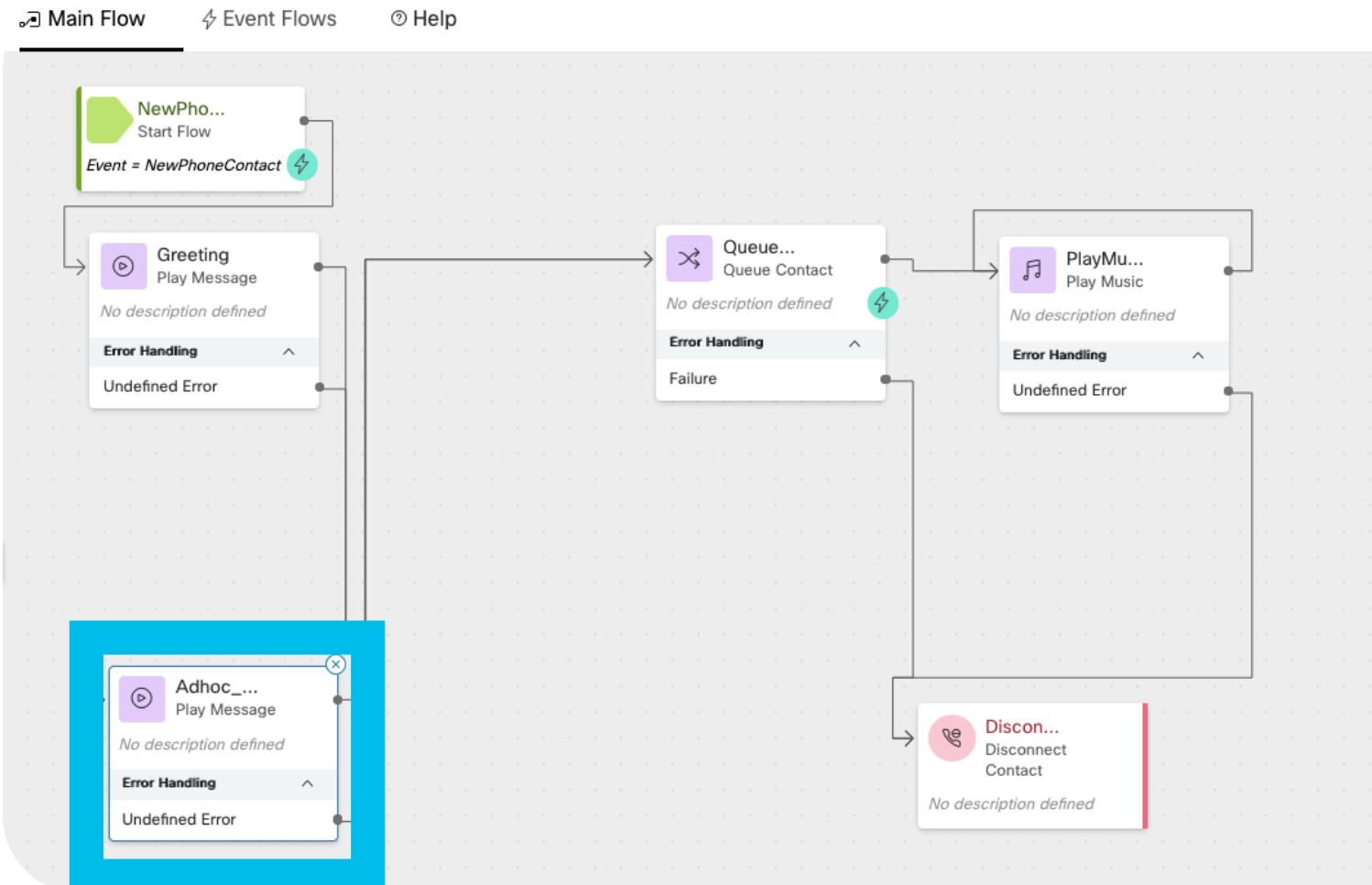
**Global Variable** (Left Panel):

- General Settings:**
  - Name:** EmergencyGlobalVariable
  - Description:** DO NOT DELETE\_SUMMIT
  - Variable Type:** String
  - Default Value:** HoldMusic (highlighted with a blue box)
  - Strings support alphanumerics and spaces.
  - Status:** Active
- Buttons:** **Update** (blue button) and **Cancel** (grey button).

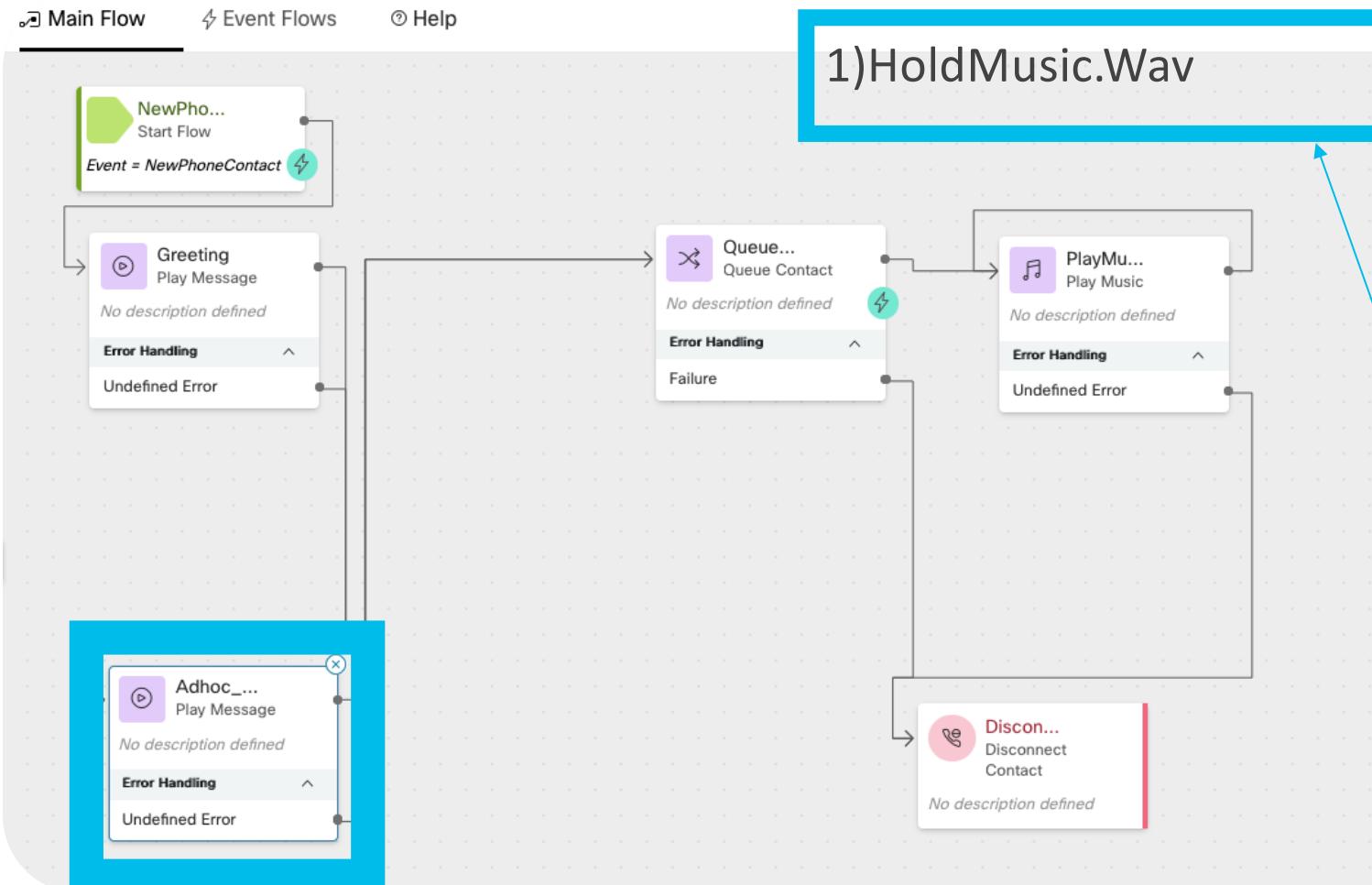
**Audio Files** (Right Panel):

- + New** button.
- Resource Name** column:
  - holdmusic 1.wav
  - HoldMusic.wav (highlighted with a blue box)

# Use case – Play emergency message.



# Use case – Play emergency message.



Adhoc\_emergency

Play Message Activity Settings

General Settings

Activity Label

Adhoc\_emergency

Activity Description

Enter a Description

Prompt

Enable Text-to-Speech

Add the ability to read dynamic messages. These messages can contain variables and be used in a sequence with audio files. If typing variables, use this syntax: {{ variable }}. You can also use SSML to construct the message. If using SSML, insert it inside the <speak></speak> tags.

Add one or more audio files to play in a sequence. ⓘ

1. Audio Prompt Variable

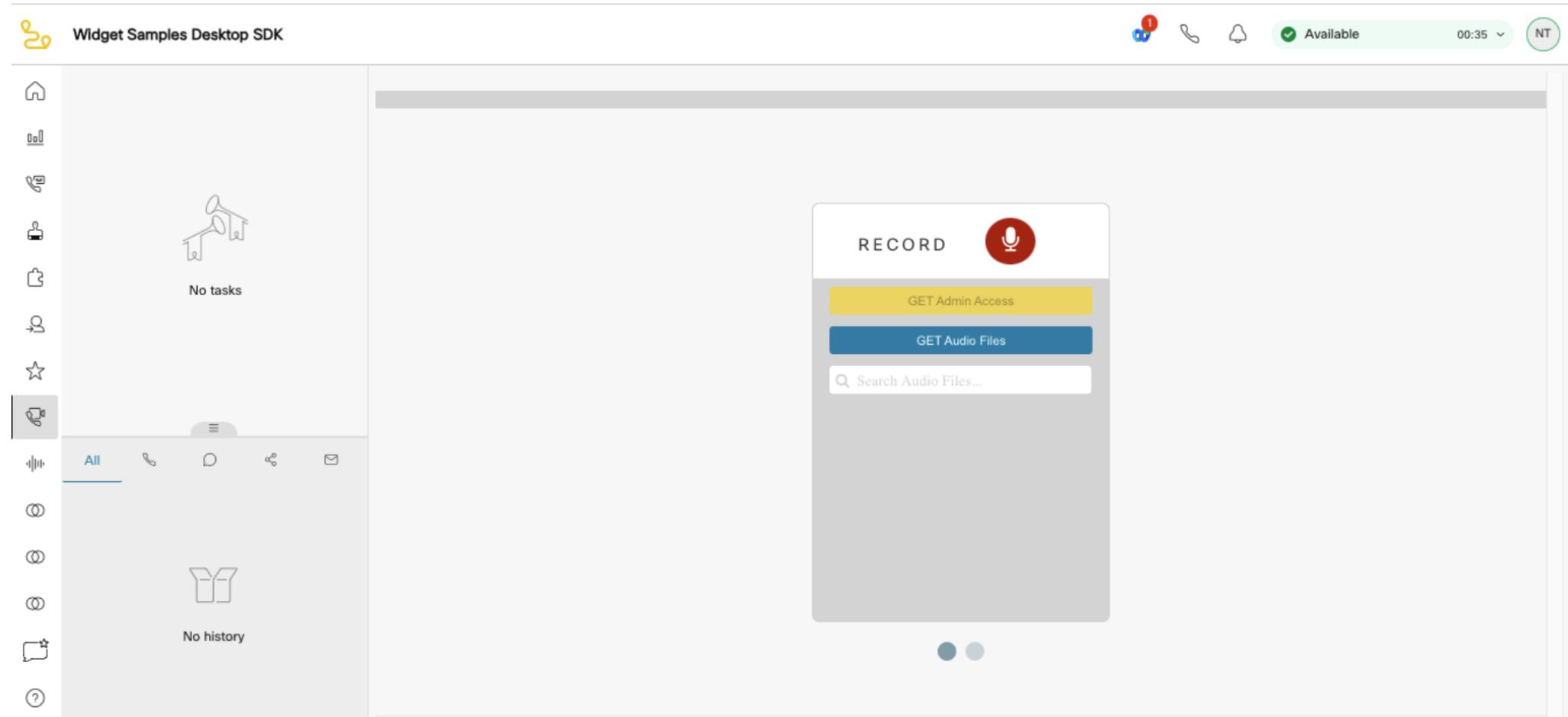
1) {{EmergencyGlobalVariable}}.wav

Add Audio File Add Audio Prompt Variable

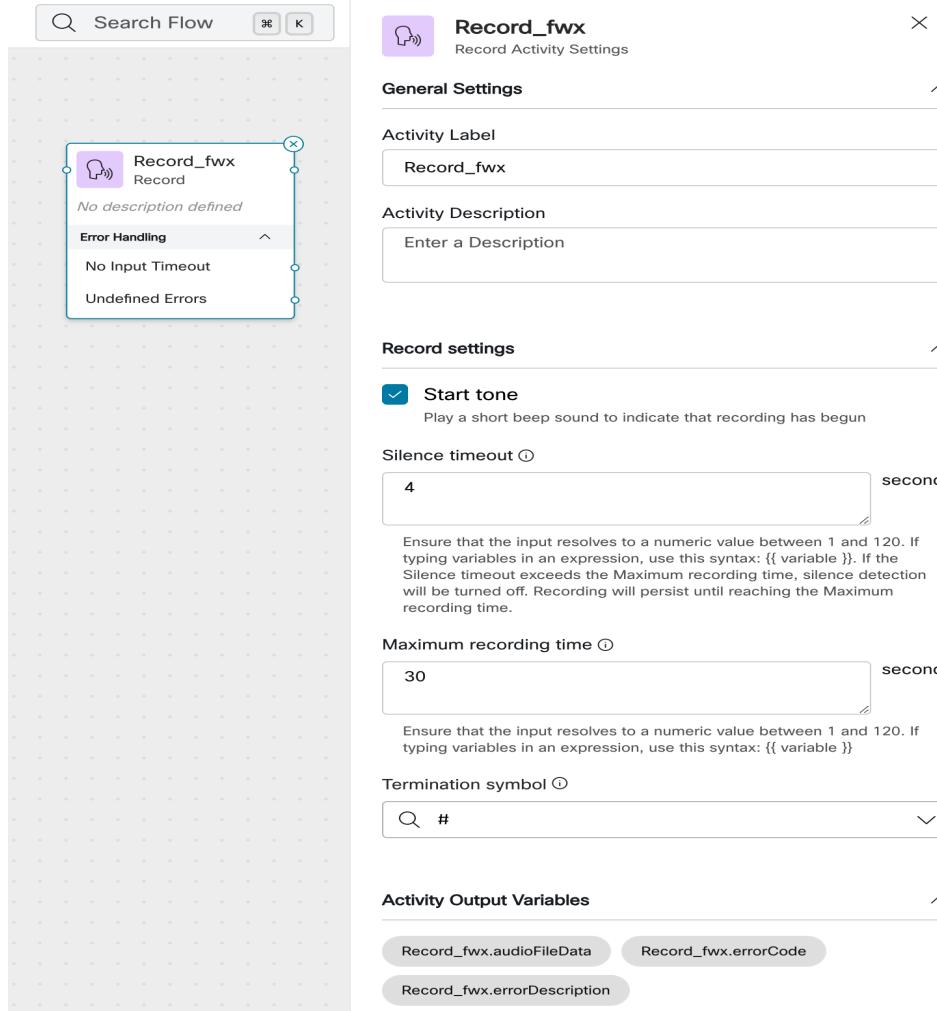
2) Newfile.wav

# Option 2: Audio Widget Sample

Can be downloaded from [github -> widget samples](#)



# Option 1: Record Activity



## Activity Interaction MetaData

Start Time : 12/20/2023 07:35:52

End Time : 12/20/2023 07:35:58

Activity Name : Record\_fwx

## Activity Inputs

Termination symbol : #

Silence timeout : 4

Start tone : true

Maximum recording time : 30

## Activity Outputs

audioFileData : {"type": "FileInfo", "name": "ed066fe2-2a38-42f7-bb27-dc85d99c636b.wav", "contentType": "audio/wav", "metaData": {"codec": "PCMU", "bitrate": "8000"}}

errorCode : 1005

errorDescription :

## Modified Variables

# Upload Recorded Utterance

## HTTP Request

- Synchronous file upload
- 2MB max file size
- New Content Type – File
- CONTENT
- FILE NAME

HTTPRequest\_d8w

HTTP Request Activity Settings

POST

Query Parameters

Key	Value

Add New

HTTP Request Headers

Key	Value

Add New

Content Type

File

Request Body

CONTENT	FILE NAME
AgentOffered.CAD	2000 milliseconds
flowHandoffRecordingInfo	1
Record_fwf.audioFileData	
testRecordingInfo	

Content Type

Select

Output Variable

Select

Path Expression ⓘ

# Feature Overview

- Supported only in RTMS Platform
- Records up to 120 sec
- Stops Recording on detecting Silence
- Recorded utterance can be used as prompt in the same call
- Recorded utterance can be uploaded to the external API
- Recording is stored as .wav file in RTMS S3 bucket
- Accessed using Presigned URL
- Presigned URL is valid for 60mins
- Recordings will be deleted at the end of the call

# Supervisor Alerts



**Krishna**

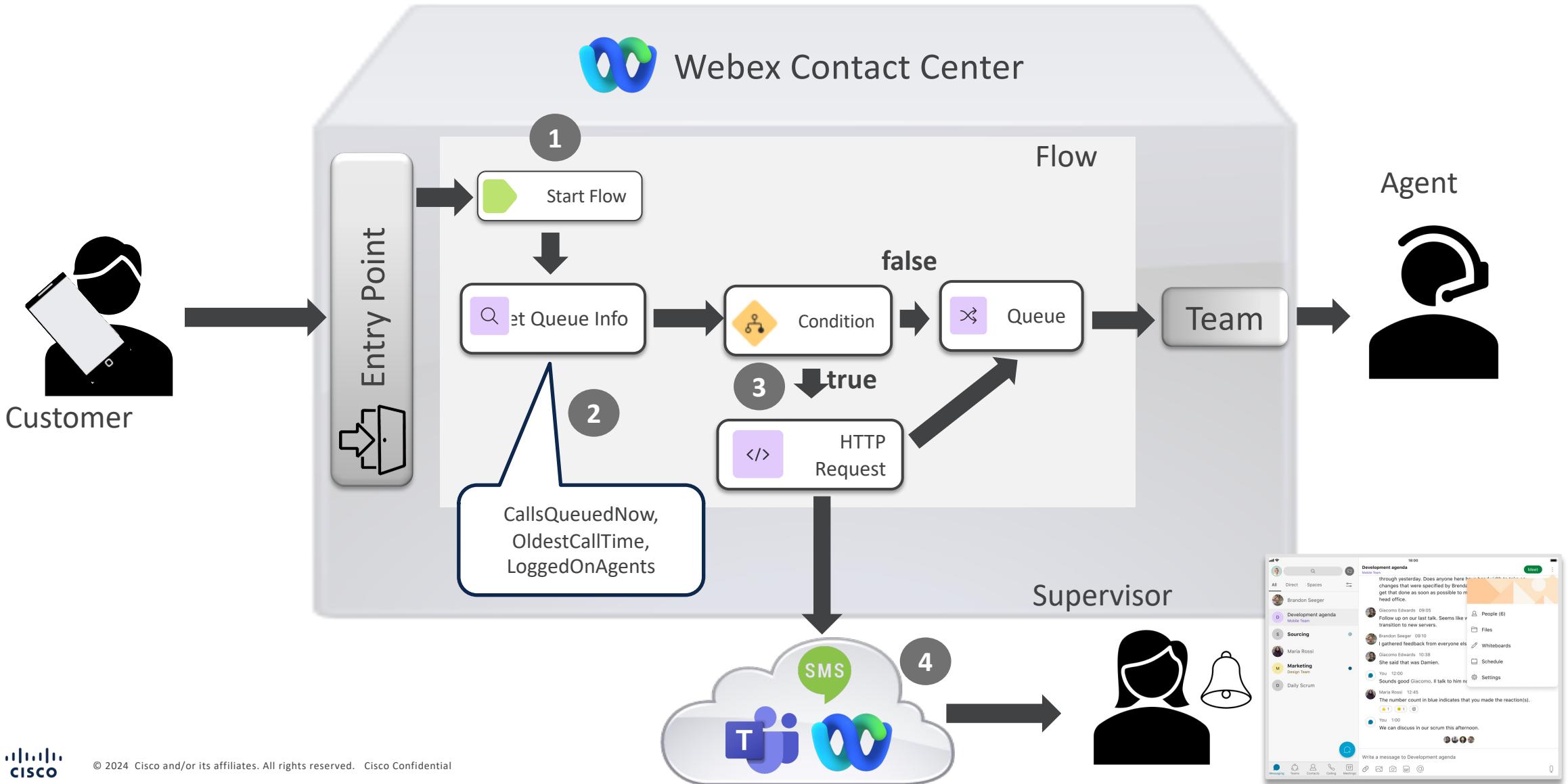
# Supervisor Alerts

Two Components

Types of APIs used to implement this use case

A supervisor wants to know about the critical number of calls in his Queue because. It should be an instant notification through preferred Messenger.

# Solution Design





# Webex Alert DEMO

https://developer.webex.com/docs/bots

Cisco Dev Home CL WxCC Sales WxCC Docs CU Wiki EN AWS Smartsheets API IMI My CL Jira All Bookmarks

Search

Profile

## Build

- Getting Started
- Platform Introduction
- + Embedded Apps
- Bots
- Buttons and Cards
- Integrations
- Service Apps
- Login with Webex
- Guest Issuer
- Widgets
- Webex Connect
- Instant Connect
- Developer Sandbox

## Bots

Give Webex users access to outside services right from their Webex spaces. Bots help users automate tasks, bring external content into the discussion, and gain efficiencies.

[Create a Bot](#)

**Bots Explained**

Bots are similar to regular Webex users. They can participate in 1-to-1 and group spaces and users can message them directly or add them to a group space. A special badge is added to a bot's avatar in the Webex clients so users know they're interacting with a bot instead of a human.

A bot can only access messages sent to it directly. In group spaces, bots must be @mentioned to access the message. In 1-to-1 spaces, a bot has access to all messages from the user.

Bots do not, however, perform actions within Webex on behalf of a Webex user. If you're creating an application that needs to participate in Webex and perform actions with a user's account, check out [Integrations](#).

**Types of Bots**

Bots come in all different shapes and sizes. Here are some ideas for a few different types of bots you can make:

**Notifiers**

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

CISCO

# Webex API – POST message to room

1. Create a BOT and copy API Token: <https://developer.webex.com/docs/bots>
2. Install webex and create a Webex space
3. Execute send msg API: <https://developer.webex.com/docs/api/v1/messages>

The screenshot shows a Postman API client interface. The top bar indicates a POST method and the URL <https://webexapis.com/v1/messages>. The 'Body' tab is selected, showing the following JSON payload:

```
1 ... "roomId": "Y2lzY29zcGFyazovL3VzL1JPT00vMjYxNGIxZjAtOWUzM0xMWVkLTTh0WMtN2RiY2QwMGmzNWMw",
2 ... "text": "There are {{GetQueueInfo.AvailableAgentsCurrent}} available agents in queue!"
```

The bottom section shows the response details: Status: 200 OK, Time: 576 ms, Size: 842 B. The response body is also displayed in JSON format:

```
1 {
2   "id": "Y2lzY29zcGFyazovL3VzL01FU1NBR0Uv0DdjYTg1MzAtYWJkZS0xMWVlLTljNzUtNTUzzjdjZTlhZjU0",
3   "roomId": "Y2lzY29zcGFyazovL3VzL1JPT00vMjYxNGIxZjAtOWUzM0xMWVkLTTh0WMtN2RiY2QwMGmzNWMw",
4   "roomType": "group",
5   "text": "There are {{GetQueueInfo.AvailableAgentsCurrent}} available agents in queue!",
6   "personId": "Y2lzY29zcGFyazovL3VzL1BFT1BMRS9kZDA4Njg2NC02NDB1LTQxYWItYWM3Yi0zOTM0ZGEyZmRjMWM",
7   "personEmail": "TADGG@webex.bot",
8   "created": "2024-01-05T15:24:34.435Z"
9 }
```

# Flow Configuration for Webex API

PDF  
ONLY

Cisco Webex Flow Designer CL\_DEMO Draft Edit: On

Main Flow

```
graph LR; Start((Start Flow)) -- "Event = NewPhoneContact" --> Cond{Condition}; Cond -- True --> HTTP[HTTP Request]; Cond -- False --> Queue[Queue Contact]; HTTP --> Play[Play Music]; Queue --> Play;
```

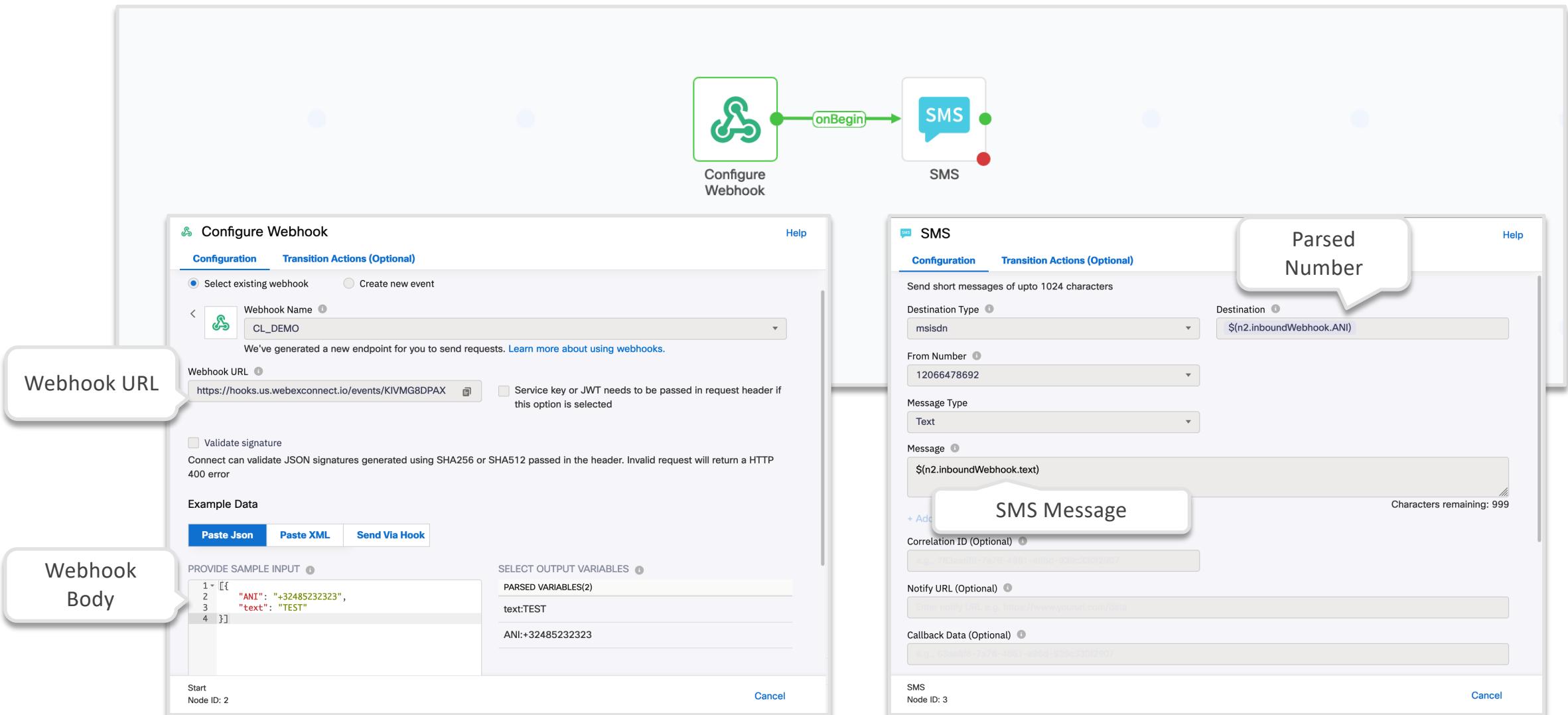
Activity Library CALL HANDLING

- Disconnect Contact
- Virtual Agent
- Blind Transfer
- Play Message
- Collect Digits
- Play Music
- Feedback
- Get Queue Info
- Callback
- Screen Pop
- Queue Contact
- Menu
- Feedback V2
- Queue To Agent

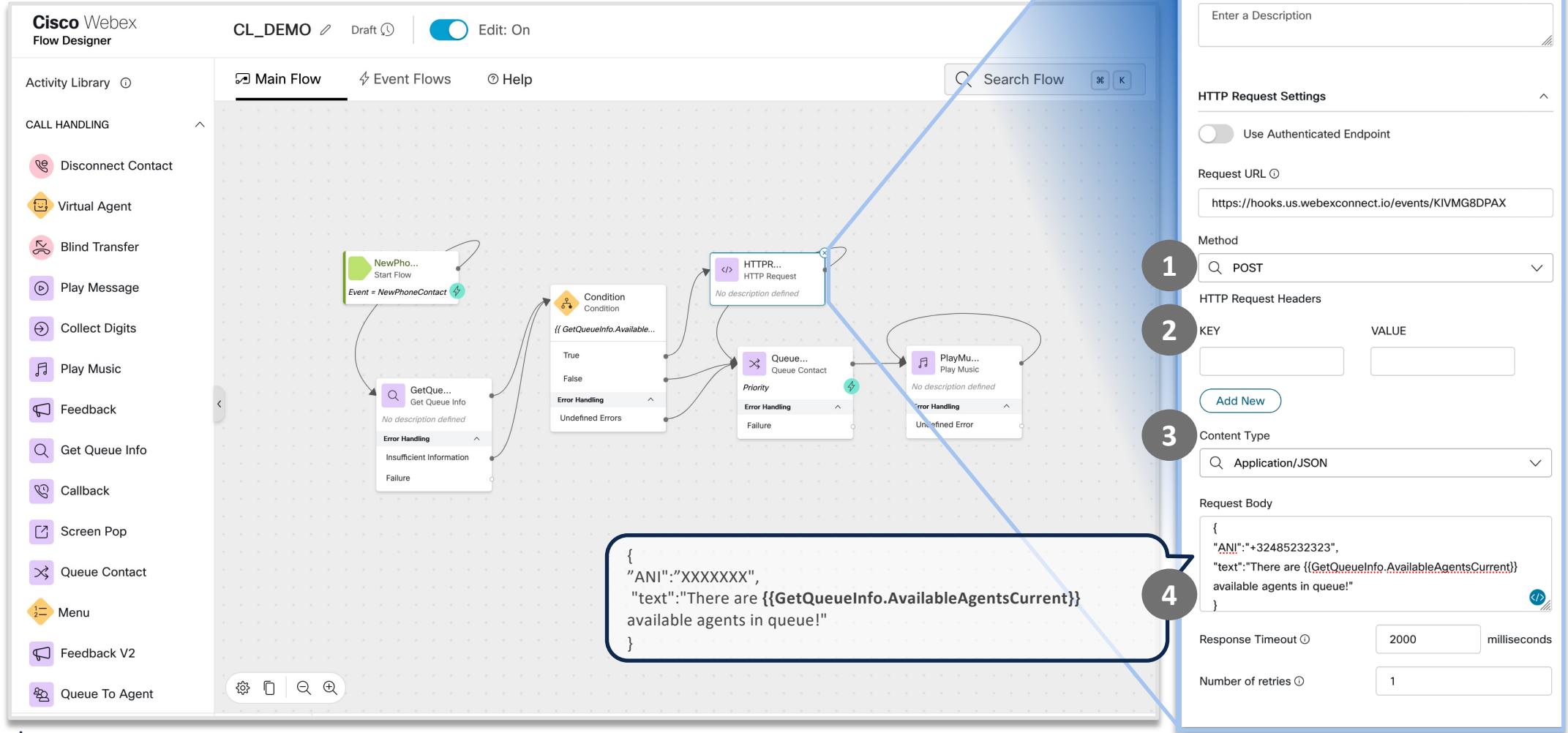
HTTP Request Activity Settings

- Request URL: https://webexapis.com/v1/messages
- Method: POST
- HTTP Request Headers: Authorization: Bearer MzAzYzk4ZWN
- Content Type: Application/JSON
- Request Body: { "roomId": "XXXXXXXX", "text": "There are {{GetQueueInfo.AvailableAgentsCurrent}} available agents in queue!" }
- Response Timeout: 2000 milliseconds
- Number of retries: 1

# Webex Connect Flow



# Flow Configuration for SMS or WhatsApp

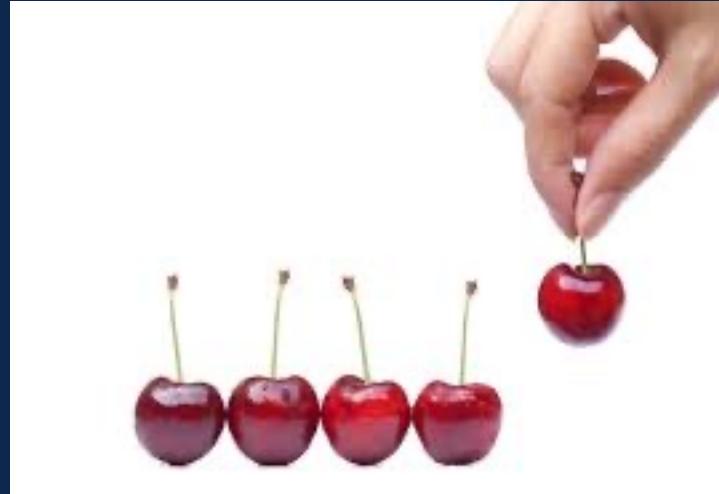


# MS Teams API – POST message to room

1. Open the channel in which you want to add the webhook
2. [Create Incoming Webhooks](#) and add to the team
3. Use the webhook URL for triggering from the Flow Control

```
webhookUrl: "https://xxxxx.webhook.office.com/yyyyyyyy";  
Method: POST  
Body:  
{  
  "@type": "MessageCard",  
  "summary": "Channel Message",  
  "sections": [  
    {  
      "activityTitle": "Sent Channel Message",  
      "facts": [  
        {  
          "name": "Send For",  
          "value": "XXXX"  
        },  
        {  
          "name": "Message",  
          "value": "There are {{GetQueueInfo.AvailableAgentsCurrent}} available agents in queue!"  
        }  
      ]  
    }  
  ]  
}
```

# Cherry Picking Digital Contacts



**Kevin Simson**

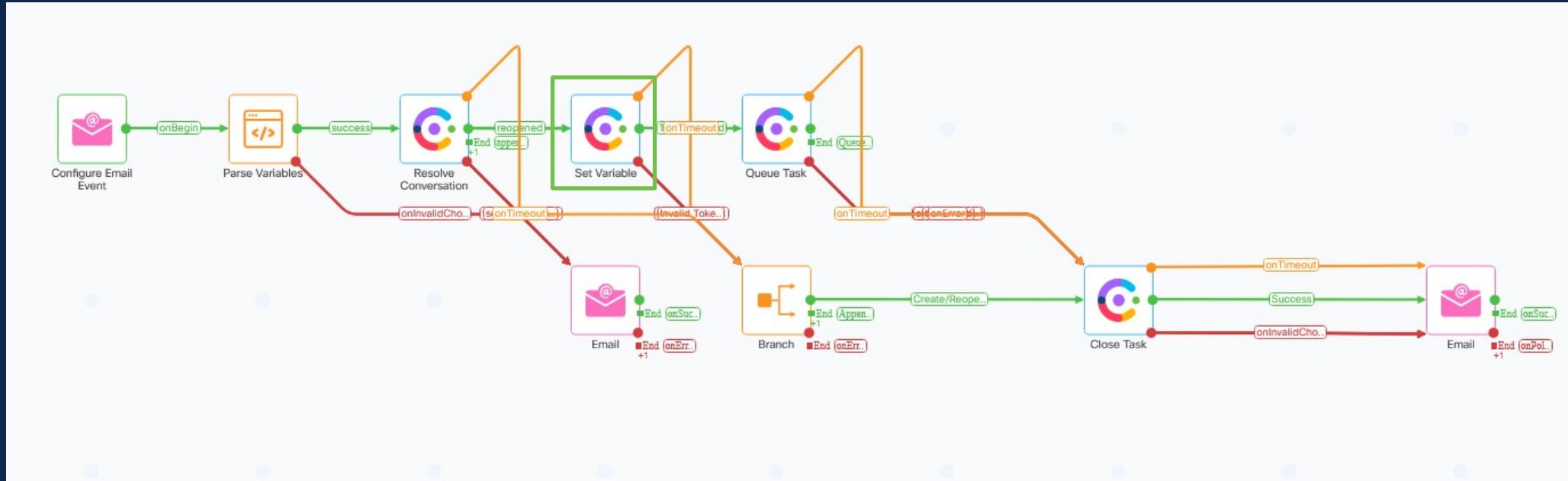
# Cherry Picking Digital

- Use Case:
  - To move an email which is in queue to a specific agent
- Features Used
  - Analyzer
  - Global Variable
  - Webhook
  - SBR
- APIs used in demo
  - Webhook

# Cherry Picking Digital

- In the email flow
  - Add a set global variable node
  - Set the Global Variable to the Email Subject
- Agents must have a text skill (in this example we are using the agent login)
- Agents must belong to a team which is assigned to the skilled email queue

# Cherry Picking Digital



# Cherry Picking Digital

Method Name  
Set Global Variable

Node Runtime Authorization  
WxCC Authorisation

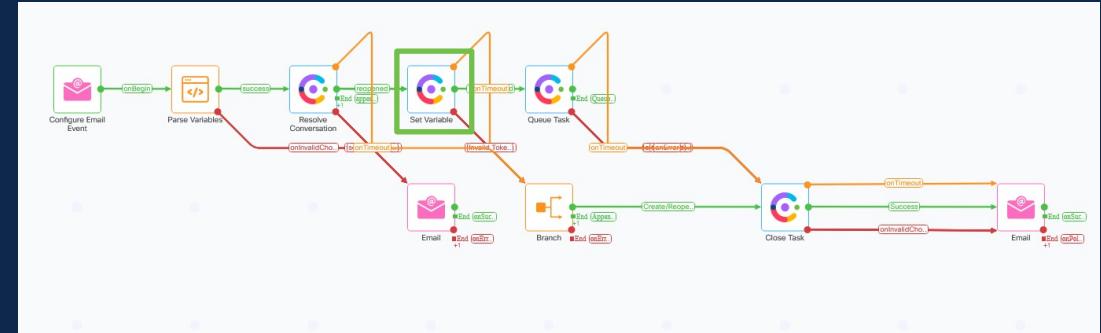
Task ID  
\$(taskId)

**Fetch Global Variables**

Variable Name	Type	Description	Default Value
<input type="checkbox"/> HttpResponse	String		
<input type="checkbox"/> AccountNumber	String		
<input checked="" type="checkbox"/> Email_Subject	String		

**Set Variable**

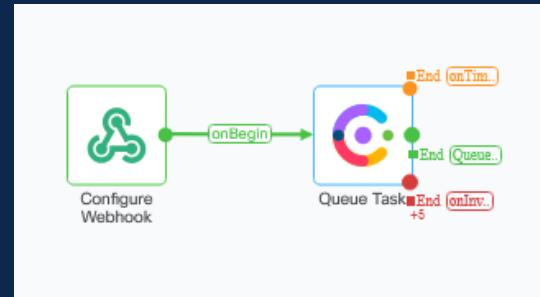
Email\_Subject ⓘ  
\$(subject)



# Cherry Picking Digital

- Create webhook which will accept
  - Conversation ID
  - Session ID
  - Skill
- Queue the task
  - Set using the variables captured from the webhook
    - Conversation ID
    - Session ID
    - Skill

# Cherry Picking Digital



Configure webhook settings to trigger this flow

Select existing webhook  Create new event

Webhook Name  We've generated a new endpoint for you to send requests. [Learn more about using webhooks.](#)

Webhook URL

Service key or JWT needs to be passed in request header if this option is selected

Validate signature  
Connect can validate JSON signatures generated using SHA256 or SHA512 passed in the header. Invalid request will return a HTTP 400 error

Example Data

[Paste Json](#) [Paste XML](#) [Send Via Hook](#)

PROVIDE SAMPLE INPUT

```
1 {  
2   "task": "2407c1d2-d388-4814-acaa-c82ba3b80d87",  
3   "conversation": "CSRCQOHYLA9YHPDP",  
4   "skill": "kevin"  
5 }
```

SELECT OUTPUT VARIABLES

PARSSED VARIABLES(3)

task:2407c1d2-d388-4814-acaa-c82ba3b80d87  
skill:kevin  
conversation:CSRCQOHYLA9YHPDP

Method Name  Node Runtime Authorization

Task ID  Conversation ID

Media Type  MEDIA CHANNEL

Queue details  
Queue Name  Contact Priority   
Queue routing Type: Skill based routing

Skill settings  
Skill  Skill Condition  Skill Value   
[+ Add New](#)

Skill relaxation

# Cherry Picking Digital

- Create a Visualization in the Analyzer
  - Queue Name
  - Conversation ID
  - Contact Session Id
  - Email Address
  - Email Subject (Global Variable)
  - Required Skills (optional)
- Create Web page or Widget
  - Collect
    - Conversation ID, Contact Session ID, Skill (agent name)

# Cherry Picking Digital

Queue Name	Conversation ID	Contact Session ID	Email Address	Email Subject	Value of Required Skills
kevsimpsEmail	CSR0XNHLXFTYS7DX	c7aaf047-6c0e-417a-9457-d12aea378bc8	miturnbo@cisco.com	Carnival tonight - Bring your exotic animals!	N/A
kevsimpsEmail	CSRBR77VW48S13ZQ	e58e07d3-e4d5-4541-8a06-6b497b61dc36	jiwyatt@cisco.com	a fun subject line	N/A
kevsimpsEmail	CSRR0WDBF6WRXDG7	db639b68-54c3-4d37-af02-2d136437f099	kgouda@cisco.com	Don't open this email...unless you love a good laugh!	N/A
kevsimpsEmail	CSRFD10BY97IS46N	1d3d4eb8-7372-4420-896d-fc4c1b826a85	evonkuge@cisco.com	Inquiry regarding free puppies	N/A
kevsimpsEmail	CSR4CPSTM6ZABP6WN	6aa9bfe7-9aa8-4a2e-89f6-d6e9d26d564a	ntheolog@cisco.com	Mars rover traveled more than 28 miles on the Martian planet!	N/A

## Cherry Pick Email

Session ID

Conversation ID

Agent

**Cherry Pick It!**

# Cherry Picking Digital

Queue Name	Conversation ID	Contact Session ID	Email Address	Email Subject	Value of Required Skills
kevsimpsEmail	CSR0XNHLXFTYS7DX	c7aaaf047-6c0e-417a-9457-d12aea378bc8	miturnbo@cisco.com	Carnival tonight - Bring your exotic animals!	N/A
kevsimpsEmail	CSRBR77VW48S13ZQ	e58e07d3-e4d5-4541-8a06-6b497b61dc36	jiwyatt@cisco.com	a fun subject line	N/A
kevsimpsEmail	CSRR0WDBF6WRXDG7	db639b68-54c3-4d37-af02-2d136437f099	kgouda@cisco.com	Don't open this email...unless you love a good laugh!	N/A
kevsimpsEmail	CSRFD10BY97IS46N	1d3d4eb8-7372-4420-896d-fc4c1b826a85	evonkuge@cisco.com	Inquiry regarding free puppies	N/A
kevsimpsEmail	CSR4CPSM6ZABP6WN	6aa9bfe7-9aa8-4a2e-89f6-d6e9d26d564a	ntheolog@cisco.com	Mars rover traveled more than 28 miles on the Martian planet!	N/A

## Cherry Pick Email

Session ID  
-4d37-af02-2d136437f099

Conversation ID  
CSRR0WDBF6WRXDG7

Agent  
kevsimps\_csam\_americas

Cherry Pick It!

```
function sendIt() {
  event.preventDefault()
  let myHeaders = new Headers();
  myHeaders.append("Content-Type", "application/json");

  let raw = JSON.stringify({
    "task": document.forms.cherry.sessionID.value,
    "conversation": document.forms.cherry.conv.value,
    "skill": document.forms.cherry.agent.value
  });
  let requestOptions = {
    method: 'POST',
    headers: myHeaders,
    body: raw,
    redirect: 'follow'
  };
  fetch("https://hooks.us.webexconnect.io/events/N2VK8MQKU", requestOptions)
    .then(response => response.text())
    .then(result => console.log(result))
    .catch(error => console.log('error', error));
}
```

# Cherry Picking Digital

The screenshot shows a web browser window with two tabs: "Customer Journey Analyzer" and "Webex Contact Center : Task". The main content area displays a table titled "Module1" with the following data:

Queue Name	Conversation ID	Contact Session ID	Email Address	Email Subject	Value of Required Skills
kevsimpsEmail	CSR0XNHLKFTYS7DX	c7aaef047-600e-417a-9457-d12aea378bc8	mturnbo@cisco.com	Carnival tonight - Bring your exotic animals!	N/A
kevsimpsEmail	CSRBR77VW48S1ZQ	e58e07d3-e4d5-4541-8a06-6b497b61dc36	jiwyatt@cisco.com	a fun subject line	N/A
kevsimpsEmail	CSRR0WDBF6WRXDG7	db639b68-54c3-4d37-af02-2d136437f099	kgouda@cisco.com	Don't open this email...unless you love a good laugh!	N/A
kevsimpsEmail	CSRFD10BY97IS46N	1d3d4eb8-7372-4420-896d-fc4c1b826a85	evonkuge@cisco.com	Inquiry regarding free puppies	N/A
kevsimpsEmail	CSR4CPSM6ZABP6WN	6aa9bfe7-9aa8-4a2e-89f6-d6e9d26d564a	ntheolog@cisco.com	Mars rover traveled more than 28 miles on the Martian planet!	N/A

At the bottom of the browser window, there is a taskbar with various icons, including a weather icon (42°F Cloudy), a search bar, and several application icons. The system tray shows the date and time as 11:26 PM on 2/28/2024.

# Payment IVR Transfer



**Carlos**

# Payment IVR

- Agent Reservation
- Customer has payment IVR that collects payment information and completes payment requests\*.
- Agents should be able to transfer a telephony contact to this payment IVR.
- Upon completion of payment, contact should be routed back to the same agent who initiated the transfer.
- Design needs to ensure minimum contact wait time upon completion of payment when routing back to the same agent.

# Requirement Analysis

- Customer has a payment IVR service.
- Agents should be able to transfer a telephony contact to this payment server for self-service.
- Upon completion of payment, contact should be routed back to the same agent who initiated the OTP.
- Design needs to ensure minimum contact wait time upon completion of payment when routing back to the same agent.

## Challenges:

- How do agents transfer contact to the payment IVR for self-service?
- Upon contact completing self-service , how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?

# Brainstorm Solution

- Webex CC
  - Agent Desktop Transfer to EP
  - Flow Control Route to Agent
  - Reporting via Dashboard
    - Global Variables for Reporting
  - Agent Desktop iFrame Widget

# Payment IVR Entry Point

Entry Point

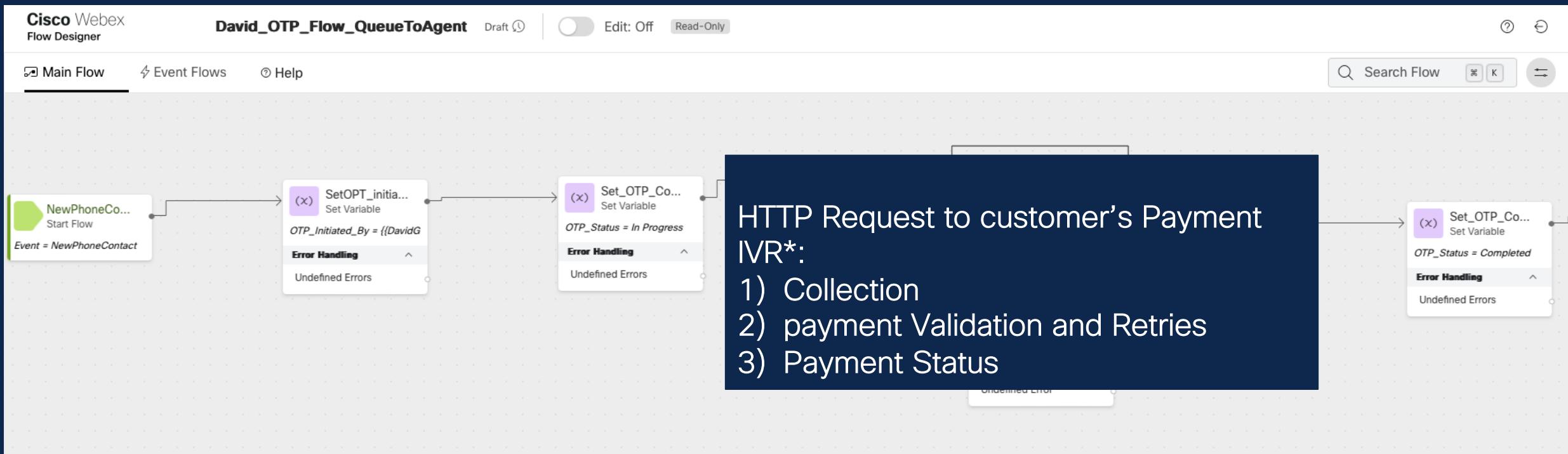
General Settings

Name	David OTP_EP
Description	
Type	Entry Point
Channel Type	Telephony
Status	Active

Advanced Settings

Service Level Threshold	100 seconds
Time Zone (Routing Strategies Only)	Australia/Sydney
Flow	David OTP_Flow_QueueToAgent
Version Label	Live
Music on Hold	CCMusic.wav

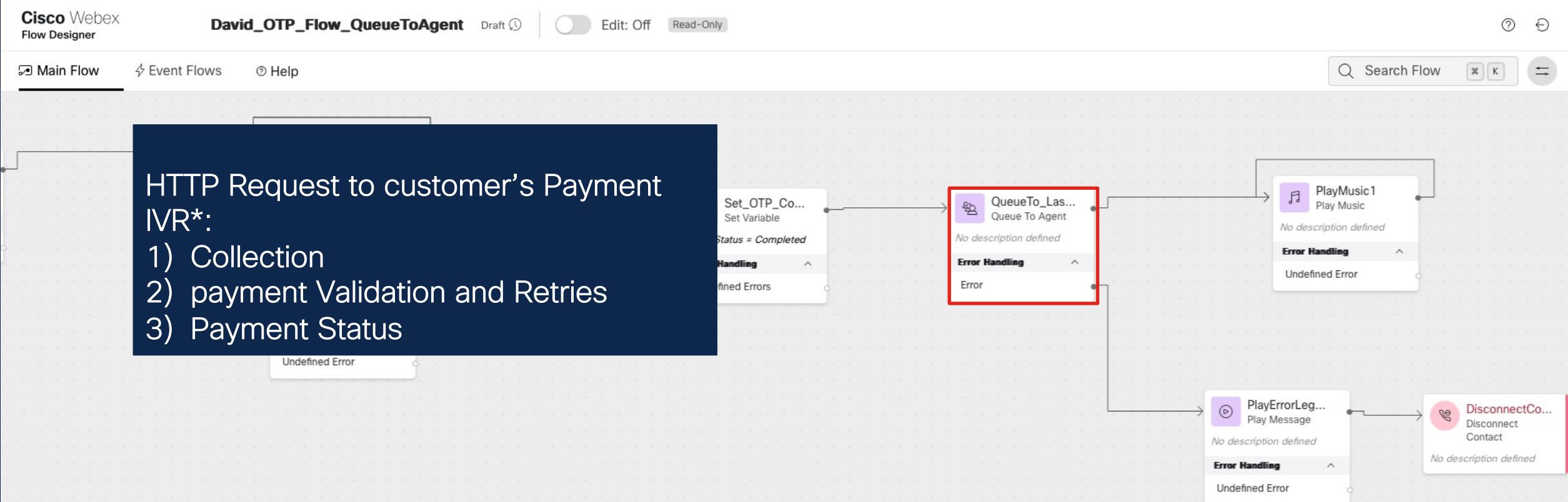
# Payment Flow Control



# Payment Flow Control

## Challenges:

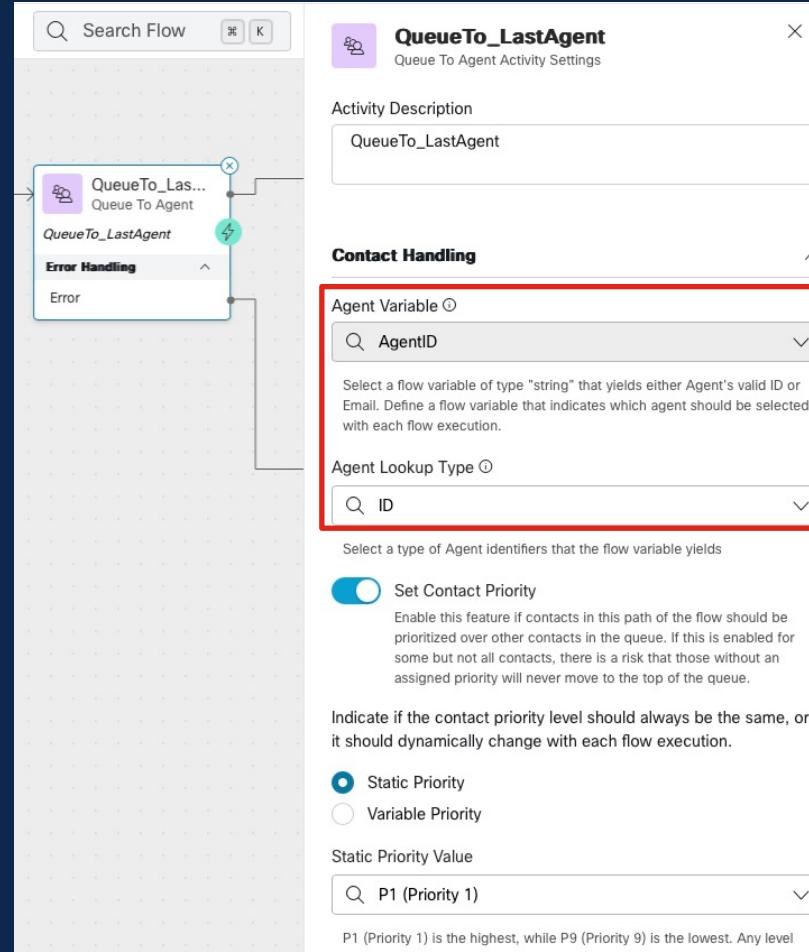
- How do agents transfer contact to Payment server for self-service?
- Upon contact completing self-service, how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?



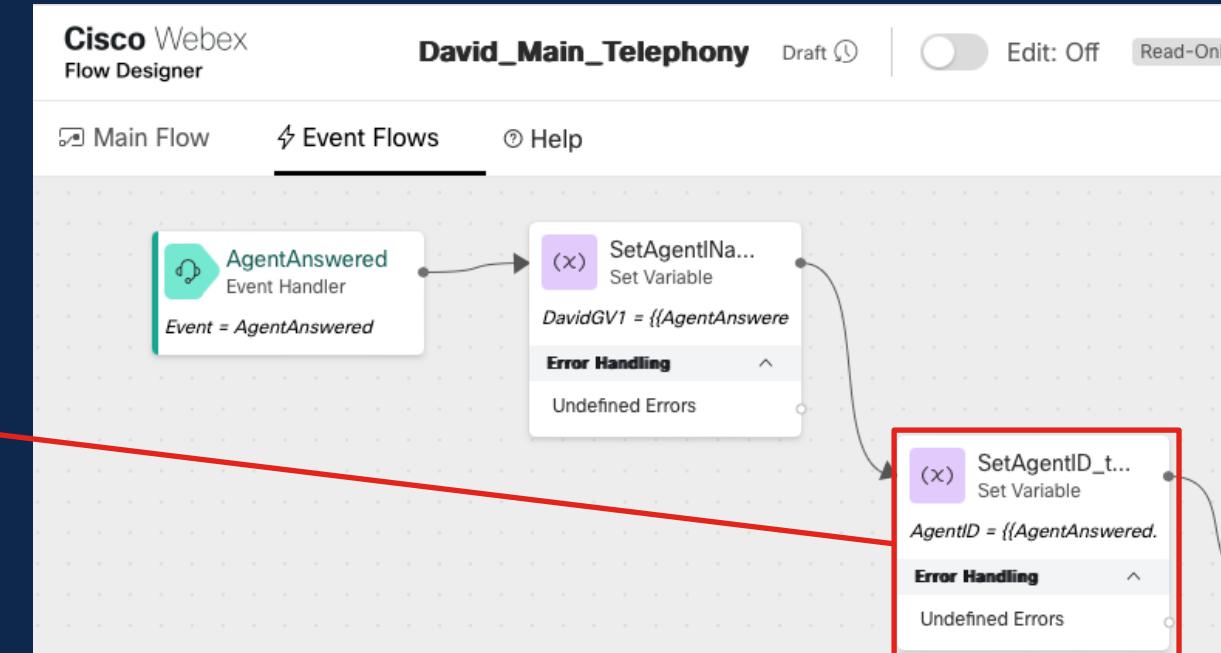
# Payment Flow Control

## Challenges:

- How do agents transfer contact to payment server for self-service?
- Upon contact completing self-service, how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?



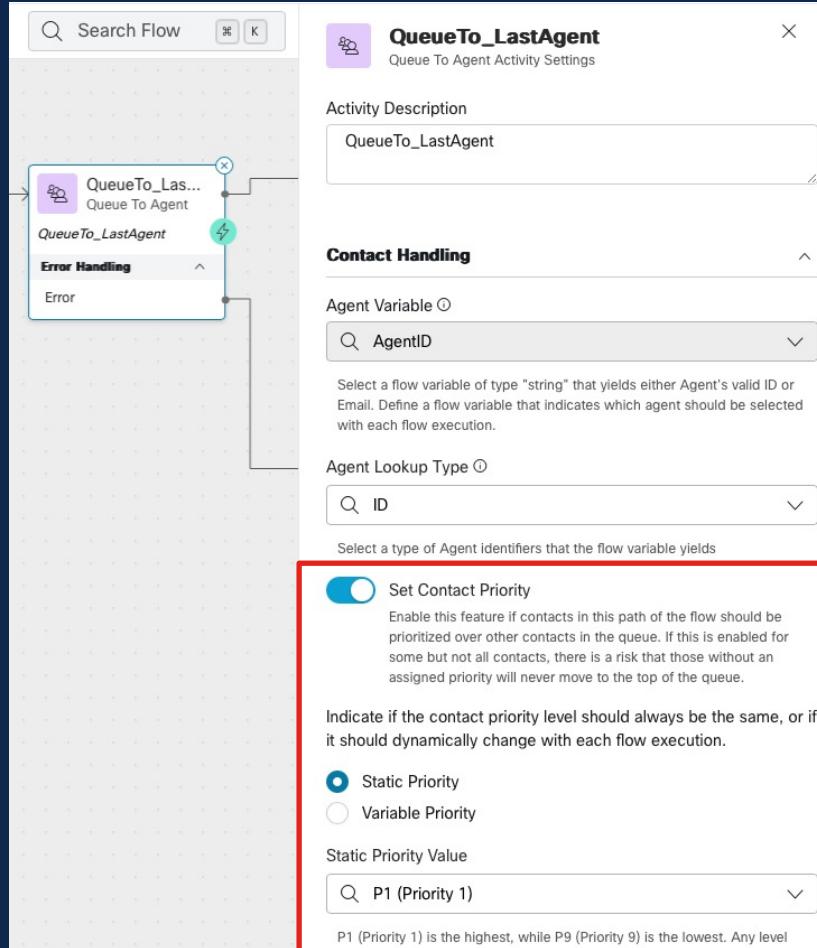
## Main Inbound Flow



## Main Inbound Flow Queue vs Agent Queue?

# Agent Reservation

## 1. Setting Higher Priority for payment contact



## 2. Toggle agent to Available when payment contact re-queued to Agent Queue

- Agent education not to wrap-up payment transfer call immediately;
- Track progress of payment prior to state change

### Challenges:

- How do agents transfer contact to payment server for self-service?
- Upon contact completing self-service payment, how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?

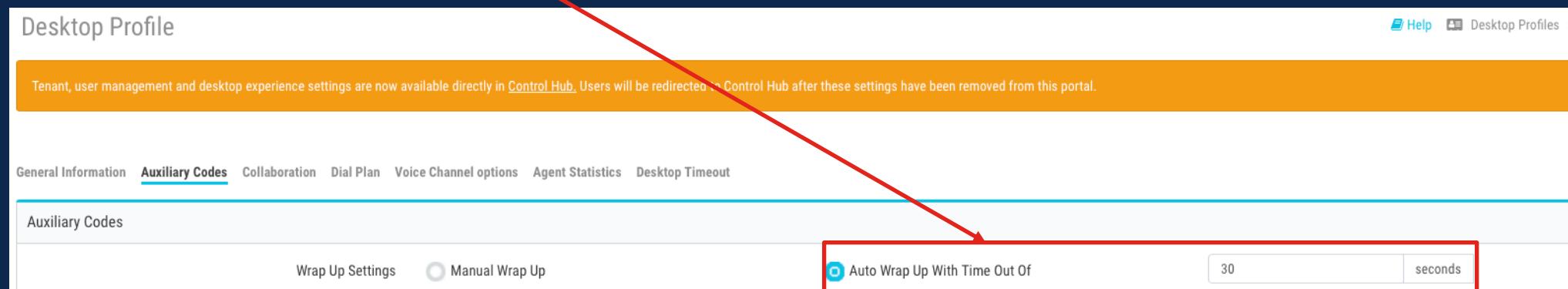
# Agent Reservation

## 2. Toggle agent to Available when Payment contact re-queued to Agent Queue

- Agent education not to wrap-up payment transfer call immediately;
  - Implement auto wrap-up timer based on possible payment duration

### Challenges:

- How do agents transfer contact to payment server for self-service authentication?
- Upon contact completing self-service payment, how do we route contact back to the same agent?
- **How do we minimize contact wait time after payment completion?**
- Any way for the agent to track progress of the payment transaction?



# Agent Reservation

## 2. Toggle agent to Available when payment contact re-queued to Agent Queue

- Track progress of payment prior to state change
  - Customized Analyzer Report to track progress

### Challenges:

- How do agents transfer contact to payment server for self-service?
- Upon contact completing self-service payment, how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?

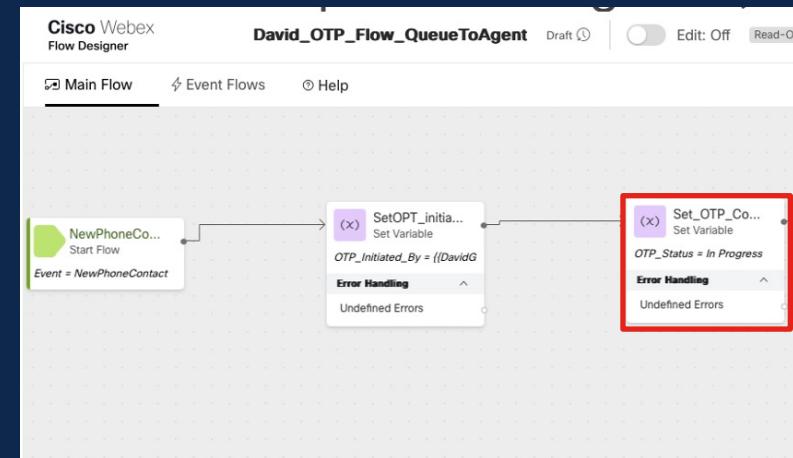
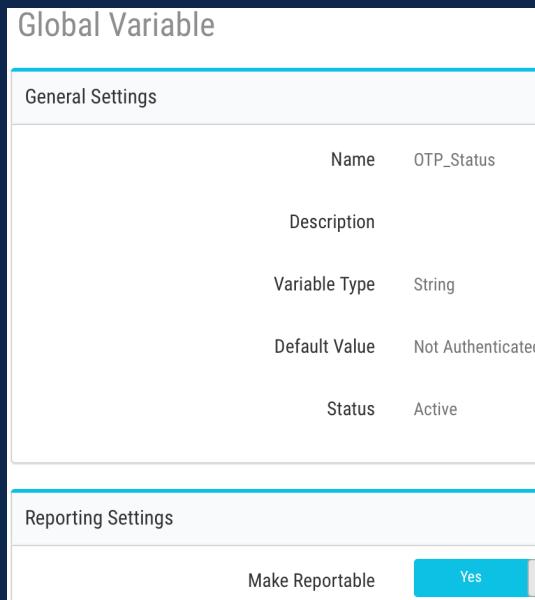
The screenshot shows a software interface titled "Realtime OTP Completed Queue". At the top, there are buttons for "Save", "Preview", and "More". Below the title, there are sections for "Row Segments" and "Column Segments", both labeled "Profile Variables". Under "Output Type", it says "Table". The main area displays a table with three columns: "ANI", "OTP Status", and "OTP Initiated By". The data in the table is as follows:

ANI	OTP Status	OTP Initiated By
ANI 1	8450	OTP Initiated By 1
ANI 2	6959	OTP Initiated By 2
ANI 3	3129	OTP Initiated By 3
ANI 4	7423	OTP Initiated By 4

# Agent Reservation

## 2. Toggle agent to Available when OTP contact re-queued to Agent Queue

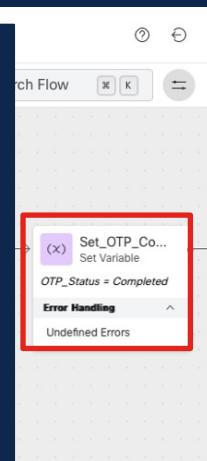
- Track progress of OTP prior to state change
  - Customized Analyzer Report via GV to track progress



### Challenges:

- How do agents transfer contact to payment server for self-service?
- Upon contact completing self-service, how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?

HTTP Request to customer's Payment IVR\*:  
1) Collection  
2) payment Validation and Retries  
3) Payment Status



# Agent Reservation

## 2. Toggle agent to Available when payment contact re-queued to Agent Queue

- Track progress of payment prior to state change
  - Customized Analyzer Report via GV to track progress
  - Agent tracking via Desktop iFrame Widget

### Challenges:

- How do agents transfer contact to payment server for self-service authentication?
- Upon contact completing self-service payment, how do we route contact back to the same agent?
- How do we minimize contact wait time after payment completion?
- Any way for the agent to track progress of the payment transaction?

The image displays two screenshots of Cisco Contact Center software. The left screenshot shows the 'Contact Center Desktop' interface with a navigation bar and a central area indicating 'No tasks'. A small lock icon is highlighted with a red box. The right screenshot shows the 'Analyzer' interface with a title 'Realtime OTP Completed Queue'. It contains a table with three columns: ANI, OTP Status, and OTP Initiated By. The data in the table is as follows:

ANI	OTP Status	OTP Initiated By
+61370385586	In Progress	anzuser2 David
+61396594200	Completed	David NG

# Implementation Summary

- Call inbound to Webex contact Center, answered by IVR and goes into queue.
- Agent A goes Available and being routed the call. Agent A performs Transfer to EP for contact self-service.
- Contact completes payment and routes back to the same Agent A.
- Reservation of agent performed via configuring Auto-wrap up of 30 secs to prevent agent from being routed the next call.
- Agent Desktop iFrame widget provides realtime payment transaction status via Analyzer report to notify agent when the Contact has completed the payment.
- Agent A can selectively changes state to Available when payment status changes to Completed.

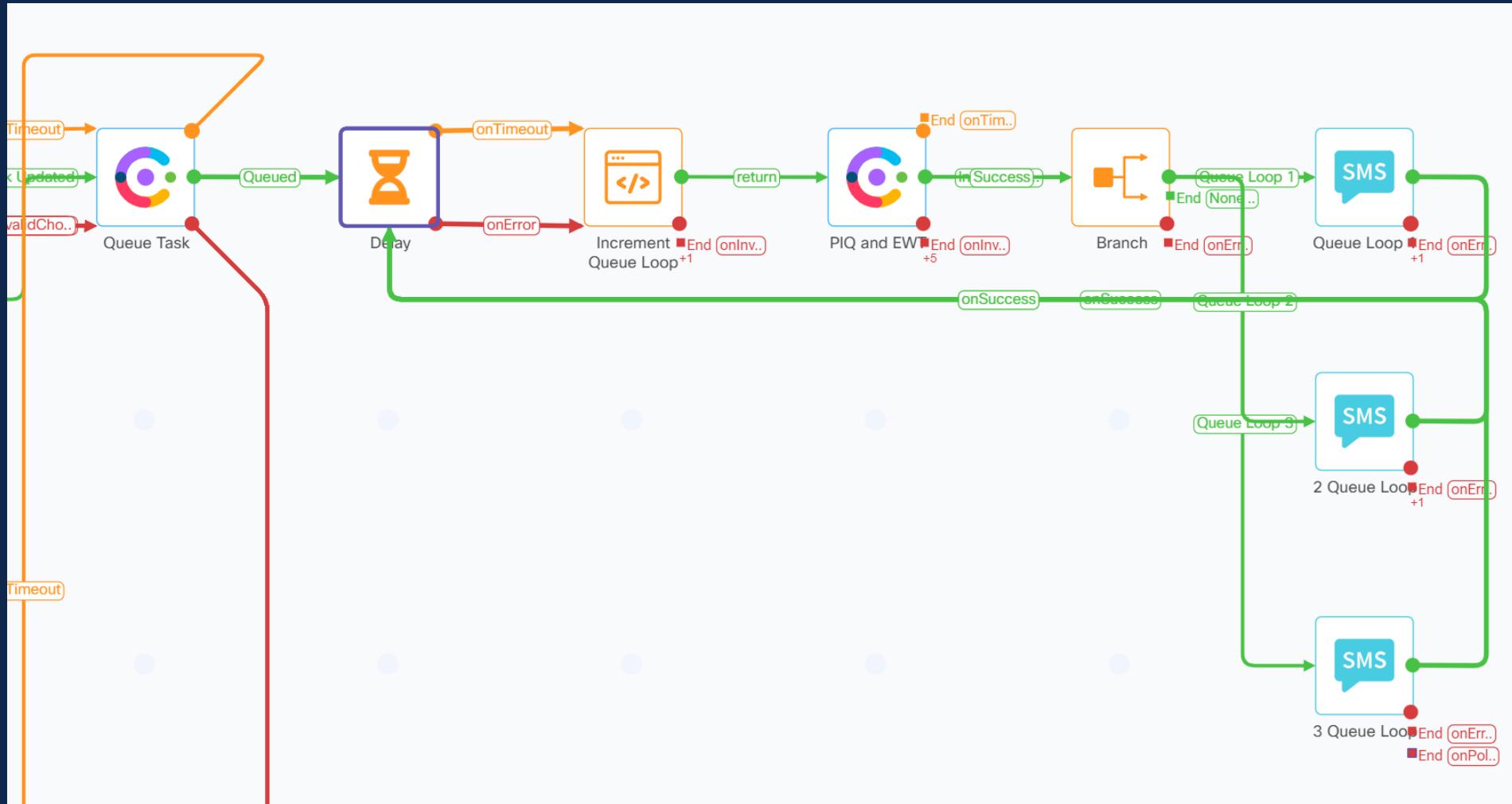
# Digital Queue Messages on hold



Chandra

**Enhanced Customer Experience:** Digital messages on hold can provide customers with valuable information, entertainment, or reassurance while they wait. This can reduce perceived wait times, prevent hang-ups, and improve overall satisfaction by engaging the caller with relevant content, such as answers to frequently asked questions, new promotions, or updates about the company's services.

# In Queue Message Flow



# Delay between messages



Delay between  
queue loops

**Delay**

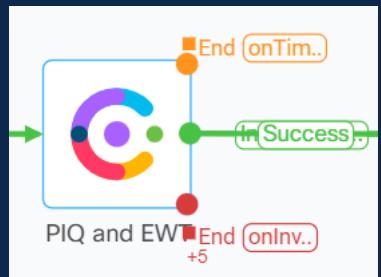
**Configuration**    **Transition Actions (Optional)**

Set the duration for which a flow session must wait before proceeding to the next node.

Delay time     Wait for date

Time Out (In Seconds) i

# PIQ and EWT



webex Control Hub

< Main Menu < Queues

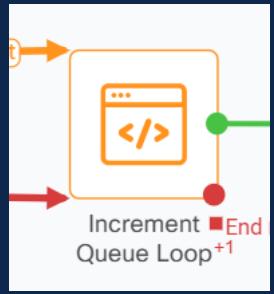
**bmclella\_SMS**

ID: 74fbef5f-63a3-441d-8636-1f8f8a21f63b Last Modified: September 29, 2023 11:12 AM

**General**

Name *	bmclella_SMS
Description	Type here
Queue Type *	Inbound Queue
You can't change Queue Type once the queue is created	
Channel Type *	Social Channel
You can't change Channel Type once the queue is created	
Referenced by	There are no references available.

# Main processing evaluate flow



Increment Queue Loop

Configuration Transition Actions (Optional)

Enter JavaScript to perform logic and return values as output. Configured script c

```
1 //Keeps defining my int as a real
2 if (typeof QueueLoop === 'undefined') {
3     var QueueLoop = 0; //if it's not defined set it and initialize to 0
4 } else {
5     QueueLoop = $(QueueLoop) + 1;
6 }
7
8 1;
```

Help

Input Variables  
List of variables available as input for this node

queue

Custom Variables [F34653]

- QueueLoop
- InQueueMessage1
- InQueueMessage2
- InQueueMessage3

+ Add New Custom Variable

Start Node ID: 2

Evaluate Node ID: 9

Queue Task Node ID: 736

SMS Node ID: 1186

Set Variable Node ID: 1445

Init QueueLoop to 0

Set Values for InQueueMessages

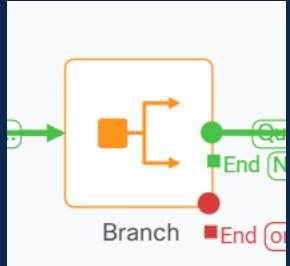
Configure Script Output

Script Output 1

Branch Name return

A large blue callout box highlights the "Init QueueLoop to 0" and "Set Values for InQueueMessages" sections of the configuration interface. A large blue arrow points from the "Set Values for InQueueMessages" section towards the "Custom Variables" list on the right.

# Branch Loop



Ensure PIQ is not 0  
(already talking)

Play message  
based on  
QueueLoop

Branch

Configuration Transition Actions (Optional)

Branch your flow based on conditional logic

▼ Queue Loop 1

Variable: \$(n1909.positionInQueue) Condition: Greater than Value: 0

AND

\$(QueueLoop) Equals 1

AND OR

▼ Queue Loop 2

Variable: \$(n1909.positionInQueue) Condition: Greater than Value: 0

AND

\$(QueueLoop) Equals 2

AND OR

▼ Queue Loop 3

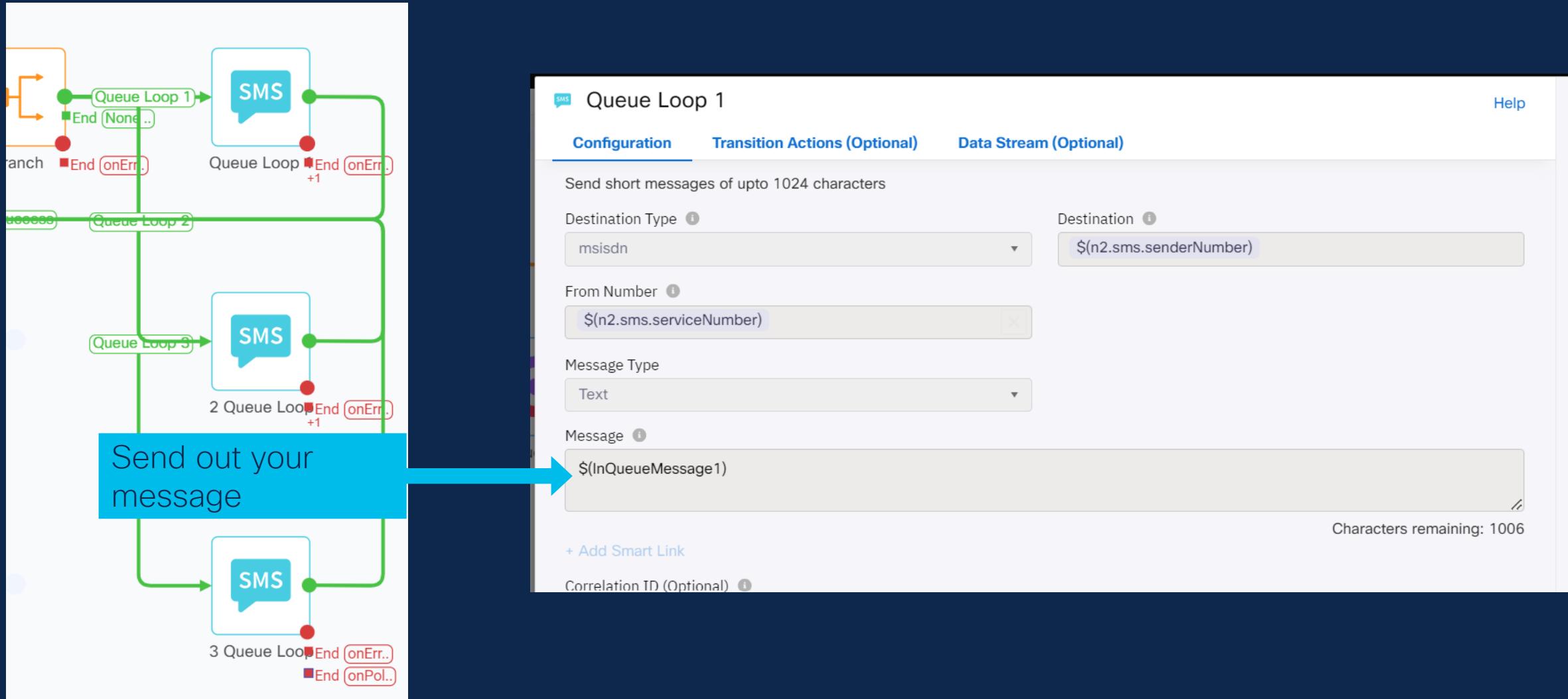
Variable: \$(n1909.positionInQueue) Condition: Greater than Value: 0

AND

\$(QueueLoop) Equals 3

AND OR

# Message Loop



# Digital to Voice



Chandra

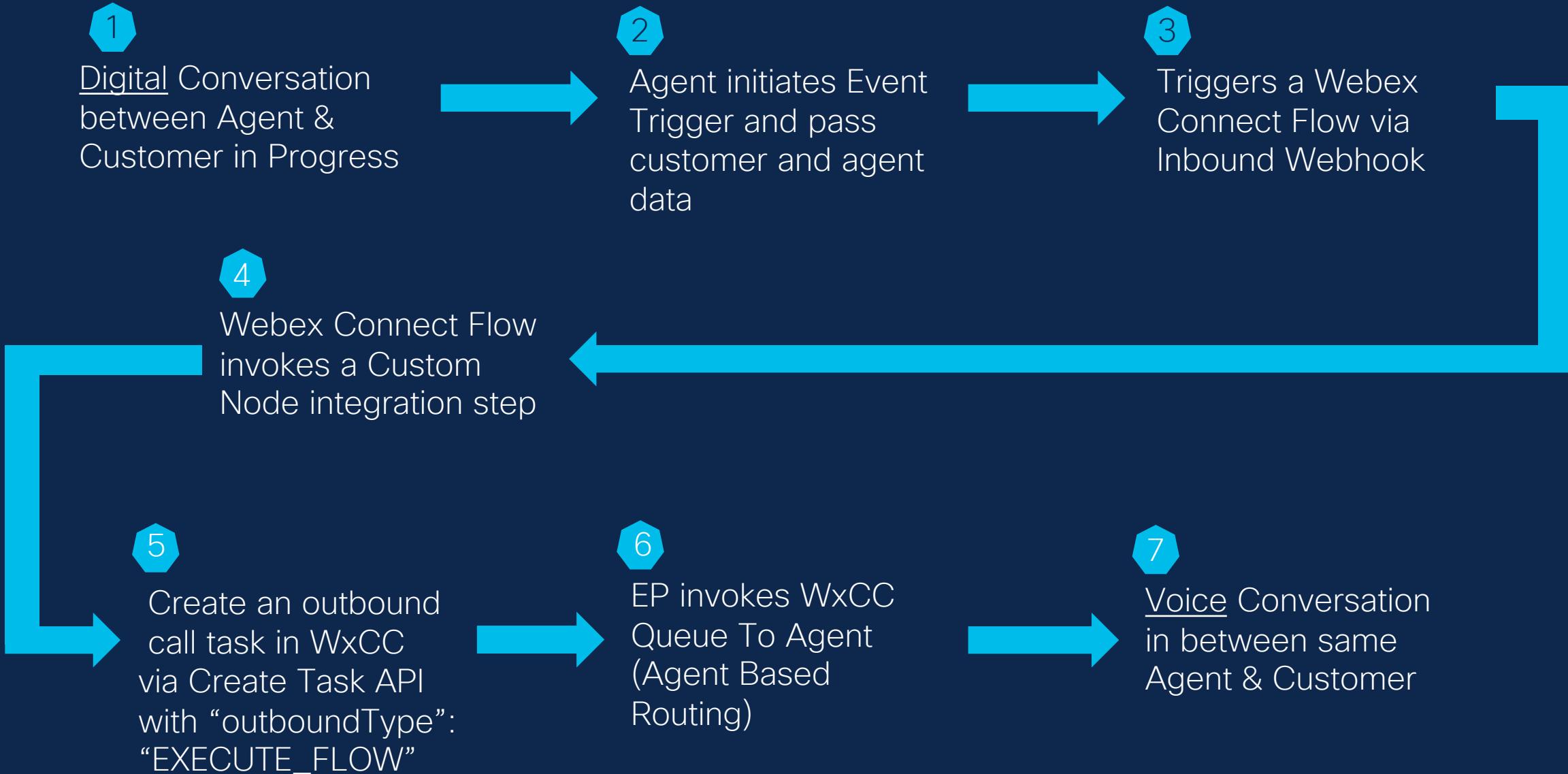
Digital to Voice: Patient is trying to describe their symptoms over chat but is having trouble conveying the severity or complexity of their situation, the agent could offer to switch to voice to get a better understanding of the situation and provide more personalized care.

Transferring to a Department with Voice Lines: A customer is interested in purchasing a product that requires a custom installation. While the sales department can answer general product questions, the installation process requires technical expertise that is only available via phone. By escalating the chat to a voice call with the technical support department, the customer can get the information they need to make an informed purchase, and the sales agent can ensure a seamless customer experience.

## WHY?

- Sometimes it's easier to explain complex issues or convey emotions over voice rather than text.
- You might feel like the chat isn't making progress and switching to voice could help move things forward.
- If you're having trouble typing quickly, voice could be a more efficient way to communicate.
- You might prefer voice as a form of communication in general and want to switch to it even if the chat is going well.
- Escalating to voice could help build a stronger connection with the agent and improve the overall customer experience.
- May need to transfer to a department where digital is not used

## HOW? (Digital to Voice Escalation)



## HOW?

Watch the video and read the content for Basic Authorization:

<https://github.com/CiscoDevNet/webex-contact-center-api-samples/tree/main/app-auth-sample>

Review CREATE TASK API

<https://developer.webex-cx.com/documentation/tasks/v1/create-task>

Test Manually Creating your tasks

EXECUTE\_FLOW – Execute an Entry Point to route to a specific routing script

Set up token key management in WxConnect for WxCC API request

Create a Webex App in Developer portal to obtain Client ID and Client Secret

Set up your Triggers under your Portal / Digital Channels

Create your webhooks to pass data from agent desktop to your WxConnect routing script

# DEMO TIME

<https://app.vidcast.io/share/22bea88e-eea6-4d2f-b94a-39abc5ab86c3>

# Webex Engage: Events and Rules (Digital to Voice Escalation)

Webex Contact Center

Group / Team / Default  
< Default  
Manage users, skills and configure settings within a team.

Events and Rules

Events can be manually fired by agents anytime during a conversation.

Name: Tech Summit Digital to Voice Escalation RTA

Method: POST

URL: https://hooks.au.webexconnect.io/events/A6QL89OHNZ

Expected Response Format: JSON

Wait for Response i    Display Response on Chat Console i

**Payload**  
Choose how you want to define your payload

Key Value Pair    Custom Payload

**Key Value Pair**  
The following parameters will be passed in the API request.

Form-encode Request Body

NAME	PASS THROUGH	PARAM TYPE	MANDATORY	MULTIPLE	VALUES	ACTION	
destination	Request Body	Custom Param	No	No	-		
agentemailid	Request Body	System Param	No	No	agent_emailid		

2  
Agent initiates Event  
Trigger and pass  
customer and agent  
data

# Webex Connect: Webhook and Flow (Digital to Voice Escalation)



3  
Triggers a Webex Connect Flow via Inbound Webhook

**Configure Webhook**

**Configuration**   **Transition Actions (Optional)**

Configure webhook settings to trigger this flow

Select existing webhook    Create new event

Webhook Name: Tech Summit Digital to Voice RTA

We've generated a new endpoint for you to send requests. [Learn more about using webhooks.](#)

Webhook URL: <https://hooks.au.webexconnect.io/events/A6QL890HNZ>

Service key or JWT needs to be passed in request header if this option is selected

Validate signature

Connect can validate JSON signatures generated using SHA256 or SHA512 passed in the header. Invalid request will return a HTTP 400 error

Example Data

Paste Json   Paste XML   Send Via Hook

PROVIDE SAMPLE INPUT:

```
1 [{"destination": "+6596232584", "agentemailid": "anzuser2@email.carehybrid.com"}]
```

SELECT OUTPUT VARIABLES:

PARSSED VARIABLES(2)

destination:+6596232584  
agentemailid:anzuser2@email.carehybrid.com

**LabGuide\_CustomNode**

**Configuration**   **Transition Actions (Optional)**

Method Name: Digital to Voice Escalation RTA

# Webex Connect Custom Node Integration (Digital to Voice Escalation)

Settings    Node UI

**Request Details**

Request Name: Digital to Voice Escalation RTA

Request Timeout (Ms): 10000

Connection Timeout (Ms): 10000

Type: Post    Resource URL: https://api.wxcc-anz1.cisco.com/v1/tasks

**Authorization**

Type: OAuth 2.0    Grant Type: Authorization Code

Consumer ID: Cf62566043a46813e0bceff3807262e01e9e315fe53ecd47f3e6c22fc1c8a5

Consumer Secret: fae931a00ec9d525d8fb76a0e4cc0bbb95b517301712dd6dbe6b494c79c

Call Back URL: https://anzcsam20call.au.webexconnect.io/callback

Authorization URL: https://webexapis.com/v1/authorize

Scope: cjp:config cjp:config\_read cjp:config\_write

Access Token URL: https://webexapis.com/v1/access\_token

The access token has a limited validity

4 Webex Connect Flow invokes a Custom Node integration step

**Validity:** 43199    **Refresh Token URL:** https://webexapis.com/v1/access\_token

**Client Authentication:** Send client credentials in body

**Advance Settings:**

**Auth URL parameters:**

Name	Value
prompt	consent
access_type	offline

**Add New**

**Access Token URL Method:** POST    **Access Token URL Parameter Type:** Body

**Access Token URL parameters:**

Name	Value
grant_type	authorization_code

**Add New**

**Access Token URL Headers:**

**Get Access Token**

**Access Token:** MjMyZTUyMGItYTYwYS00OGZILTgzM2ltYzlyYWRIZDg4MDBIM2VhNzZmMC

**Refresh Token:** MWQyZjhjYTAtYjMwOC00OWExLWIzOTYtYTI2ZDZjMTgyYTEzMzA4MmlwO

**Validity:** 43199

# Webex Connect Custom Node Integration (Digital to Voice Escalation)

Headers

Parameter	Parameter Value Type	Parameter Value
Accept	Static	application/json
Parameter	Parameter Value Type	Parameter Value
Content-Type	Static	application/json

[+ Add New](#)

URL Parameters

[+ Add New](#)

Body

Text (text/plain) ▾

```
1  {
2    "destination": "${param1}",
3    entryPointId": "dd7b5bf8-07bb-429f-8901-5be5d026d5d5",
4    "attributes": {
5      "agentemailid": "${param2}",
6      "rtा": "Voice Escalation"
7    },
8    "outboundType": "EXECUTE_FLOW",
9    "mediatype": "telephony"
10 }
```

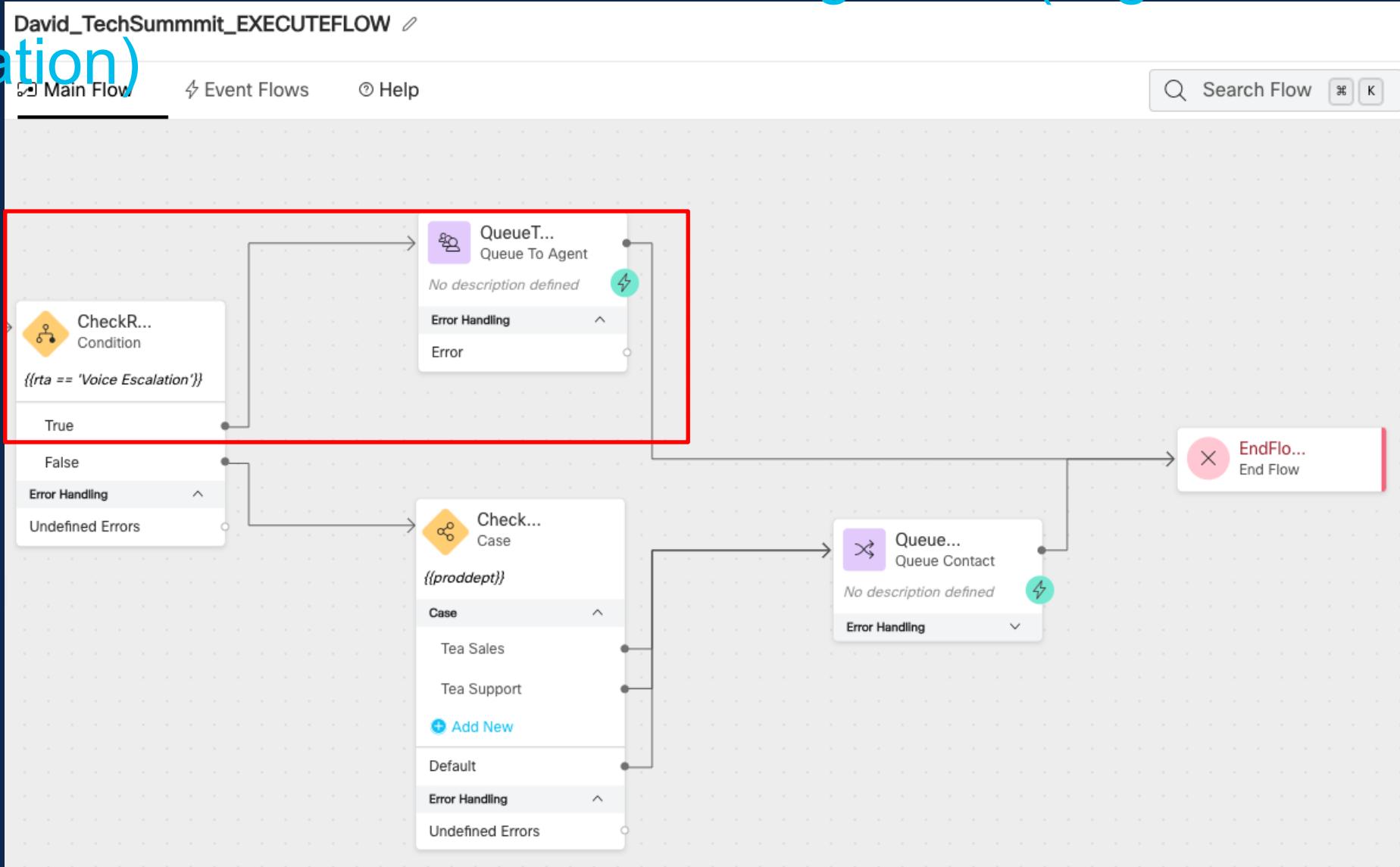
[+ Add Parameters](#)

[Parse](#)

Parameter	Parameter Value Type	Parameter Value
param1	Static	\$(destination)
Parameter	Parameter Value Type	Parameter Value
param2	Static	\$(agentemailid)

5  
Create an outbound call task in WxCC via Create Task API with "outboundType": "EXECUTE\_FLOW"

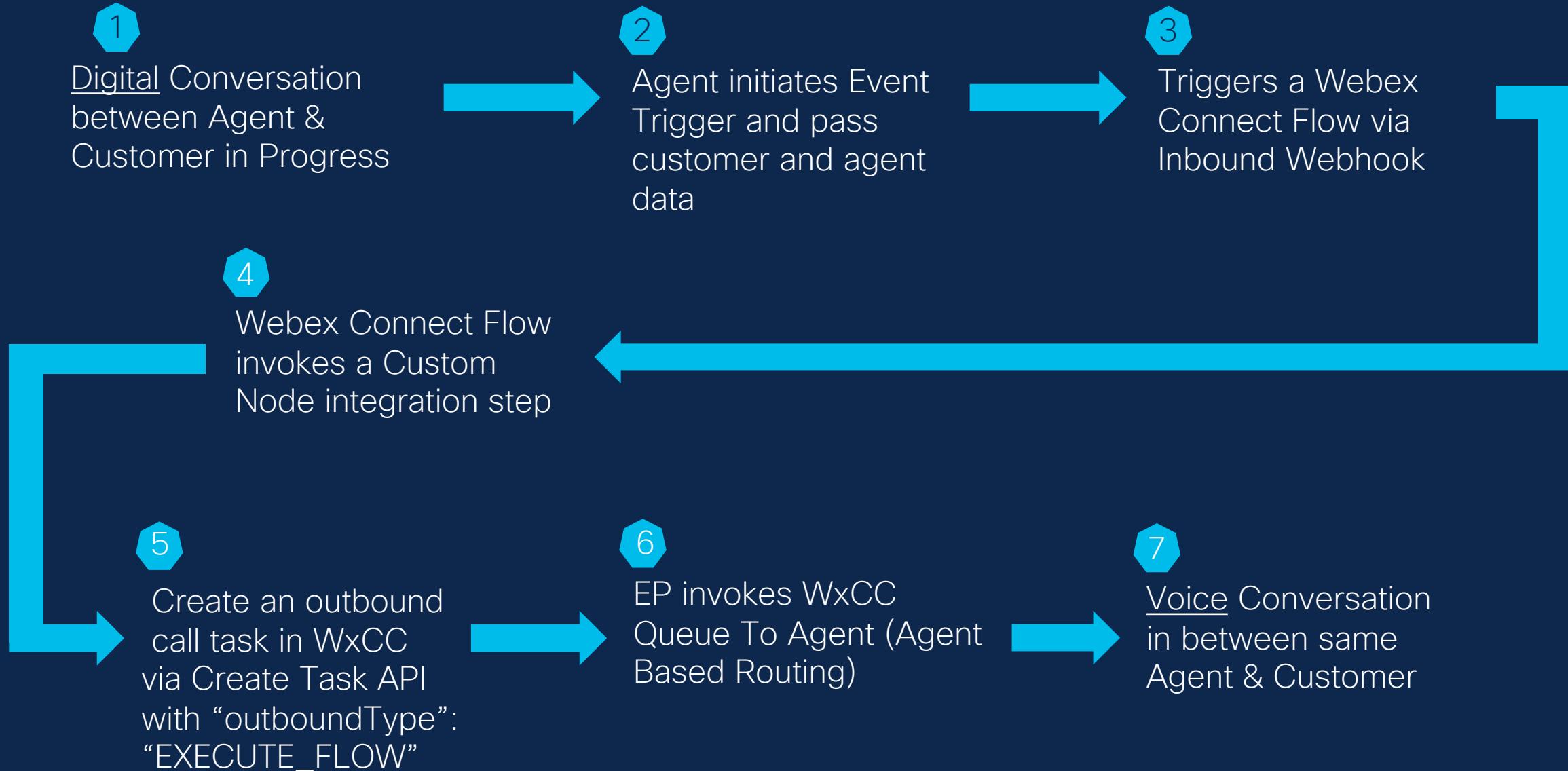
# Webex Connect Custom Node Integration (Digital to Voice Escalation)



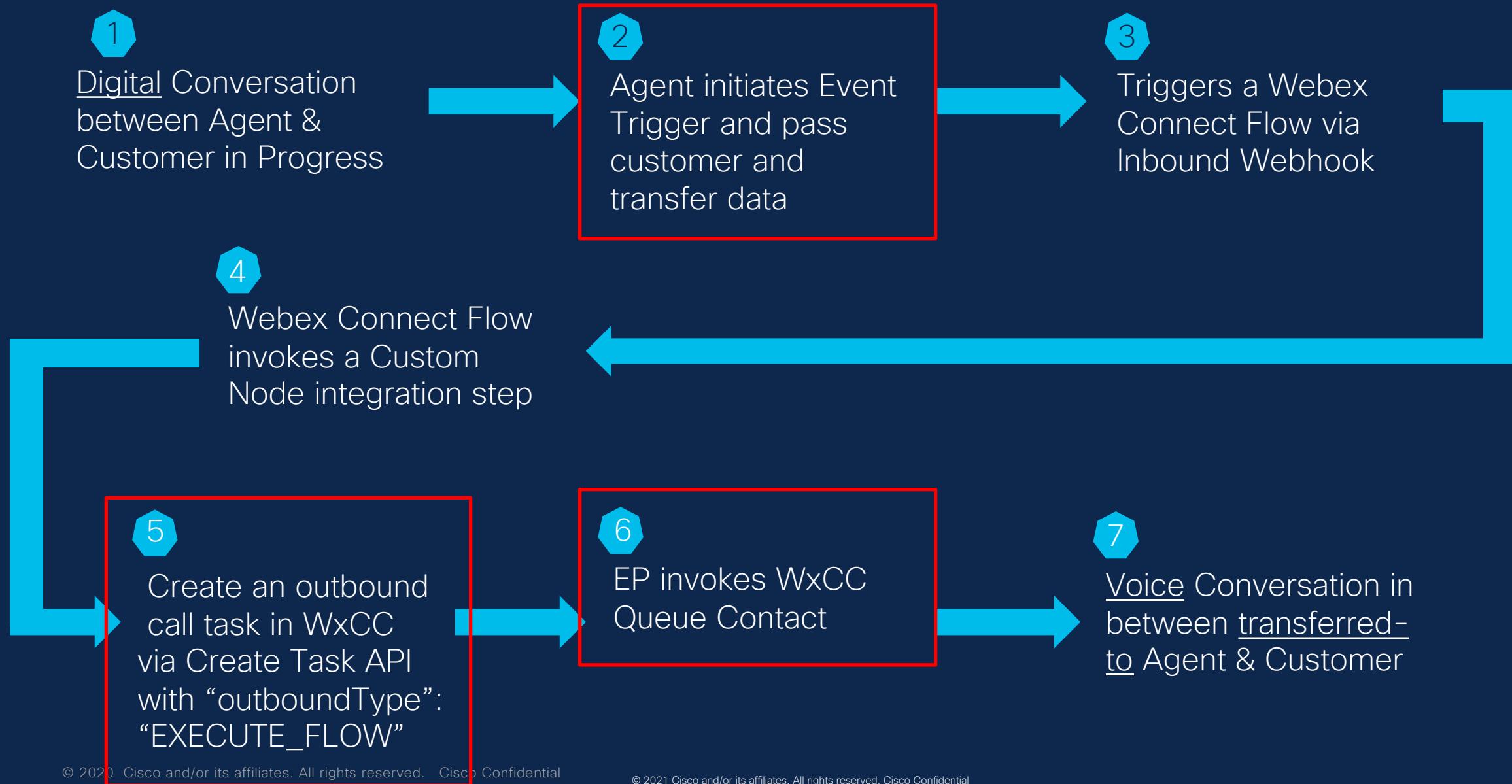
6

EP invokes WxCC  
Queue To Agent  
(Agent Based Routing)

## HOW? (Digital to Voice Escalation)



## HOW? (Digital to Voice Transfer)



# Webex Engage: Events and Rules (Digital to Voice Transfer)

Webex Contact Center

Group / Team / Default < Default

Manage users, skills and configure settings within a team.

Users Events and Rules

Tech Summit Digital to Voice Other Dept

Method: POST

URL: https://hooks.au.webexconnect.io/events/JKGO4098OD

Expected Response Format: JSON

Wait for Response 1  Display Response on Chat Console 1

**Payload**

Choose how you want to define your payload

Key Value Pair  Custom Payload

**Key Value Pair**

The following parameters will be passed in the API request.

Form-encode Request Body

NAME	PASS THROUGH	PARAM TYPE	MANDATORY	MULTIPLE	VALUES	ACTIONS	
destination	Request Body	Custom Param	No	No	-		
proddept	Request Body	Custom Param	No	No	-		
transreason	Request Body	Custom Param	No	No	-		

2 Agent initiates Event Trigger and pass customer and agent data

# Webex Connect Custom Node Integration (Digital to Voice Transfer)

Body

Text (text/plain) ▾

```
1 {  
2   "destination": "${param1}",  
3   "entryPointId": "dd7b5bf8-07bb-429f-8901-5be5d026d5d5",  
4   "attributes": {  
5     "proddept": "${param2}",  
6     "transreason": "${param3}"  
7   },  
8   "outboundType": "EXECUTE_FLOW",  
9   "mediaType": "telephony"  
10 }
```

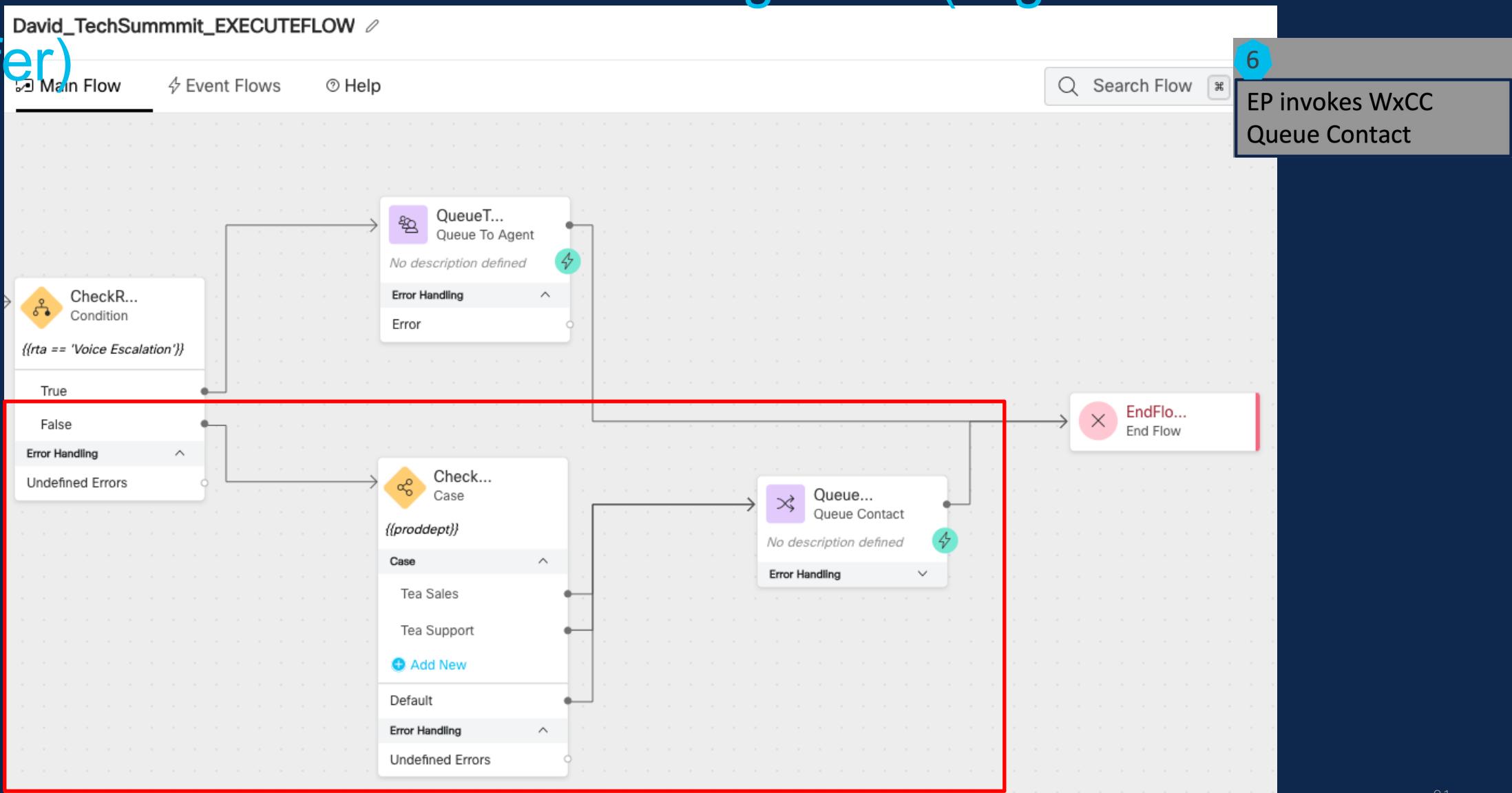
5 Create an outbound call task in WxCC via Create Task API with "outboundType": "EXECUTE\_FLOW"

No 'agentemailid' and 'rta' variable

Add Parameters Parse

Parameter	Parameter Value Type	Parameter Value
param1	Static	\$(destination)
Parameter	Parameter Value Type	Parameter Value
param2	Static	\$(proddept)
Parameter	Parameter Value Type	Parameter Value
param3	Static	\$(transreason)

# Webex Connect Custom Node Integration (Digital to Voice Transfer)



# Webex Connect Custom Node Integration – Digital To Voice

**Request Details**

Request Name: Callback

Request Timeout (Ms): 10000 Connection Timeout (Ms): 10000

Type: Post Resource URL: https://api.wxcc-us1.cisco.com/v1/tasks Parse Variables

**Authorization**

Type: OAuth 2.0 Grant Type: Authorization Code

Consumer ID: C4d9ab5e6a5c34814f10db79b67120ad9c7f0c2c3dae88d72f8s Consumer Secret: 52ac69323680f21b4aa6993ded39ecbc28e6b0b9f87e10f81bf8i

Call Back URL: https://tsacc.us.webexconnect.io/callback

Authorization URL: https://webexapis.com/v1/authorize

Scope: cjp:config cjp:config\_read cjp:config\_write

Access Token URL: https://webexapis.com/v1/access\_token

The access token has a limited validity

Validity: 43199 Refresh Token URL: https://webexapis.com/v1/access\_token

Client Authentication: Send client credentials in body

**Advance Settings**

Auth URL parameters: + Add New

Access Token URL Method: POST Access Token URL Parameter Type: Body

Access Token URL parameters:

Name: grant_type	Value: authorization_code
------------------	---------------------------

**Access Token URL Headers**

+ Add New Get Access Token

Access Token: NzBhyWZlNjctNzNmZC00MzBiLTkyNDiTYTlZWQxNTg2YTMzMm Refresh Token: YzRmZjlmlNTQtZGU1YS00NzIzLThkODItMWI4ZmRlNGYyMDZjMz

Validity: 43199

**Security Configuration**

**Throttling**

On enabling, an additional outcome throttleLimitReached\* will be available with this custom node.

**Rate Limit**

Limit total number of request per second. Beyond this limit, node will return 429 with description "Throttle rate limit reached - Too Many requests" status

Rate: 1 Requests per second

**Concurrency Limit**

Limit maximum number of parallel custom node executions. Beyond this limit, node will return 430 with description "Throttle concurrency limit reached - Too Many requests" status

Maximum Limit: 20 Parallel executions

**Volume Limit**

Limit maximum number of custom node executions against a time period. Beyond this limit, node will return 431 with description "Throttle volume limit reached - Too Many requests" status. Atleast one limit has to be configured.

Maximum Limit: 2 Requests per minut

**Headers**

Parameter	Parameter Value Type	Parameter Value
Content-Type	Static	application/json
Parameter	Parameter Value Type	Parameter Value

Accept: Static application/json

+ Add New

**URL Parameters**

+ Add New

**Body**

Text (text/plain)

```

1 [
2   {
3     "destination": "${MobileNumber}",
4     "entryPoint": "${Entry_PointID}",
5     "attributes": "${desktopFields}",
6     "outboundType": "CALLBACK",
7     "mediatype": "telephony",
8     "callback": {
9       "callbackorigin": "${Origin_web_chat_or_ivr}",
10      "callbackType": "immediate"
11    }
12  }
13 ]

```

**Response**

Import from sample JSON (application)

A node event signifies outcome of a node. Create a node event by specifying the response path. One or more responses can be mapped to an event.

**Configure Node Events**

Node Event	Body	Response Path	Condition	Value	Node Edge
Success	HTTP Status		starts with	2	Success
ClientError	HTTP Status		starts with	4	Error

+ Add New

A response object can be used in the node by saving it as a node variable.

Set data to be returned in a flow session

Parameter Name	Body	Response Path
request_body	Body	\$

+ Add New

## Webex Connect Custom Node Integration – Digital To Voice

Resource: <https://api.wxcc-us1.cisco.com/v1/tasks>

Authorization: <https://webexapis.com/v1/authorize>

Scope: cjp:config cjp:config\_read cjp:config\_write

Access Token: [https://webexapis.com/v1/access\\_token](https://webexapis.com/v1/access_token)

Validity: 43199

Refresh Token: [https://webexapis.com/v1/access\\_token](https://webexapis.com/v1/access_token)

Access Toke URL Parameters: Name: grant\_type      Value: authorization\_code

Body:

```
{  
    "destination": "$(MobileNumber)",  
    "entryPointId": "$(Outdial_Entry_PointID)",  
    "attributes": "{$(DesktopFields)}",  
    "outboundType": "CALLBACK",  
    "mediaType": "telephony",  
    "callback": {  
        "callbackOrigin": "$(Origin_web_chat_or_ivr)",  
        "callbackType": "immediate"  
    }  
}
```

## Webex Connect Custom Node Integration – Digital Escalation

Resource: <https://api.wxcc-us1.cisco.com/v1/tasks>

Authorization: <https://webexapis.com/v1/authorize>

Scope: cjp:config cjp:config\_read cjp:config\_write

Access Token: [https://webexapis.com/v1/access\\_token](https://webexapis.com/v1/access_token)

Validity: 43199

Refresh Token: [https://webexapis.com/v1/access\\_token](https://webexapis.com/v1/access_token)

Access Toke URL Parameters: Name: grant\_type      Value: authorization\_code

Body:

```
{ "destination": "$(MobileNumber)",  
  "entryPointId": "$(Outdial_Entry_PointID)",  
  "attributes": {"AgentLoginID": "$(AgentLoginID)", "OrginalChannel": "$(OrginalChannel)", $(DesktopFields)},  
  "outboundType": "EXECUTE_FLOW",  
  "mediaType": "telephony"  
}
```

# Webex Connect Flow - Digital Escalation

Portal -> New Digital Channels Cross Launch

Groups ->

Default ->

Default ->

Events and Rules ->

Add New Event

{

```
"contactnumber": "Number To Call Back On",
"CustomerMobileNo": "12263762555 - cell number",
"AssetID": "Asset defined in Connect, i.e SMS From Number",
"Channel": "Facebook, SMS, LiveChat, WhatsApp, Email",
"FacebookPSID": "Facebook PageID",
"AppID": "Livechat appID",
"CustomerID": "Livechat CustomerID",
"BrowserFingerprint": "Livechat BrowserID",
"ThreadID": "Livechat Thread",
"LiveChatID": "Livechat ID",
"AgentID": "Agent login ID",
"AgentEmailID": "Agent Email ID",
"AgentName": "Agents Name",
"TriggerName": "Name you called your trigger"
```

Name  
Escalate To Voice

Method  
POST

URL  
<https://hooks.us.webexconnect.io/events/SWTLLB1ZEC>

Expected Response Format  
JSON

Wait for Response ⓘ  Display Response on Chat Console ⓘ

**Payload**  
Choose how you want to define your payload

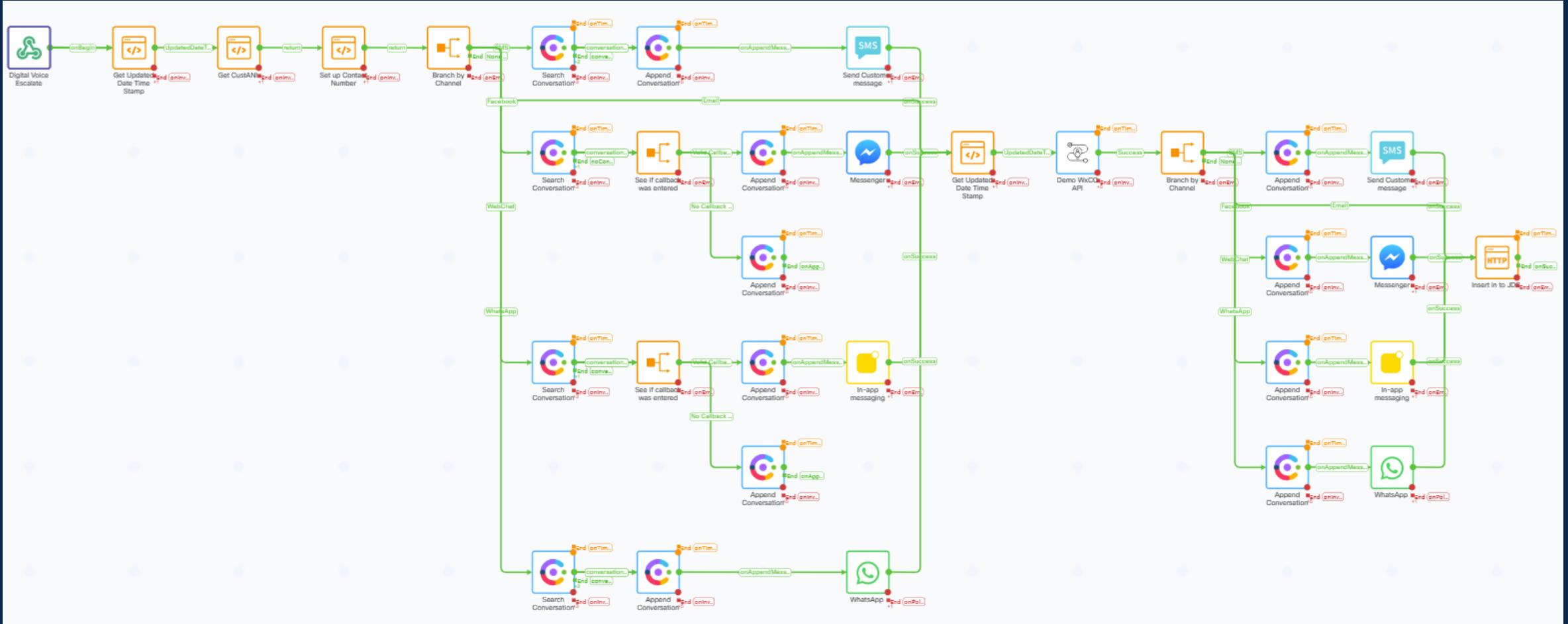
Key Value Pair  Custom Payload

**Key Value Pair**  
The following parameters will be passed in the API request.

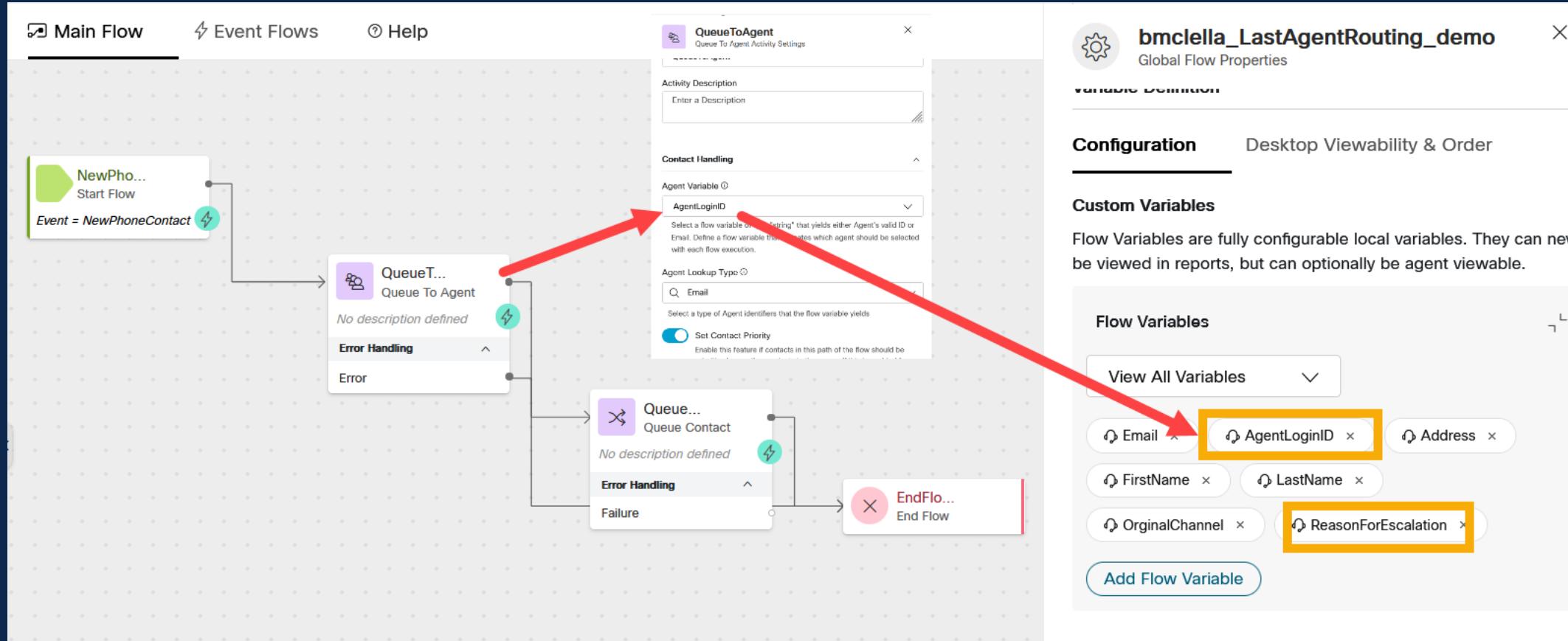
Form-encode Request Body

NAME	PASS THROUGH	PARAM TYPE	MANDATORY	MULTIPLE	VALUES
Content-Type	Header	System Param	No	No	application/json
key	Header	System Param	No	No	33566a0c-d32b-11ed-8
contactnumber	Request Body	Custom Param	No	No	-
CustomerMobileNo	Request Body	System Param	No	No	customer mobile no
AssetID	Request Body	System Param	No	No	assetid
Channel	Request Body	System Param	No	No	channel
FacebookPSID	Request Body	System Param	No	No	facebook.psid
AppID	Request Body	System Param	No	No	appid
CustomerID	Request Body	System Param	No	No	customer id
BrowserFingerprint	Request Body	System Param	No	No	browser fingerprint
ThreadID	Request Body	System Param	No	No	thread id
LiveChatID	Request Body	System Param	No	No	livechat id
AgentID	Request Body	System Param	No	No	agent id
AgentEmailID	Request Body	System Param	No	No	agent_emailid
AgentName	Request Body	System Param	No	No	agent name
TriggerName	Request Body	System Param	No	No	EscalateVoice

# Webex Connect Flow – Digital Escalation



# Webex Flow Builder – Digital Escalation



# Digital To Voice Department

Portal -> New Digital Channels Cross Launch

Groups ->

Default ->

Default ->

Events and Rules ->

Add New Event

{

```
"agentmessage": "Message to send on transfer",
"contactnumber": "number to call back on",
"department": "sales or service",
"CustomerMobileNo": "12263762555 - cell number",
"AssetID": "Asset defined in Connect, i.e SMS From Number",
"Channel": "Facebook, SMS, LiveChat, WhatsApp, Email, ",
"FacebookPSID": "Facebook PageID",
"ApplD": "Livechat applD",
"CustomerID": "Livechat CustomerID",
"BrowserFingerprint": "Livechat BrowserID",
"ThreadID": "Livechat Thread",
"LiveChatID": "Livechat ID",
"AgentID": "Agent login ID",
"AgentEmailID": "Agent Email ID",
"AgentName": "Agents Name",
"TriggerName": "Name you called your trigger"
```

Name

Method

URL

Expected Response Format

Wait for Response i  Display Response on Chat Console i

**Payload**  
Choose how you want to define your payload

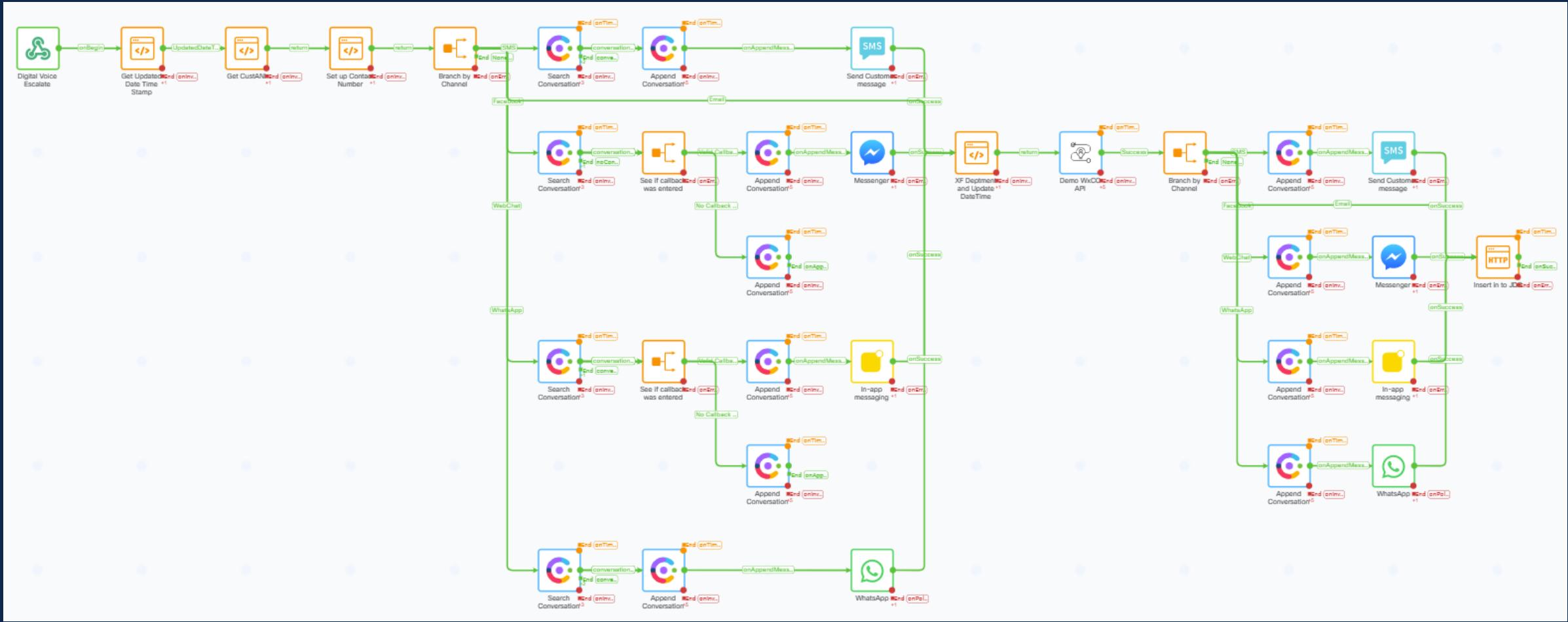
Key Value Pair  Custom Payload

**Key Value Pair**  
The following parameters will be passed in the API request.

Form-encode Request Body

NAME	PASS THROUGH	PARAM TYPE	MANDATORY	MULTIPLE	VALUES
Content-Type	Header	System Param	No	No	application/json
key	Header	System Param	No	No	33566a0c-d32b-11ed-89a4-06b100
agentmessage	Request Body	Custom Param	No	No	-
contactnumber	Request Body	Custom Param	No	No	-
department	Request Body	Custom Param	No	Yes	sales, service
CustomerMobileNo	Request Body	System Param	No	No	customer mobile no
AssetID	Request Body	System Param	No	No	assetid
Channel	Request Body	System Param	No	No	channel
FacebookPSID	Request Body	System Param	No	No	facebook.psid
ApplD	Request Body	System Param	No	No	appid
CustomerID	Request Body	System Param	No	No	customer id
BrowserFingerprint	Request Body	System Param	No	No	browser fingerprint
ThreadID	Request Body	System Param	No	No	thread id
LiveChatID	Request Body	System Param	No	No	livechat id
AgentID	Request Body	System Param	No	No	agent id
AgentEmailID	Request Body	System Param	No	No	agent_emailid
AgentName	Request Body	System Param	No	No	agent name
TriggerName	Request Body	System Param	No	No	DigitalToVoiceXFR

# Webex Connect Flow – Digital To Voice



# Write notes to store in customers journey



Tarek

## Adding Notes to a customer journey

Adding notes and sentiment to a customer journey data storage can benefit sales by providing reps with a complete view of the customer journey, enabling personalized and empathetic service, and increasing the chances of customer retention. By accessing relevant information, sales reps can address issues and provide better experiences, ultimately driving sales and customer loyalty.

# WHY?

- Notes and sentiment provide a comprehensive record of each interaction with the customer
- Improved customer experience
- Better understanding of customer needs and preferences leads to more personalized and empathetic service
- Identification of pain points
- Reviewing customer journey data helps to identify areas where customers are having difficulty
- Facilitation of customer retention
- Understanding the customer journey enables proactive steps to address needs and ensure satisfaction

# Setting up the trigger to execute the flow

Portal -> New Digital Channels Cross Launch

Groups ->

Default ->

Default ->

Events and Rules ->

Add New Event

{

```
"emoticon": "Happy or Sad or Neutral",
"agentmessage": "Agent Notes Comment",
"CustomerMobileNo": "12263762555 - cell number",
"AssetID": "Asset defined in Connect, i.e SMS From Number",
"Channel": "Facebook, SMS, LiveChat, WhatsApp, Email, ",
"FacebookPSID": "Facebook PageID",
"AppID": "Livechat appID",
"CustomerID": "Livechat CustomerID",
"BrowserFingerprint": "Livechat BrowserID",
"ThreadID": "Livechat Thread",
"LiveChatID": "Livechat ID",
"AgentID": "Agent login ID",
"AgentEmailID": "Agent Email ID",
"AgentName": "Agents Name",
"TriggerName": "Name you called your trigger"
```

Webex Contact Center

Group / Team / Default < Default

Manage users, skills and configure settings within a team.

Users Events and Rules

< Configure Events

Events can be manually fired by agents anytime during a conversation.

Name: JDS Write Notes

Method: POST

URL: https://hooks.us.webexconnect.io/events/47FTYUA9RX

Expected Response Format: JSON

Wait for Response ?  Display Response on Chat Console ?

Form-encode Request Body

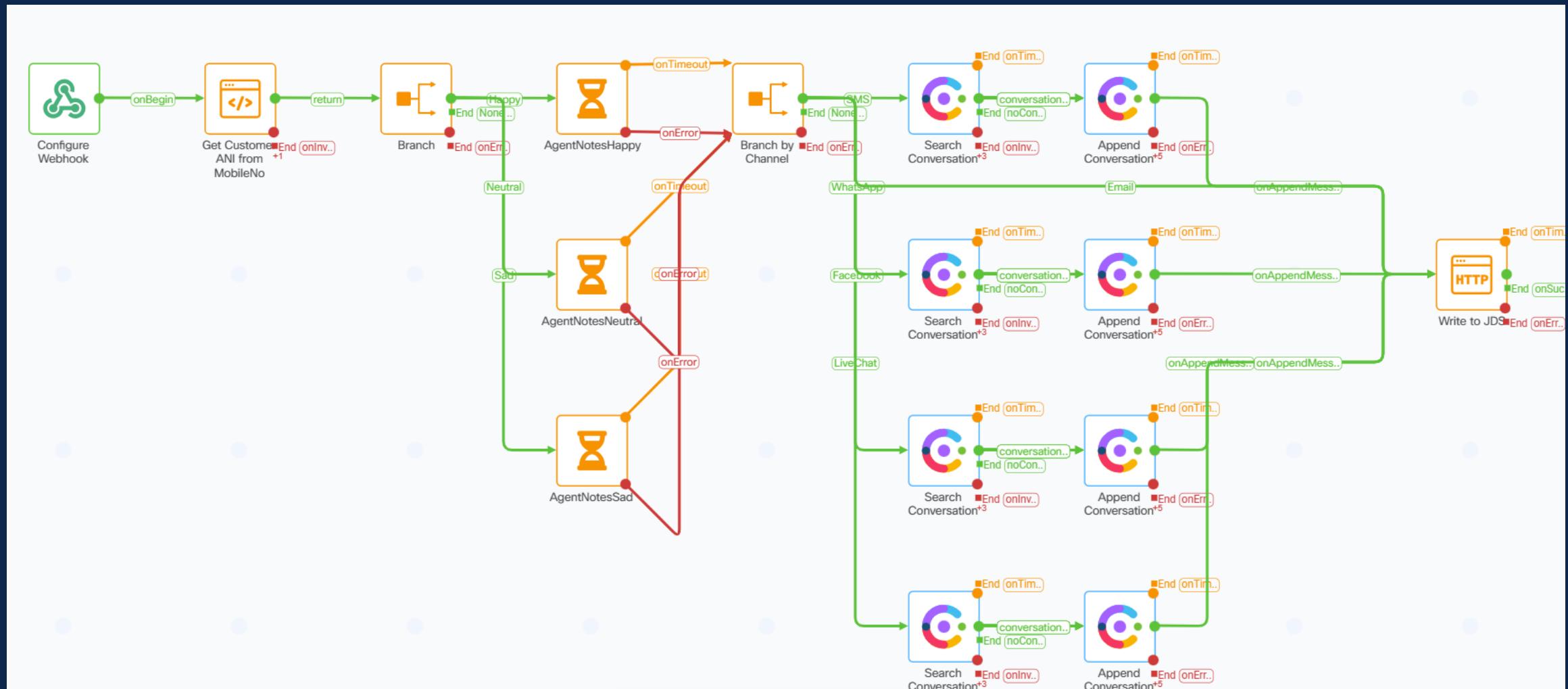
NAME	PASS THROUGH	PARAM TYPE	MANDATORY	MULTIPLE	VALUES	ACTIONS
Content-Type	Header	System Param	No	No	application/json	
key	Header	System Param	No	No	33566a0c-d32b-11ed-	
emoticon	Request Body	Custom Param	No	Yes	happy, sad, neutral	
agentmessage	Request Body	Custom Param	No	No	-	
CustomerMobileNo	Request Body	System Param	No	No	customer mobile no	
AssetID	Request Body	System Param	No	No	assetid	
Channel	Request Body	System Param	No	No	channel	
FacebookPSID	Request Body	System Param	No	No	facebook.psid	
AppID	Request Body	System Param	No	No	appid	
CustomerID	Request Body	System Param	No	No	customer id	
BrowserFingerprint	Request Body	System Param	No	No	browser fingerprint	
ThreadID	Request Body	System Param	No	No	thread id	
LiveChatID	Request Body	System Param	No	No	livechat id	
AgentID	Request Body	System Param	No	No	agent id	
AgentEmailID	Request Body	System Param	No	No	agent_emailid	
AgentName	Request Body	System Param	No	No	agent name	
TriggerName	Request Body	System Param	No	No	WriteNotes	

# Custom Icons

```
"AgentNotesHappy": {  
    "name": "icon-emoticons_16",  
    "color": "cobalt"  
},  
"AgentNotesSad": {  
    "name": "icon-emoticon-sad_24",  
    "color": "red"  
},  
"AgentNotesNeutral": {  
    "name": "icon-emoticon-passive_24",  
    "color": "yellow"  
}
```

<https://momentum.design/icons>

# Webex Connect Flow



Escalate from Digital to Video



Tarek

# Escalate from Digital to Video

- Several customer have enquired whether customers can speak to agents using Video rather than traditional voice.
- When in a digital conversation, customers are offered choice of chat, standard voice or video when escalating to an agent.
- If they choose video, still route call to agent but also send video link to customer via SMS.
- Agent has video embedded in Desktop
- Customer clicks on link to start video in browser on their phone.

# Video Call- Steps Required

Brainstorm - What Pre-Requisites/Steps/Nodes are required to setup this flow?

## Challenges

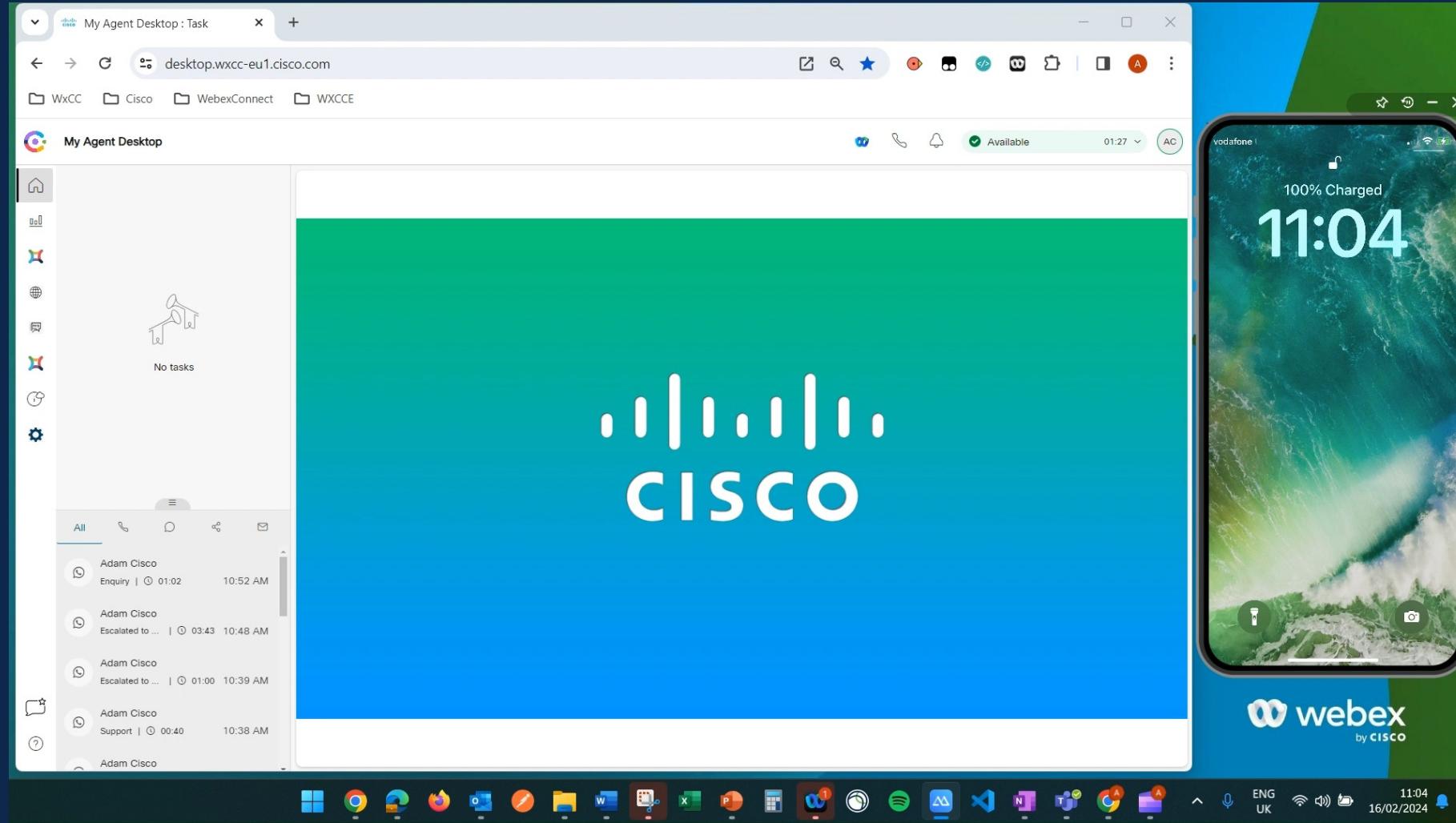
- How are we going to do video?
- How to present video to agent and customer?

## Flow Steps

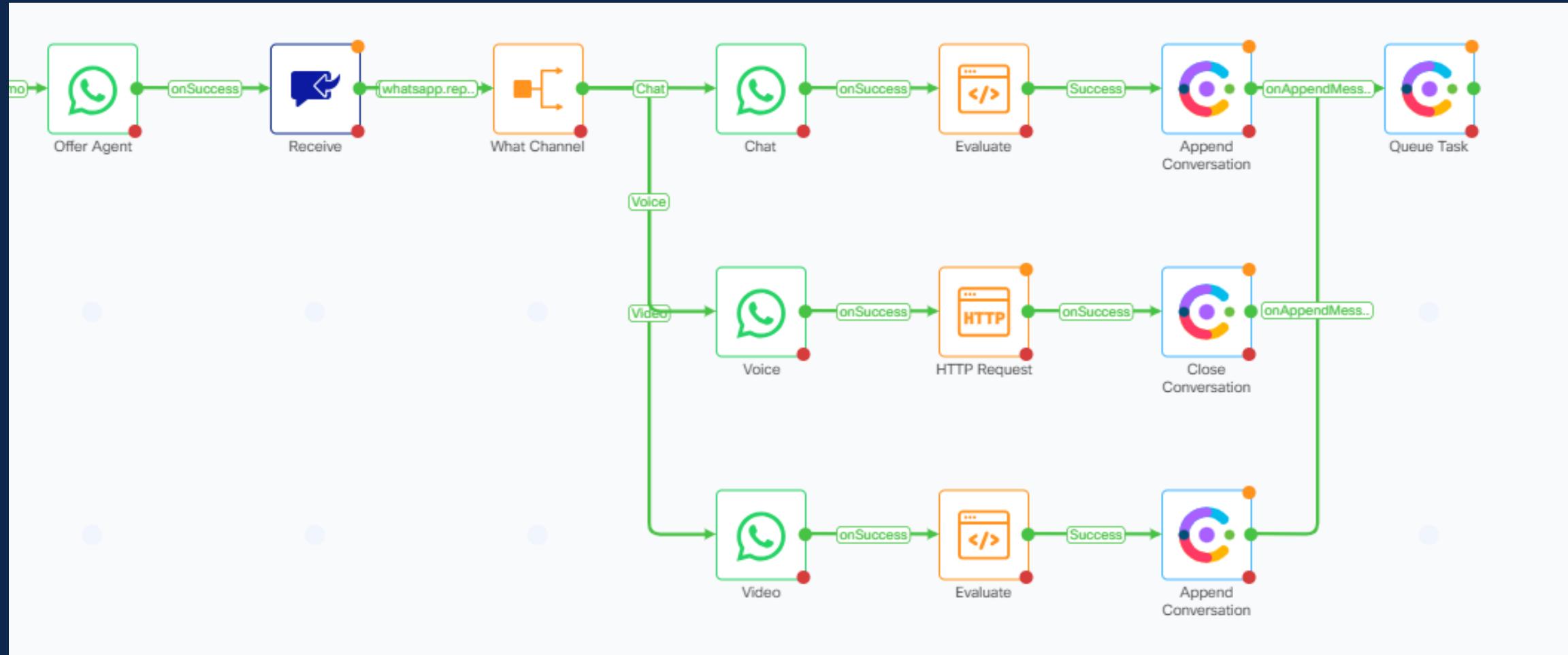
- Find a suitable place in the digital flow to offer video
- If chosen, use Instant Connect APIs to create host and guest links
- Store host link for the agent to use in the Screen Pop widget
- Send guest link to the customer
- Route the digital conversation
- Agent starts video call and waits for customer to join.

- Several customer have enquired whether customers can speak to agents using Video rather than traditional voice.
- When in a digital conversation, customers are offered choice of chat, standard voice or video when escalating to an agent.
- If they choose video, still route call to agent but also send video link to customer via SMS.
- Agent has video embedded in Desktop
- Customer clicks on link to start video in browser on their phone

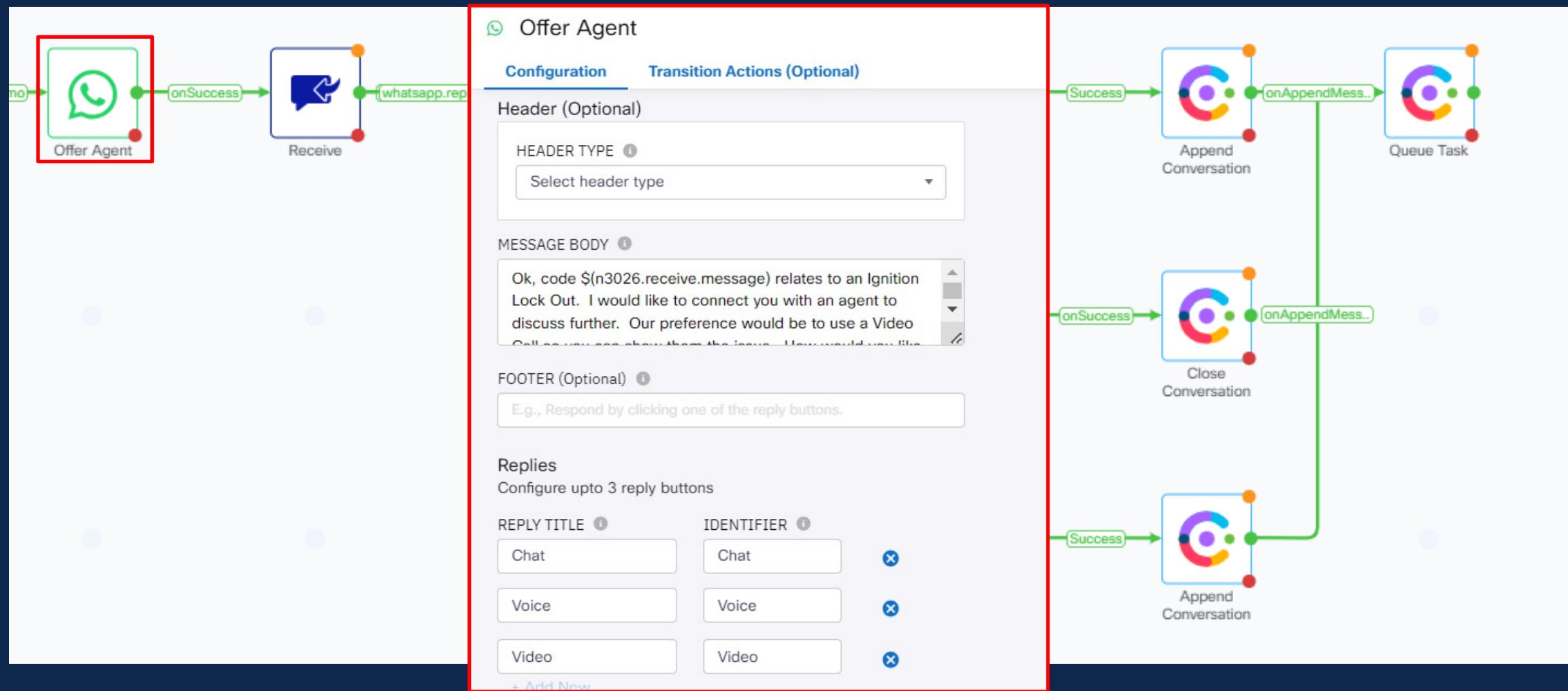
# Escalate to Video - Demonstration



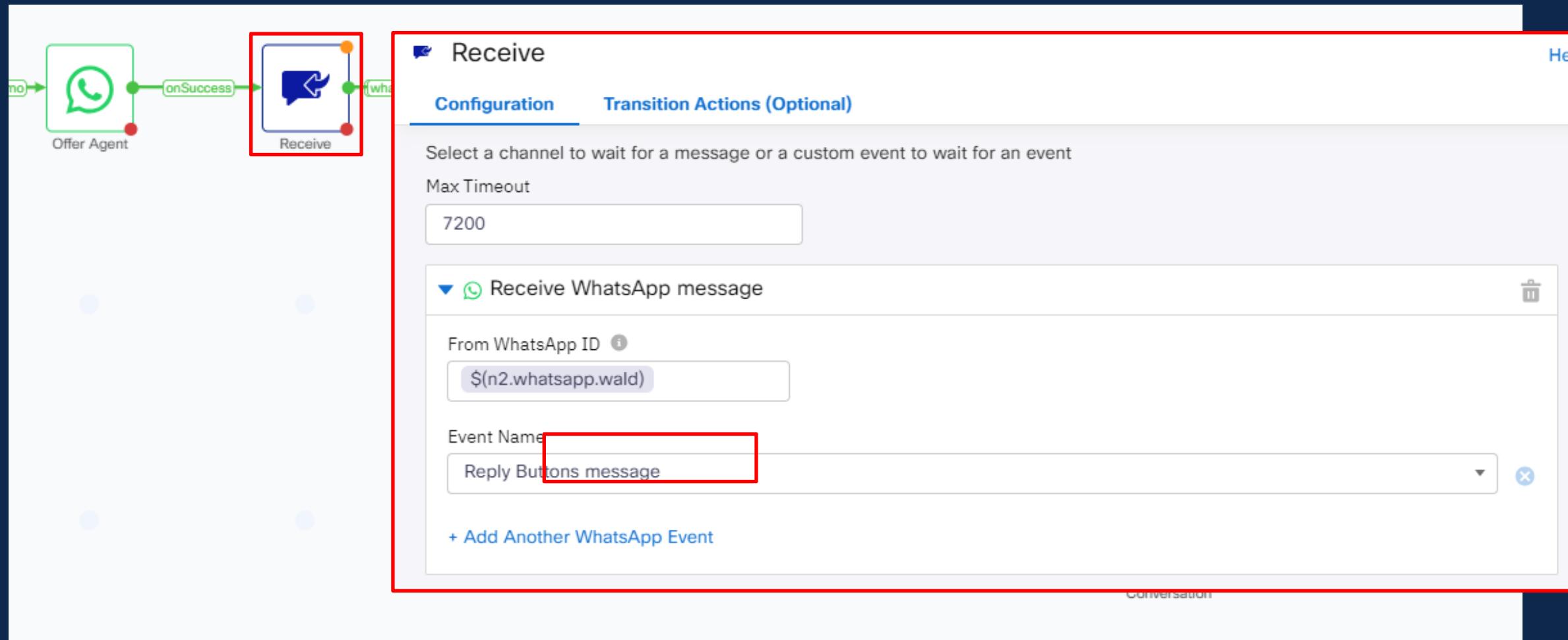
# Escalate to Video- Main Flow



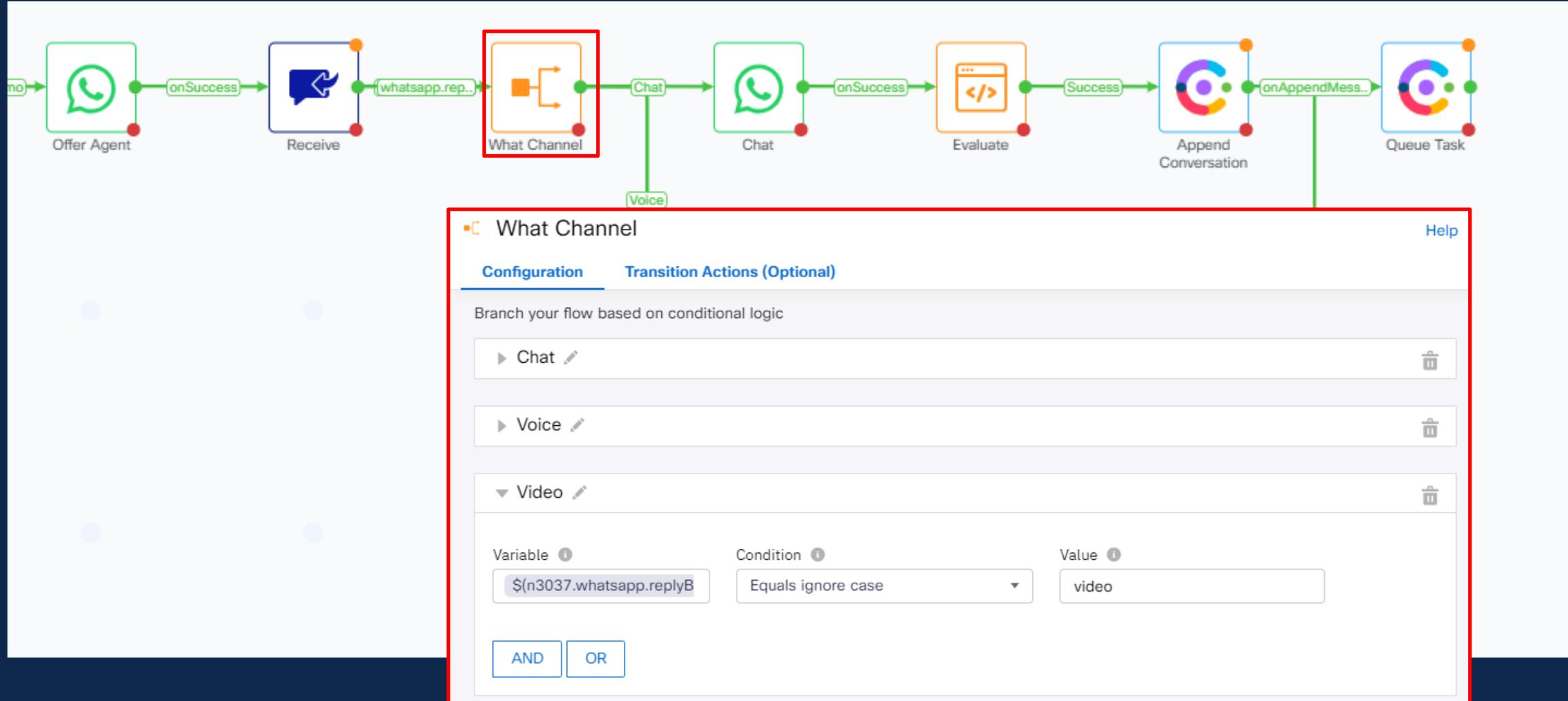
# Escalate to Video- Main Flow



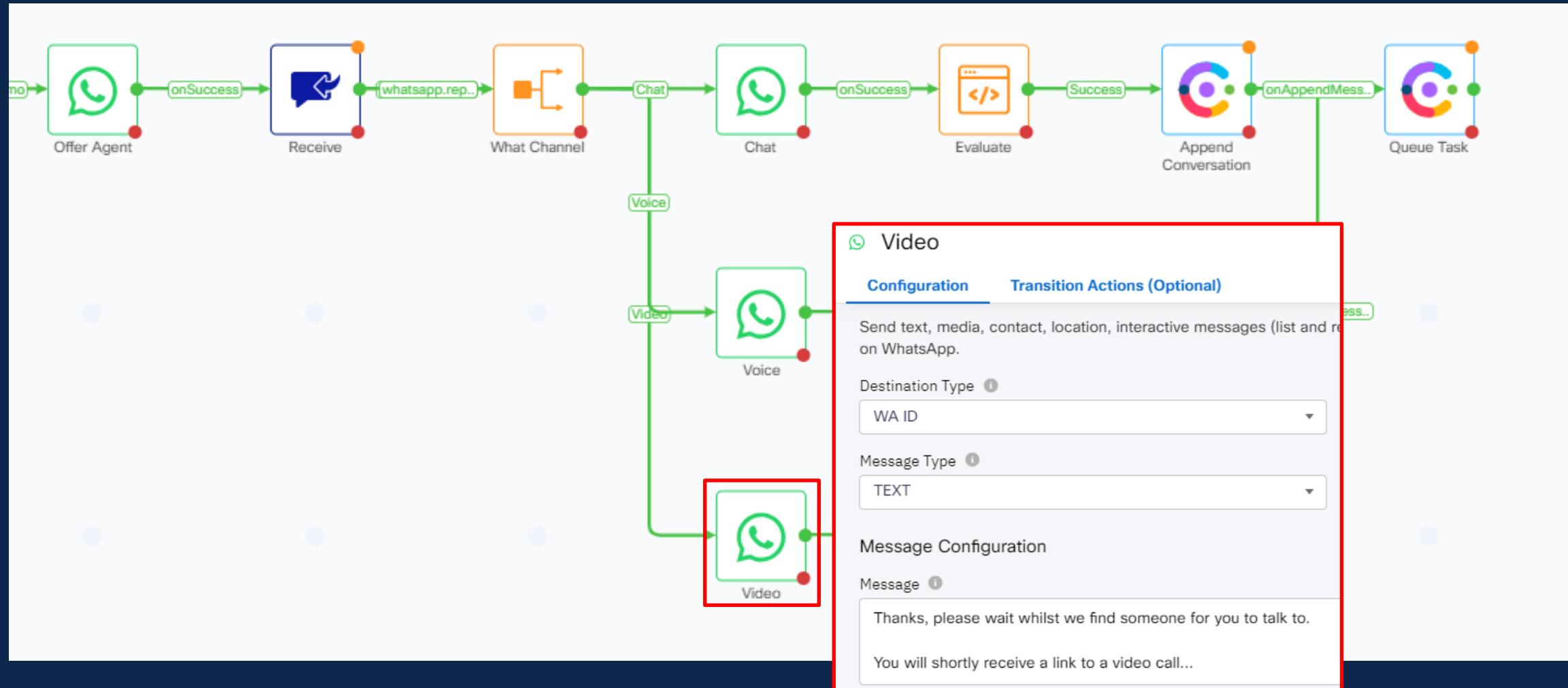
# Escalate to Video- Main Flow



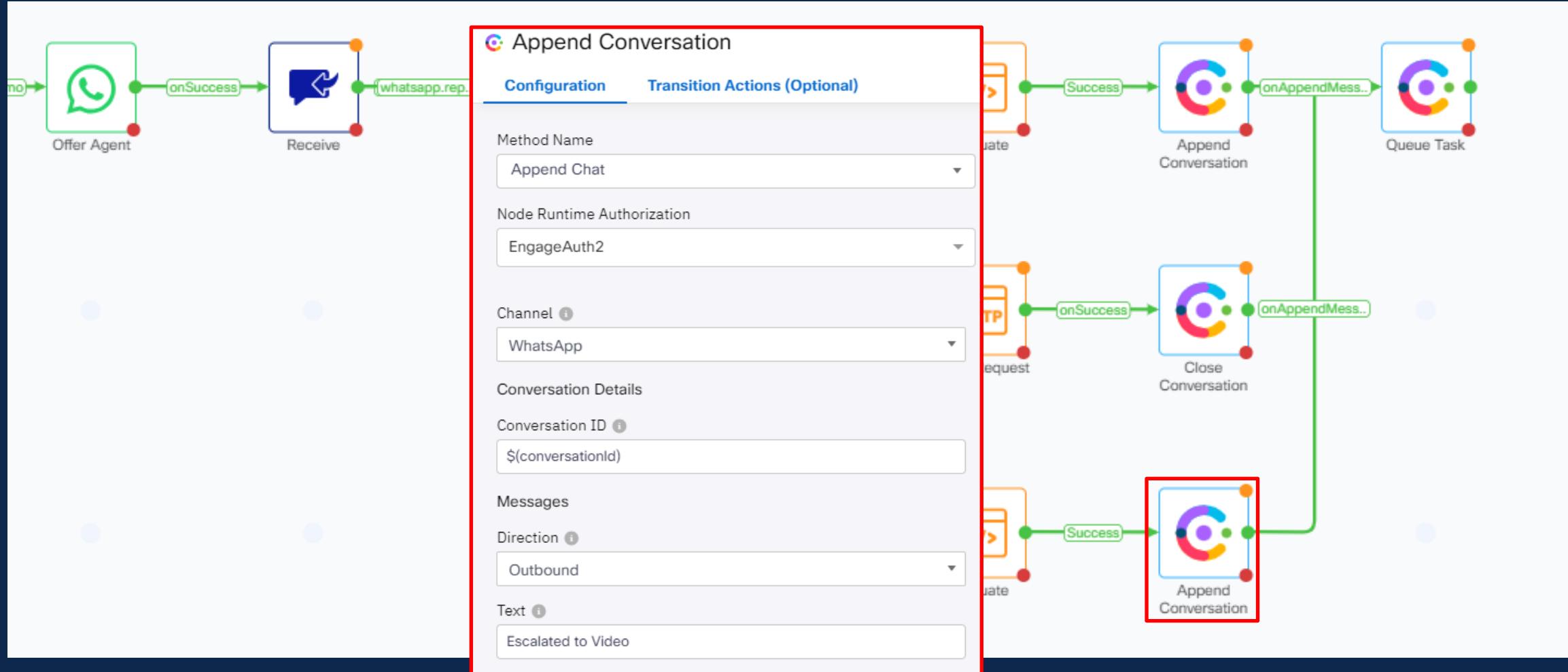
# Escalate to Video- Main Flow



# Escalate to Video- Main Flow



# Escalate to Video- Main Flow



# Escalate to Video- Main Flow

Queue Task

**Configuration**    **Transition Actions (Optional)**

Method Name: Queue task

Node Runtime Authorization: WxCCAuth2

Task ID: \${n1561.Task ID}

Conversation ID: \${conversationId}

Media Type: Social

MEDIA CHANNEL: WhatsApp

**Queue details**

Queue Name: Q\_Social\_Adam

Queue routing Type: Longest available agent

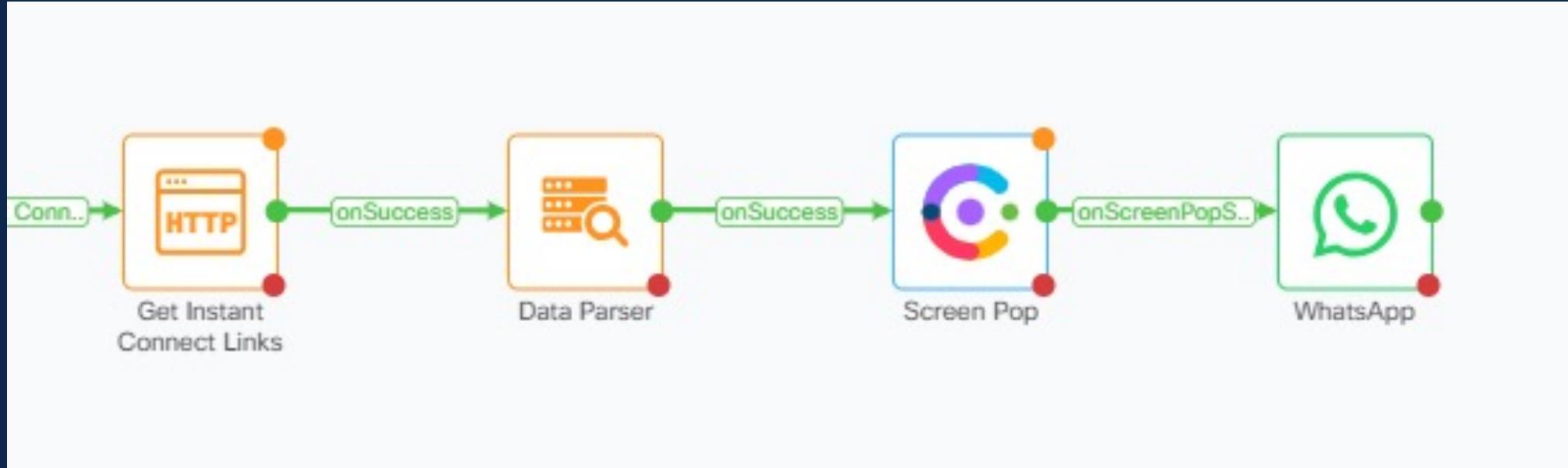
The diagram shows a 'Queue Task' node with a red border. A green arrow labeled 'onAppendMessage' points from the bottom of the node to a green line that extends downwards and then turns right. This line ends at a blue circle, which is part of a larger green line that loops back to the left, forming a cycle. The entire loop is enclosed in a red box.

# Escalate to Video- Task Routed Flow

## Why Task Routed Flow?

- Send customer a personalised message containing the agent they will be speaking with.
- Allow agent to have video portal in their agent desktop

# Escalate to Video- Task Routed Flow



# Escalate to Video- Task Routed Flow

Get Instant Connect Links

Configuration Transition Actions (Optional)

Make GET, POST, PUT, PATCH and DELETE requests to your application

Method: POST      Endpoint URL: \${mtg-broker-url} /api/v2/joseencrypt

Header: Content-Type      Value: application/json

Header: Authorization      Value: Bearer \${videobottoken}

+ Add Another Header

Body:

```
{  
  "jwt": {  
    "sub": "Video Call with your expert"  
  },  
  "aud": "a4d886b0-979f-4e2c-a958-3e8c14605e51",  
  "provideShortUrls": "true",  
  "verticalType": "gen",  
  "loginUrlForHost": false  
}
```

Conn..

HTTP

Get Instant Connect Links

Help

# Escalate to Video- Task Routed Flow

The image shows a task routed flow and its configuration details.

**Task Routed Flow:**

```
graph LR; Start((Conn..)) --> GetInstant[Get Instant Connect Links]; GetInstant --> DataParser[Data Parser]; DataParser -- onSuccess --> End(( ));
```

**Data Parser Configuration:**

**Configuration Tab:**

- Sample Body (optional):**

```
1 [ {  
2   "host": [{  
3     "cipher": "eyJwMnMiOij1bEpxzUtbF8tenR3bkhwNDFLRV  
4     "short": "XoMmeH4"  
5   }],  
6   "guest": [{  
7     "cipher": "eyJwMnMiOijUc0h3VUJTURCumpabUxXejNmU1  
8     "short": "N8tRNkD"  
9   }],  
10  "baseUrl": "https://instant.webex.com/visit/"  
11 }]
```
- SELECT OUTPUT VARIABLES:**
  - \$host[0].short XoMmeH4
  - \$guest[0].cipher eyJwMnMiOijUc0h3VUJTURCumpabUxXejNmU1... JjfwuZv7QbPBdhf1cINTOJihQ7dxX2iTxAwg.vVPu... IRIiIGTXhV1l2IE45BO5w1LdfhmFDPYdc0- b2T2SvxEKn8b7KkFCthpyR4bMlmFtXsuRGPg866K... DBWB\_1ERQdYEeqwMgdcKwbwtFGY\_TDQL2B5f7s... \_GQuNSbc\_rWdwZk9aNcbxaPKRrVQW7iaFsUF4mj... 9xpcUwYSrfqd7JsXopC8\_gVPoWwCU4UhJEw\_yt... TOKQ
  - \$guest[0].short N8tRNkD
- Parsed data output:**

Variable Path	Output Variable Name	Mandatory
\$baseUrl	baseurl	<input type="checkbox"/>
\$host[0].short	hosturl	<input type="checkbox"/>
\$guest[0].short	guesturl	<input type="checkbox"/>
- Buttons:** Parse, Import

# Escalate to Video- Task Routed Flow

**Screen Pop**

**Configuration** **Transition Actions (Optional)**

Method Name: Screen Pop

Node Runtime Authorization: WxCCAuth2

Task ID: \$(n2.webex.ID)

Agent ID: \$(n2.webex.agentId)

**URL Setting**  
Define a URL for Screen Pop configurations and add relevant parameters that need to be associated. If typing variables, use \$variableName.

Screen Pop URL: \$(n2562.baseurl) \$(n2562.hosturl)

**Query Parameters**

Key	Value
Key	Value

+ ADD NEW

**Display Settings**

Screen Pop Loading Behavior: Inside Desktop

```
graph LR; A[Screen Pop] -- "onScreenPopS.." --> B[WhatsApp]
```

The diagram illustrates a task routed flow. It starts with a 'Screen Pop' node (represented by a blue rounded rectangle with a colorful 'C' logo and the text 'Screen Pop') on the left. An arrow labeled 'onScreenPopS..' points from this node to a 'WhatsApp' node (represented by a green rounded rectangle with a green phone icon and the text 'WhatsApp') on the right. Both nodes have small red circular handles at their bottom right corners, indicating they are interactive components.

# Escalate to Video- Task Routed Flow

WhatsApp

Configuration Transition Actions (Optional)

Send text, media, contact, location, interactive messages (list and reply buttons) and proactive template messages to your customers on WhatsApp.

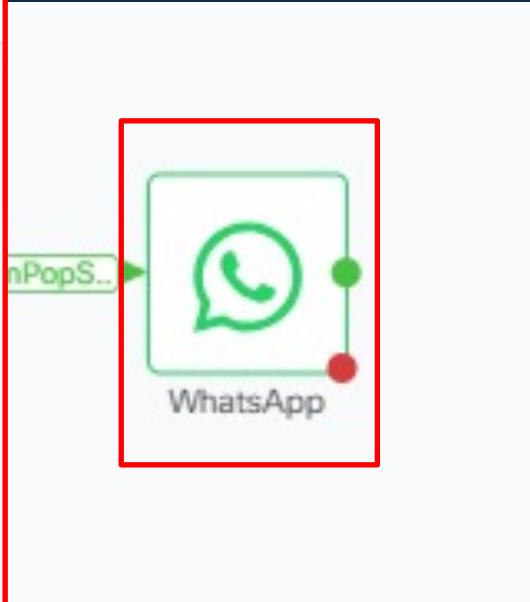
Destination Type i  
WA ID

Message Type i  
TEXT

Message Configuration

Message i  
Thanks you for waiting. Please click on this link to be connected to \$(AgentName) .  
\$(n2562.baseurl)\$(n2562.guesturl)

+ Add Smart Link Characters remaining: 3978



# Call Deflection with context



Tarek

# What is Call Deflection?

Call Deflection is reducing the number of calls that come into your contact center, help desk, or other areas of your business by diverting calls to alternate means of answering customer questions, such as self-service or digital channels.

# Why Call Deflection?

- Digital channels

Voice is an “Expensive” channel

Agents can multitask Digital Channels

- Self-service

Chat bots allow for more complex self-service scenarios like product ordering, appointment management or flight booking

# How to Call Deflect?

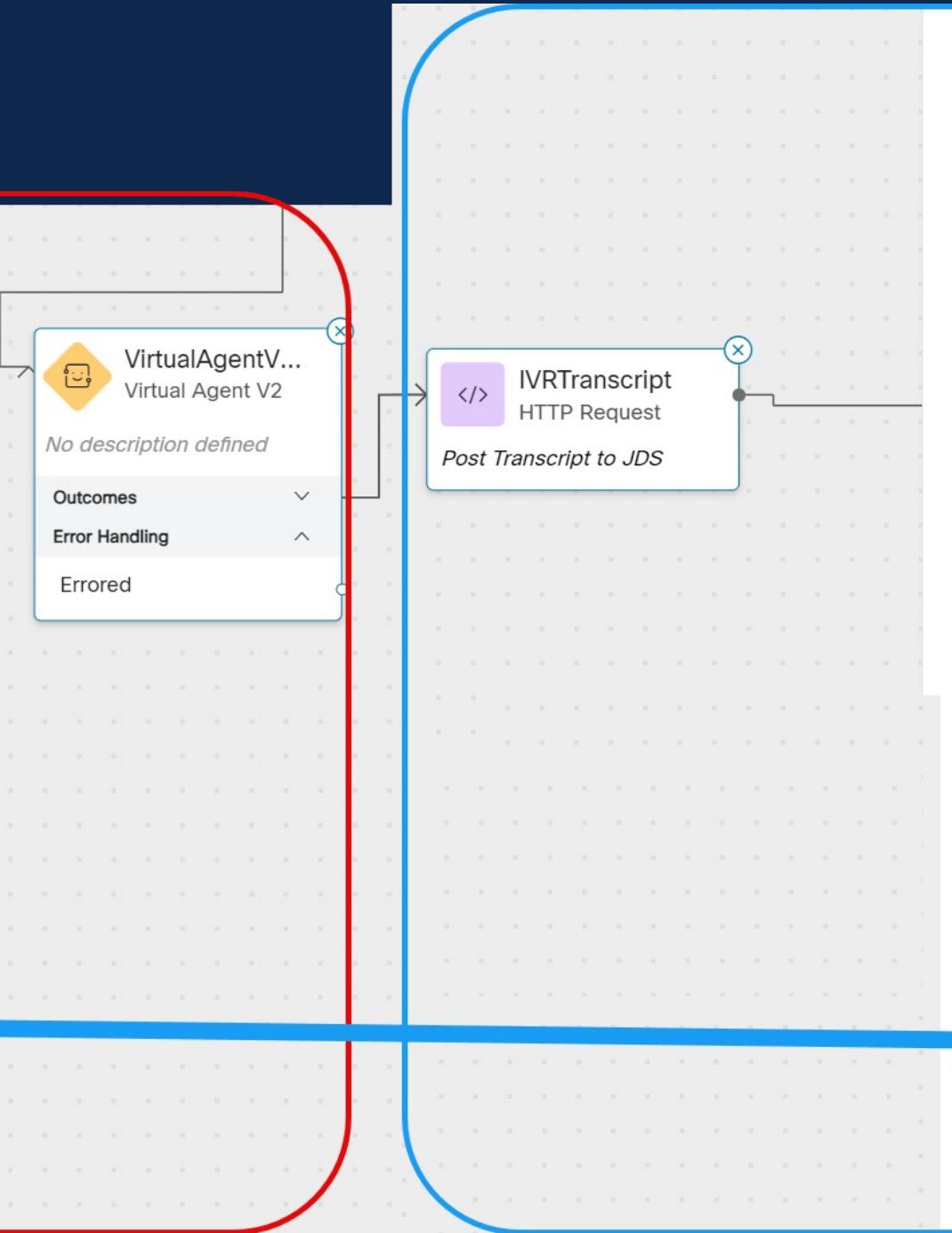
**Volume Gain** ⓘ  
0.0  
Enter a static value or expression that resolves to a value between -96.0 and 16.0. If typing an expression, use this syntax:  
`VirtualAgentV2_6ba.VolumeGain`

**Pitch** ⓘ  
0.0  
Enter a static value or expression that resolves to a value between -20.0 and 20.0. If typing an expression, use this syntax:  
`VirtualAgentV2_6ba.Pitch`

**Enable Conversation Transcript**  
Allow Desktop to display the transcript of the call between the VAV and the customer. The raw transcript is available via a dynamic URL which can be used to extract specific segments of the transcript with a HTTP request.

**Activity Output Variables**

- VirtualAgentV2\_6ba.TranscriptURL
- VirtualAgentV2\_6ba.MetaData
- VirtualAgentV2\_6ba.StateEventName



**Method**  
POST

**Query Parameters**

Key	Value

**Add New**

**HTTP Request Headers**

Key	Value
Content-Type	application/json

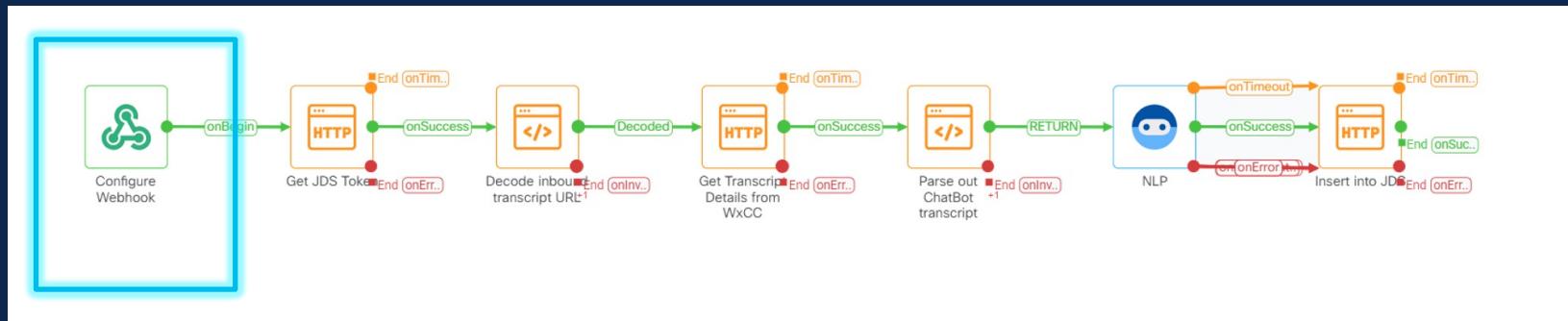
**Add New**

**Content Type**  
Application/JSON

**Request Body**

```
{  
  "transcriptType": "IVR",  
  "transcriptURL": "{{VirtualAgentV2_6ba.TranscriptURL}}",  
  "TaskId": "{{NewPhoneContact.interactionId}}",  
  "JDSWorkspaceld": "{{JDSWorkspaceld}}",  
  "ANI": "{{NewPhoneContact.ANI|slice(1)}}"  
}
```

# What about context?



# Create your webhook

Select existing webhook     Create new event

Webhook Name

We've generated a new endpoint for you to send requests. [Learn more about using webhooks.](#)

Webhook URL

Service key or JWT needs to be passed in request header if this option is selected

Validate signature

Connect can validate JSON signatures generated using SHA256 or SHA512 passed in the header. Invalid request will return a HTTP 400 error

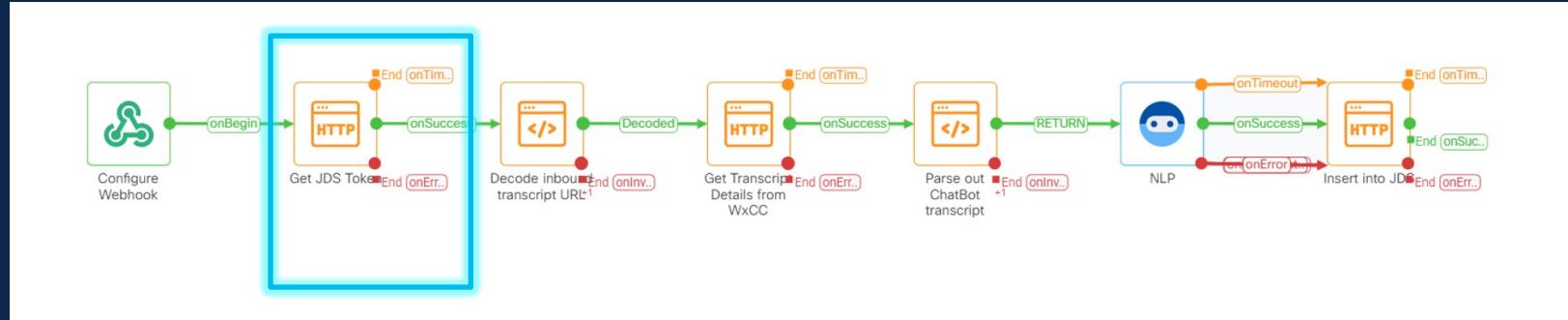
Example Data

**Paste Json** **Paste XML** **Send Via Hook**

PROVIDE SAMPLE INPUT

SELECT OUTPUT VARIABLES

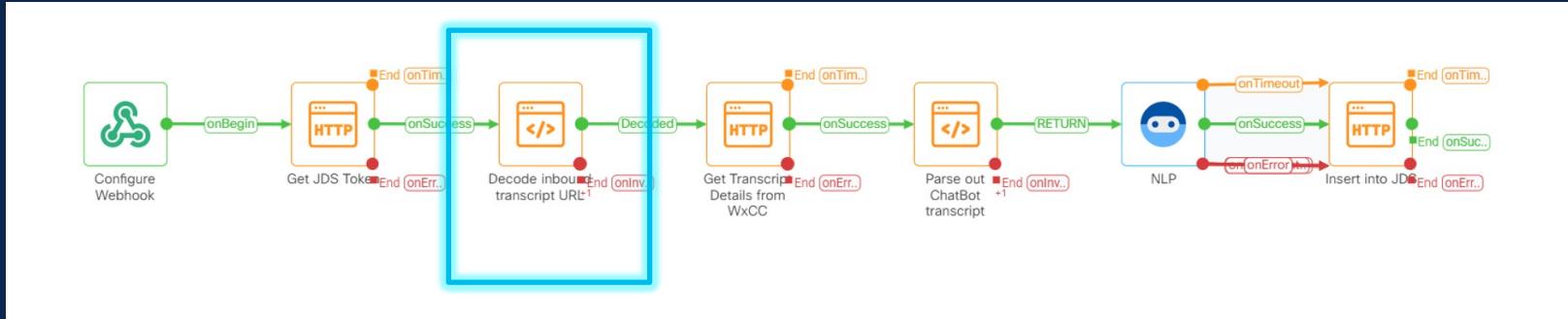
# Manage your token for JDS (for now)



# Convert Hex URL

## Before

```
https://cjp-ccone-produs1-media-storage-recording.s3.amazonaws.com/2572a6d8-b3f7-43a5-9550-7223250c4f5d/6181520b-11f4-4f91-b9eb-8959d6289414/09a3f548-0c9e-4003-91c0-f03590194dc0/ivrTranscript.json?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240403T013059Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=AKIATRZUOXQOKWHH6JML%2F20240403%2Fus-east-1%2Fs3/aws4_request&X-Amz-Signature=87c157893da43051c9234d4f35656b5ae9cf25edf9df9d0456032e822fad0ab1c0a",
```



## After

```
https://cjp-ccone-produs1-media-storage-recording.s3.amazonaws.com/2572a6d8-b3f7-43a5-9550-7223250c4f5d/6181520b-11f4-4f91-b9eb-8959d6289414/09a3f548-0c9e-4003-91c0-f03590194dc0/ivrTranscript.json?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240403T013059Z&X-Amz-SignedHeaders=host&X-Amz-Expires=3600&X-Amz-Credential=AKIATRZUOXQOKWHH6JML/20240403/us-east-1/s3/aws4_request&X-Amz-Signature=87c157893da43051c9234d4f35656b5ae9cf25edf9d0456032e822fad0ab1c0a
```

**Decode inbound transcript URL**

**Configuration** **Transition Actions (Optional)**

Enter JavaScript to perform logic and return values as output. Configured script outputs will be available as node outcome.

```
1 var decode = $("#inboundWebhook.transcriptURL");
2 decode = decode.replace(/%2F/gi, "/");
3 1;
```

**Configure Script Output**

Script Output i Branch Name i

1	Decoded
---	---------

# Get transcript from WxCC

The screenshot shows a workflow diagram at the top and its configuration interface below.

**Workflow Diagram:**

```
graph LR; ConfigureWebhook[Configure Webhook] -- "onBegin" --> GetJDSToken[Get JDS Token]; GetJDSToken -- "End (onTim.)" --> DecodeInbound[Decode inbound transcript URL]; DecodeInbound -- "Decoded" --> GetTranscript[Get Transcript Details from WxCC]; GetTranscript -- "End (onTim.)" --> ParseOut[Parse out ChatBot transcript]; ParseOut -- "onSuccess" --> NLP[NLP]; NLP -- "onTimeout" --> InsertIntoJD[Insert into JD]; NLP -- "onSuccess" --> InsertIntoJD; NLP -- "onError" --> InsertIntoJD; InsertIntoJD -- "End (onSuc.)" --> End[End (onSuc.)]; InsertIntoJD -- "End (onErr.)" --> End;
```

**Configuration Interface:**

**Section Headers:** Configuration, Transition Actions (Optional), Help

**Method:** GET

**Endpoint URL:** \${decode}

**Header:** e.g., Authorization

**Value:** e.g., Bearer ya29.c.Elo4BQQH\_js2AmoDSaln\_J-3hQ-k0SYDHI

**Connection Timeout:** e.g., 1000 ms

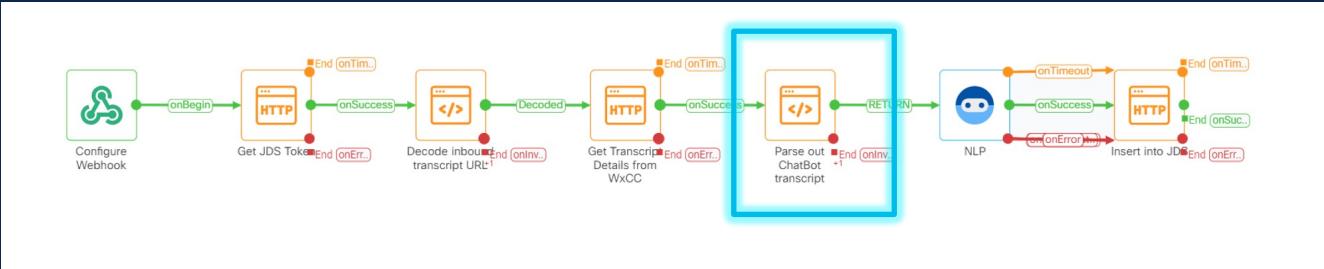
**Request Timeout:** e.g., 1000 ms

**Proxy Address (Optional):** e.g., 192.128.19.243:80

**Output Variables:** JSON (selected), XML, Import From Sample, + Add Variable

**Response:** The response body will be stored in n4.http.responseBody variable.

# Parse the transcript



```
jsonObject = JSON.parse("${n4.http.responseBody}"); //Value returned as {"value":"{JSON}"}
```

```
var customerSentence = "";
var customerQueries = "";
var botSentence = "";
var FullConversation = "";
var arrayLength = jsonObject.conversation.length;
for (var i = 0; i < arrayLength; i++) {
    // Check if the customer object and the query property exist
    if (jsonObject.conversation[i].customer && jsonObject.conversation[i].customer.query !== undefined) {
        customerSentence = jsonObject.conversation[i].customer.query.replace(/\r\n/g, "");
    } else {
        customerSentence = ""; // Use "no response" if the query property does not exist
    }

    // Check if the bot object and the reply property exist
    if (jsonObject.conversation[i].bot && jsonObject.conversation[i].bot.reply !== undefined) {
        botSentence = jsonObject.conversation[i].bot.reply.replace(/\r\n/g, "");
    } else {
        botSentence = "no response"; // Use "no response" if the reply property does not exist
    }

    if (customerSentence === "") {
        FullConversation += 'VA: ' + botSentence + '\n';
    } else {
        FullConversation += 'Cust: ' + customerSentence + '\nVA:' + botSentence + '\n';
    }
    customerQueries += customerSentence + ". ";
}
FullConversation = JSON.stringify(FullConversation);
1;
```

## Parse out ChatBot transcript

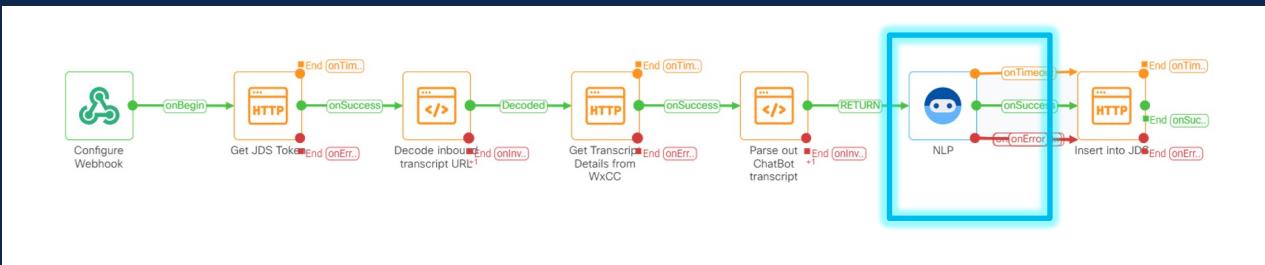
### Configuration Transition Actions (Optional)

Enter JavaScript to perform logic and return values as output. Configured script outputs will be available in the transition actions.

```
1 jsonObject = JSON.parse("${n4.http.responseBody}"); //Value returned as {"value":"{JSON}"}
2 //jsonObject = JSON.parse('{"conversation":[{"bot":{"intentName":"Default Welcome Intent","co
3
4 var customerSentence = "";
```



# NLP Sentiment Analysis



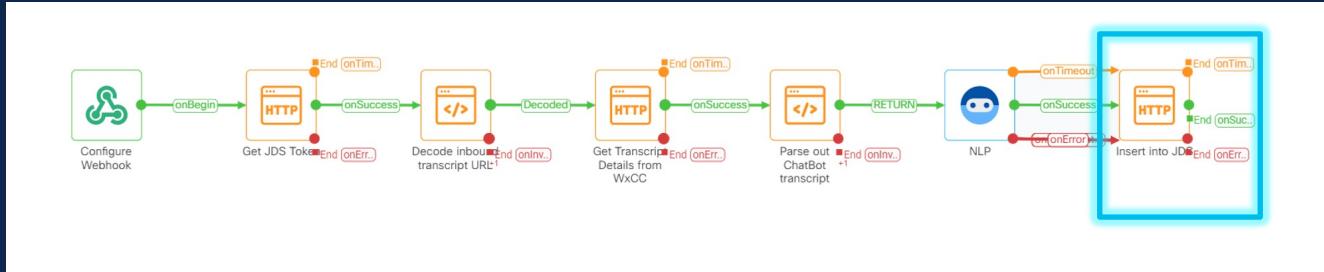
### NLP

[Configuration](#)    [Transition Actions \(Optional\)](#)

Method Name  
Sentiment Analysis

MESSAGE  
\$(customerQueries)

# Post Into JDS



Insert into JDS

**Configuration**    **Transition Actions (Optional)**

Make GET, POST, PUT, PATCH and DELETE requests to your application

Method: POST    Endpoint URL: \${JDS\_BaseURL} /publish/v1/api/event?workspaceId= \${JDS\_workspaceId}

Header

Content-Type	Value: application/json
Authorization	Bearer \${JDS_Token}

+ Add Another Header

Body

```
{  
  "id": "${n2.inboundWebhook.transId}",  
  "time": "${inboundWebhook.timestamp}",  
  "specversion": "1.0",  
  "type": "IVR Transcript",  
  "source": "VirtualAgent",  
  "identity": "${inboundWebhook.ANI}",  
  "identitytype": "phone",  
  "datacontenttype": "application/json",  
  "data": {  
    "transId": "${n2.inboundWebhook.TaskId}",  
    "phone": "${inboundWebhook.ANI}",  
    "customerSentiment": "${n22.DetectedSentiment}",  
    "transcript": ${FullConversation},  
    "channelBreakout": "telephony",  
    "uiData": {  
      "title": "IVR Transcript",  
      "iconType": "bot",  
      "subTitle": "Virtual Agent and Customer Transcript",  
      "filterTags": [  
        "Transcript",  
        "Virtual Agent"  
      ]  
    }  
  }  
}
```

```
{  
  "id": "${n2.inboundWebhook.transId}",  
  "time": "${inboundWebhook.timestamp}",  
  "specversion": "1.0",  
  "type": "IVR Transcript",  
  "source": "VirtualAgent",  
  "identity": "${inboundWebhook.ANI}",  
  "identitytype": "phone",  
  "datacontenttype": "application/json",  
  "data": {  
    "transId": "${n2.inboundWebhook.TaskId}",  
    "phone": "${inboundWebhook.ANI}",  
    "customerSentiment": "${n22.DetectedSentiment}",  
    "transcript": ${FullConversation},  
    "channelBreakout": "telephony",  
    "uiData": {  
      "title": "IVR Transcript",  
      "iconType": "bot",  
      "subTitle": "Virtual Agent and Customer Transcript",  
      "filterTags": [  
        "Transcript",  
        "Virtual Agent"  
      ]  
    }  
  }  
}
```

# Get Transcript back out of JDS

```
$(JDS_BaseURL)/v1/api/events/workspace-  
id/$(JDS_workspaceId)?identity=$(n2.inboundWebhook.ANI)&filter=type=='IVR  
Transcript'&data=transId=='$(n2.inboundWebhook.TaskId)''&pageSize=1
```

**Get Transcript**

[Configuration](#) [Transition Actions \(Optional\)](#) [Help](#)

Make GET, POST, PUT, PATCH and DELETE requests to your application

Method: **GET** Endpoint URL: `$(JDS_BaseURL) /v1/api/events/workspace-id/ $(JDS_workspaceId) ?identity= $(n2.inboundW`

Header

Content-Type	Value: application/json
Authorization	Bearer \$(JDS_Token)

+ Add Another Header

Connection Timeout: 20000 ms Request Timeout: 20000 ms

Proxy Address (Optional): e.g., 192.128.19.243:80

Output Variables

JSON  XML [Import From Sample](#)

Output Variable Name	Response Entity	Response Path
data_transcript	Body	\$.data[0].data.transcript
source	Body	\$.data[0].source
type	Body	\$.data[0].type
meta_resultCount	Body	\$.meta.resultCount
DetectedSentiment	Body	\$.data[0].data.customerSentiment

+ Add Variable

Response

The response body will be stored in `n2095.http.responseBody` variable.

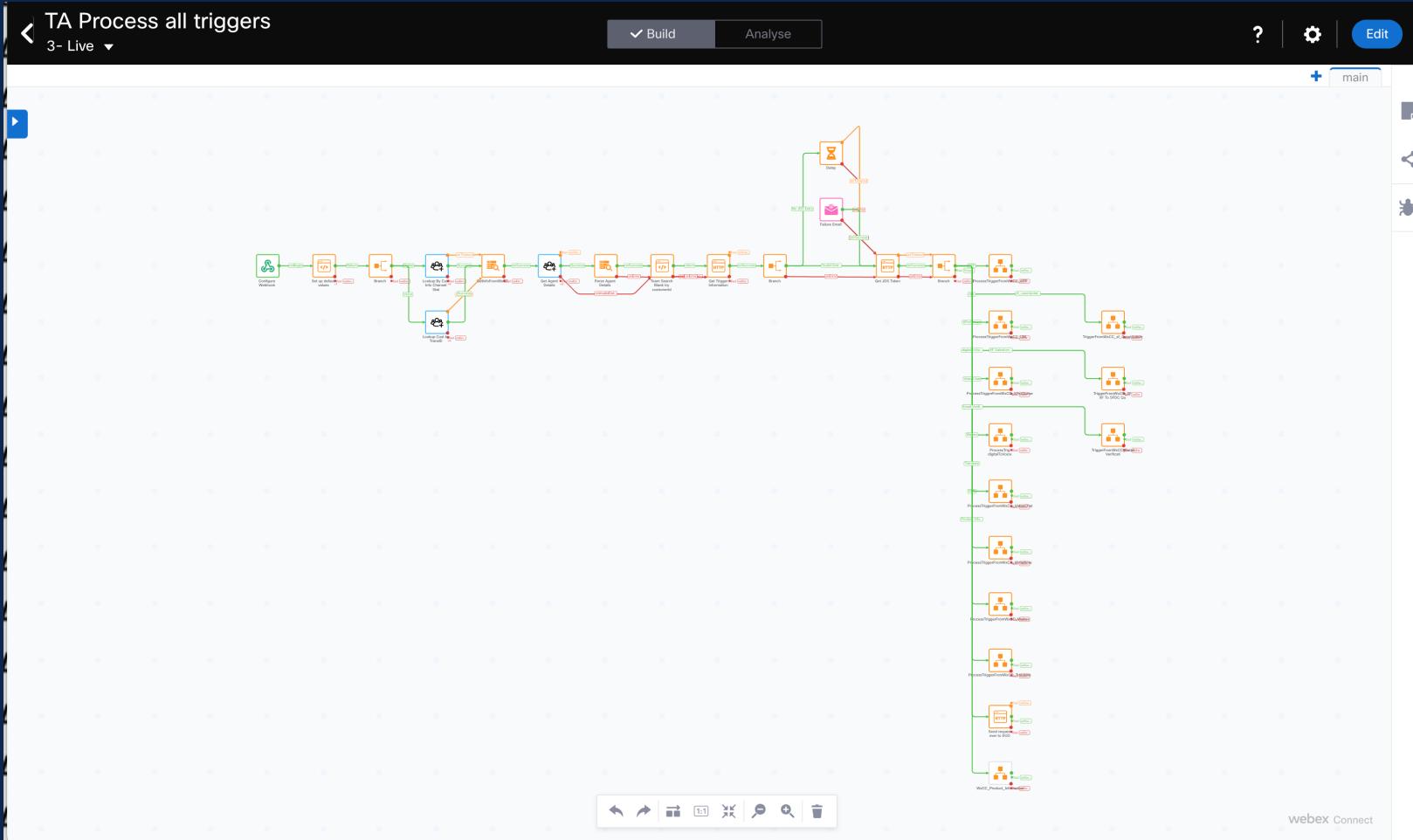
# Other ways of implementation



Tarek

# Process triggers flow to unify the collection of information

One script to rule them ALL !!



# Process triggers flow to unify the collection of information

One script to rule them ALL !!

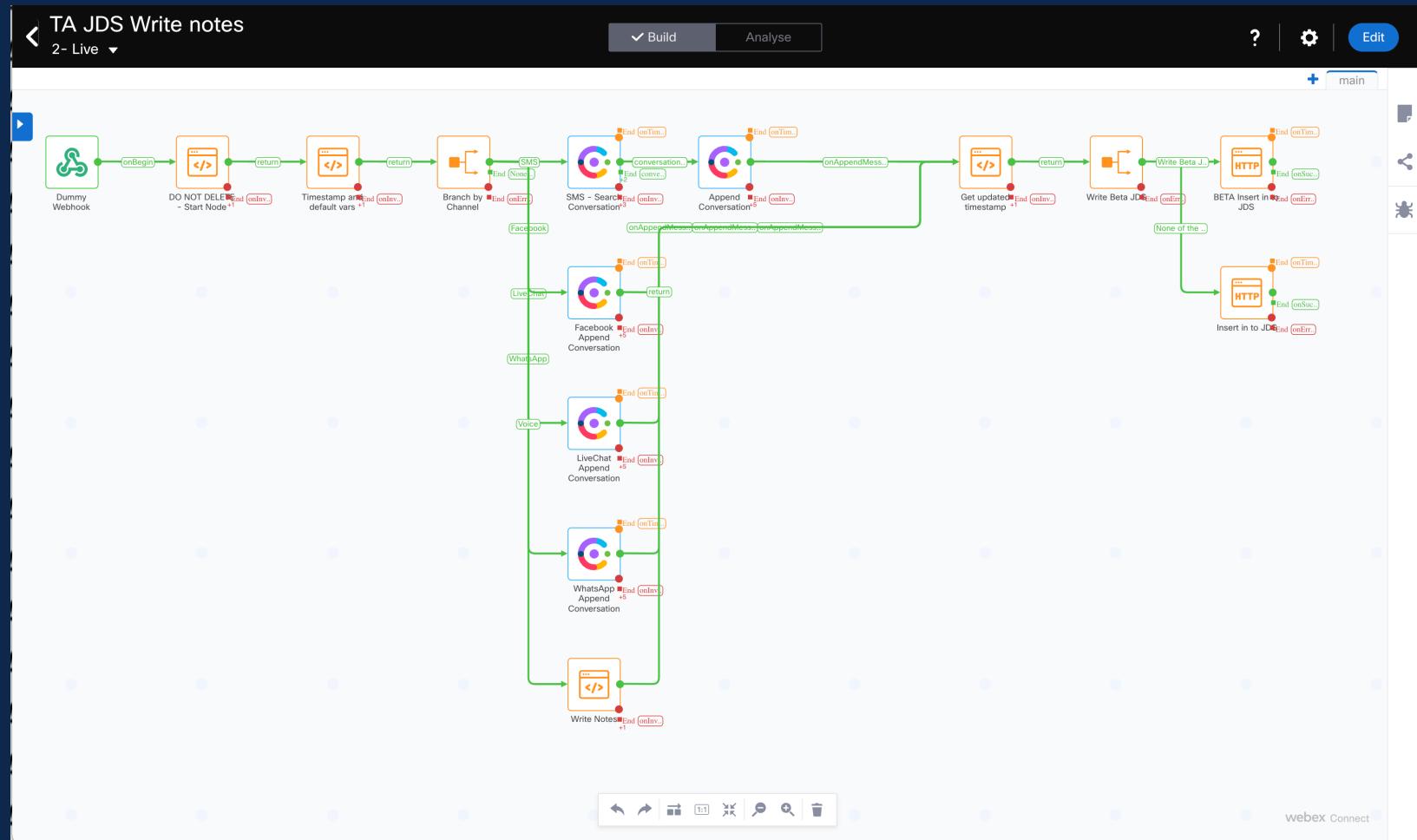
The screenshot shows the configuration interface for a process named "TA Process all triggers". The main window displays a "Configure Webhook" dialog with the following settings:

- Configuration:** Select existing webhook (radio button selected).
- Webhook Name:** TA Process all triggers.
- Webhook URL:** https://hooks.us.webexconnect.io/events/GFXQ2I4440
- Example Data:** A JSON snippet is provided for reference.

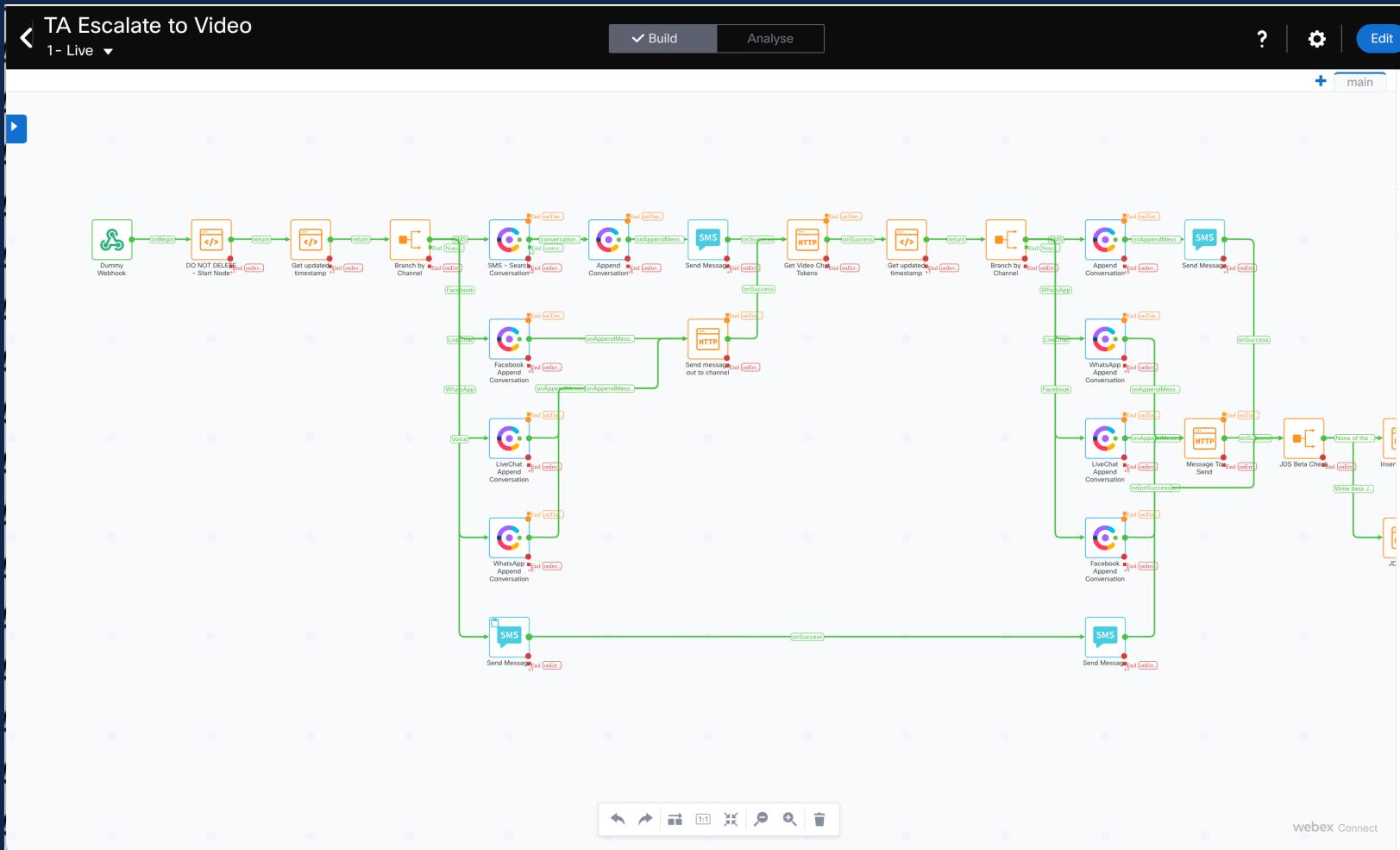
A modal dialog titled "Input Variables" is open, listing available variables:

- PARSED VARIABLES(10)
  - thread id:thread needed for chat
  - customer id:customer id for Chat
  - selection:Dropdown list on trigger
  - assetid:Asset ID ~ actual number if using SMS
  - customerEmail:If from desktop we pick specific user to send email to
- Output Variables
- Node Outcomes

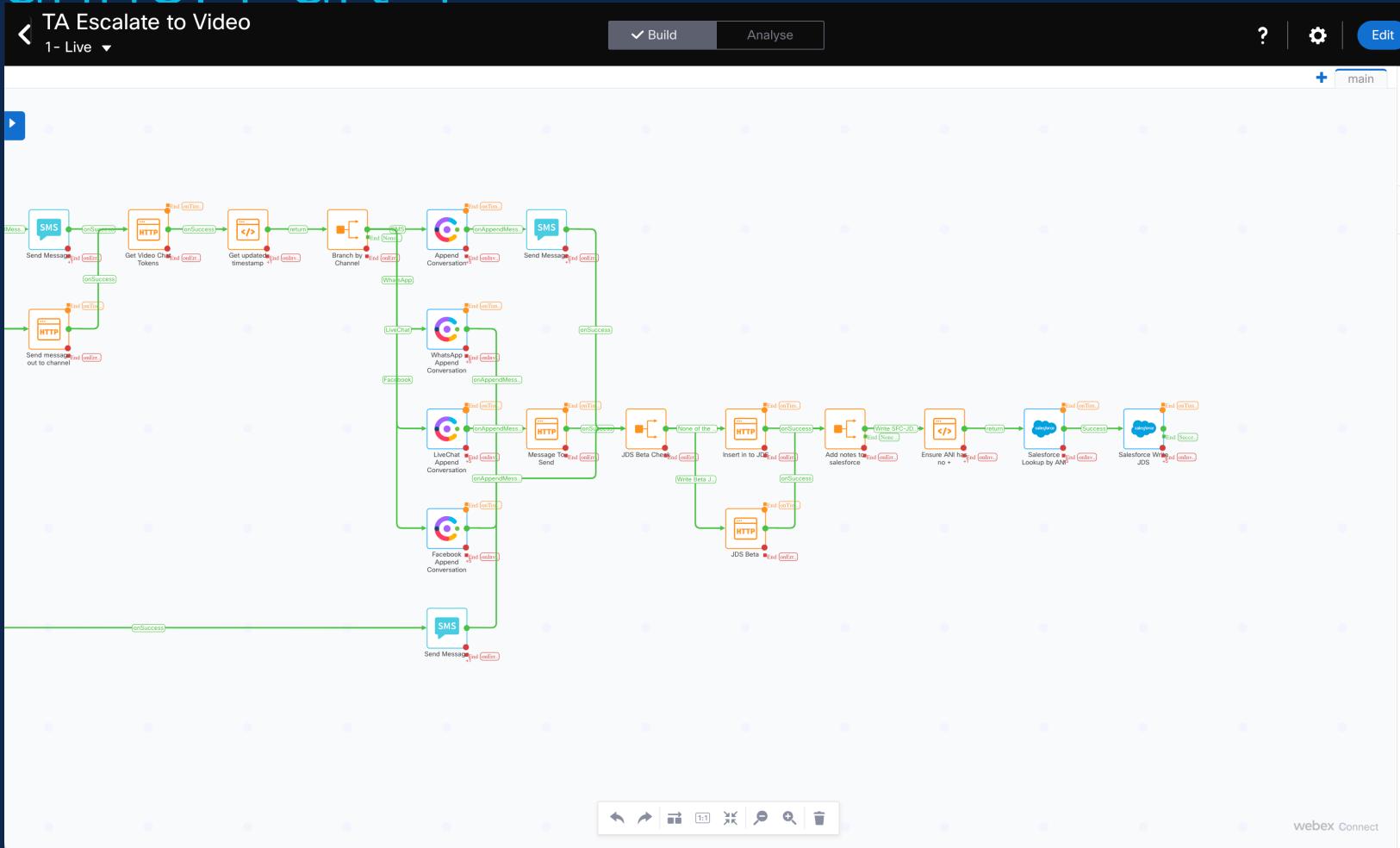
# Call into other use cases - Write notes



# Call into other use cases - Escalate to Video from any channel Part 1



# Call into other use cases - Escalate to Video from any channel Part 1



# Recording TAGs



**Mahesh  
Mansanipalli  
(Americas)**

# Tag Recording Issue

The image shows two separate web pages side-by-side. Both pages have a header with a search bar and a dropdown for 'Show' entries (set to 10).  
The left page is titled 'Custom Attributes' and lists one entry: 'ManualAttribute'. It has a red dashed box around the 'Name' column.  
The right page is titled 'Tags' and lists three entries: 'ManualTAG', 'NOTWORKINGTAG', and 'rtyhrty'. It also has a red dashed box around the 'Name' column.  
Both pages have a footer with a 'Previous' button, a page number '1', and a 'Next' button.



A modal window titled 'Tagging-Recording for session ID : 604df972-e2dc-463e-9193-a5421d228dc0'.  
The window contains three main sections:

- Available Tags:** A list box containing 'NOTWORKINGTAG', 'rtyhrty', and other tags. It has a red dashed box around the 'NOTWORKINGTAG' entry.
- Assign Tag:** A large blue button with a play icon and the text 'Assign Tag'.
- Remove Tag:** A large blue button with a back icon and the text 'Remove Tag'.

  
On the right side of the window, there is a list box titled 'Assigned Tags' containing the tag 'ManualTAG', which is also highlighted with a red dashed box.

# Tag Recording Issue

POST https://api.wxcc-{{datacenter}}.cisco.com/v1/captures/query

Send

Params Authorization Headers (13) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "query": {
3     "orgId": "e56f00d4-98d8-4b62-a165-d05a41243d98",
4     "urlExpiration": 1,
5     "taskIds": [
6       "604df972-e2dc-463e-9193-a5421d228dc0"
7     ],
8     "includeSegments": false
9   }
10 }
```

Body Cookies Headers (17) Test Results Status: 200 OK Time: 233 ms Size: 1.62 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "meta": {
3     "orgId": "e56f00d4-98d8-4b62-a165-d05a41243d98",
4     "urlExpiration": 1
5   },
6   "data": [
7     {
8       "taskId": "604df972-e2dc-463e-9193-a5421d228dc0",
9       "recording": [
10         {
11           "id": "604df972-e2dc-463e-9193-a5421d228dc0",
12           "segment": false,
13           "attributes": {
14             "fileName": "combinedRecordings.wav",
15             "filePath": "https://cjp-ccone-produs1-media-storage-recording.s3.amazonaws.com/e56f00d4-98d8-4b62-a165-d05a41243d98/604df972-e2dc-463e-9193-a5421d228dc0/
16               604df972-e2dc-463e-9193-a5421d228dc0/combinedRecordings.wav?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240306T14000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=608
17               X-Amz-Credential=AKIATRZUOXQOKWHH6JML%2F20240306%2Fs-east-1%2Fs3%2Faws4_request&X-Amz-Signature=71cdc84f34dec9ac937f00bfa8c6e878c769a3b9b16b92ddcc00943c19afe6e2",
18             "startTime": 1709731216611,
19             "stopTime": 1709731223051,
20             "participants": [
21               "+18447727524",
22               "37ce41f0-c533-4167-b212-546f0341b1a0"
23             ],
24             "channel1": "Caller",
25             "channel2": "Agent",
26             "callType": "main"
27           }
28         ]
29       }
30     ]
31   }
32 }
```

# Use Case

## **Problem Description/UseCase:**

There is a need to pull out recording from Contact Center Storage, based on a specific tag and/or account id and then save them in external storage.

The Tag should be added to a call after verbal approval from client side.

1. Client sends a request to store recording.

2. Client sends a request to store recording.

3. Client sends a request to store recording.

4. Client sends a request to store recording.

# Possible Solution

Global Variable

General Settings

Name	Recording_TAG
Description	Agent must set YES (True) if verbal approval comes from the caller
Variable Type	Boolean
Default Value	false
Status	Active

Reporting Settings

Make Reportable	Yes
This enables the variable to appear in Analyzer for reporting purposes. Each tenant has a limit of 100 variables.	

Agent Settings

Make Agent Viewable	Yes
This enables the variable to be displayed on Desktop with the value captured as part of the Flow.	
Desktop Label	Special Request
Agent Editable	Yes
If made editable, updates to the variable made from Desktop will be received by the Flow.	

[Edit](#) [Delete](#) [Back](#)

**RecordingTag**  
Global Flow Properties

Flow variables are fully configurable local variables. They can never be viewed in reports, but can optionally be agent viewable.

**Flow Variables**

[View All Variables](#)

[Add Flow Variable](#)

**Predefined Variables**

Predefined Variables are added through the Variable definition, or through the Global Variables section of a flow. These variables are available locally in a flow. These variables are available locally in a flow. These variables are available locally in a flow.

**Record\_String**

Agent Viewable
Reportable

Variable Type String  
Default Value  
Contains Sensitive Information No

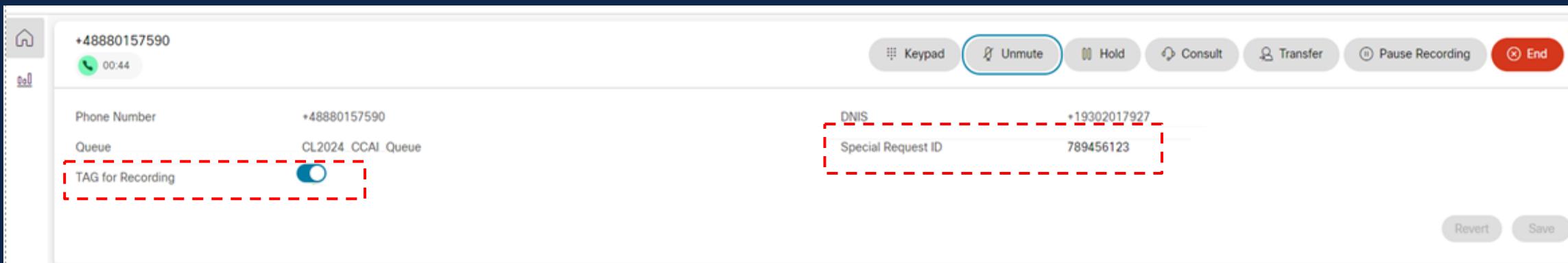
Desktop Label Special Request ID  
Agent Editable Yes

[View All Variables](#)

[Recording\\_TAG](#) [Record\\_String](#)

[Add Global Variables](#)

# Possible Solution



# Possible Solution

The screenshot illustrates a workflow for recording a phone call. On the left, a Cisco call interface shows a recording tag being set and a special request ID being passed through. On the right, a Postman API request is shown, which filters tasks based on these parameters and returns a task with a matching recording ID.

**Cisco Call Interface:**

- Phone Number: +48880157590
- Queue: CL2024\_CCAI\_Queue
- Special Request ID: 789456123
- Recording Tag: TAG for Recording (highlighted with a red box)

**Postman API Request:**

```
POST https://api.wxcc-((datacenter)).cisco.com/search
```

**QUERY**

```
1 query($epochTimeMs: Long!, $epochTimeMs1: Long!)\n2 {\n3   # Filter tasks based on Global Variables\n4   taskDetails(\n5     from: $epochTimeMs1\n6     to: $epochTimeMs\n7     filter: [\n8       and: [\n9         {\n10           booleanGlobalVariables: {\n11             name: { equals: "Recording_TAG" }\n12             value: { equals: true }\n13           }\n14         ]\n15       ]\n16     ]\n17   )\n18   tasks: [\n19     id\n20     booleanGlobalVariables(name: "Recording_TAG") {\n21       name\n22       value\n23     }\n24     stringGlobalVariables(name: "Record_String") {\n25       name\n26       value\n27     }\n28   ]\n29 }
```

**Body Results (Pretty View):**

```
4   "tasks": [\n5     {\n6       "id": "b6d2c21a-5495-4e4b-ab62-f285a9a85bf",\n7       "booleanGlobalVariables": {\n8         "name": "Recording_TAG",\n9         "value": true\n10       },\n11       "stringGlobalVariables": {\n12         "name": "Record_String",\n13         "value": "789456123"\n14       }\n15     },\n16   ],\n17 }
```

A red box highlights the "Recording\_TAG" entry in the "booleanGlobalVariables" section of the task object. A green arrow points from the "TAG for Recording" button in the Cisco interface to this highlighted section. Another green arrow points from the "Special Request ID" field in the Cisco interface to the "Record\_String" entry in the "stringGlobalVariables" section of the task object.

D = Recording ID

POST https://api.wxcc-{{datacenter}}.cisco.com/search

Params • Authorization Headers (11) Body • Pre-request Script • Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL Auto Fetch C A

**QUERY**

```
1 query($epochtimems: Long!, $epochtimems1h: Long!){  
2   # Filter tasks based on Global Variables  
3   taskDetails(  
4     from: $epochtimems1h  
5     to: $epochtimems  
6     filter: {  
7       and: [  
8         {  
9           booleanGlobalVariables: {  
10             name: { equals: "Recording_TAG" } ← Filter: What we are looking for  
11             value: { equals: true }  
12           }  
13         }  
14       ]  
15     }  
16   }  
17 } {  
18   tasks {  
19     id  
20     booleanGlobalVariables(name: "Recording_TAG") { ← Output: What we want to see  
21       name  
22       value  
23     }  
24     stringGlobalVariables(name: "Record_String") { ←  
25       name  
26       value  
27     }  
28   }  
29 }  
30 pageInfo {  
31   # Indicates if the nextPage exists or not  
32   hasNextPage  
33   # String value which can be passed to fetch the next page.  
34   endCursor  
35 }  
36 }  
37 }
```

Filter: What we are looking for

Output: What we want to see

# File Path to Recording

POST https://api.wxcc-{{datacenter}}.cisco.com/v1/captures/query

Send

Params Authorization Headers (13) Body Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "query": {
3     "orgId": "e56f00d4-98d8-4b62-a165-d05a41243d98",
4     "urlExpiration": 1,
5     "taskIds": [
6       "604df972-e2dc-463e-9193-a5421d228dc0"
7     ],
8     "includeSegments": false
9   }
10 }
```

Body Cookies Headers (17) Test Results Status: 200 OK Time: 233 ms Size: 1.62 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "meta": {
3     "orgId": "e56f00d4-98d8-4b62-a165-d05a41243d98",
4     "urlExpiration": 1
5   },
6   "data": [
7     {
8       "taskId": "604df972-e2dc-463e-9193-a5421d228dc0",
9       "recording": [
10         {
11           "id": "604df972-e2dc-463e-9193-a5421d228dc0",
12           "segment": false,
13           "attributes": {
14             "fileName": "combinedRecordings.wav",
15             "filePath": "https://cjp-ccone-produs1-media-storage-recording.s3.amazonaws.com/e56f00d4-98d8-4b62-a165-d05a41243d98/604df972-e2dc-463e-9193-a5421d228dc0/604df972-e2dc-463e-9193-a5421d228dc0/combinedRecordings.wav?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20240306T14000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=608&X-Amz-Credential=AKIATRZUOXQOKWHH6JML%2F20240306%2Fs-east-1%2Fs3%2Faws4_request&X-Amz-Signature=71cdc84f34dec9ac937f00bfa8c6e878c769a3b9b16b92ddcc00943c19afe6e2",
16             "startTime": 1709731216611,
17             "stopTime": 1709731223051,
18             "participants": [
19               {
20                 "id": "604df972-e2dc-463e-9193-a5421d228dc0",
21                 "name": "combinedRecordings.wav",
22                 "channel1": "Caller",
23                 "channel2": "Agent",
24                 "callType": "main"
25               }
26             ]
27           }
28         ]
29       }
30     ]
31   }
32 }
```

# Final & most important quiz!



# Technical Partner Summits are awesome.

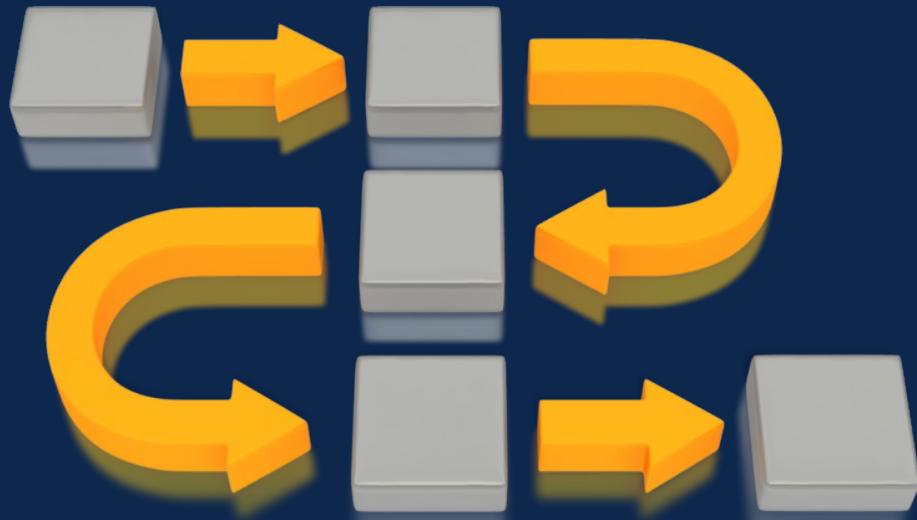


Your feedback is so valuable to US ....





# Dynamic business hour announcement



**Siva Venkatesh Arcot**

siarcot@cisco.com

# Use Case

Customer ABC would like to include the next business reopening date dynamically with the existing office hour closure announcement, whether due to a local or national holiday or a specific situation.

## Gotcha's

- Currently, WxCC has no direct integration to any calendar.
- Business Hours Node does not offer the capability out-of-box

So How do you achieve this use case??

# Approach...

- Utilize the Business Hours Node to specify the name and date of holidays for playback to the caller.
- Define a specific Global Variables, variable type: "DateTime" corresponds to reopening date for the holiday/closure.
- Flow script includes a case statement that evaluates Name of the Holiday derived from Business Hours node to the corresponding business reopening day.
- Utilize the Set Variable function to assign the reopening date.
- CCAI-TTS is employed to playback holiday/closure along with re-open day announcement.

# Solution

## Pre-requisite configurations:

- Use TTS
- BusinessHours (Holiday List) e.g., Christmas.
- Define Global Variables; corresponding day of re-open for the holiday Christmas = “**WorkDate**”

e.g. Variable Name= **WorkDate** , Variable Type: **Date Time** , Default Value= “**ReOpenDay**” Supported input formats; **YYYY-MM-DD** or **YYYY-MM-DDThh:mm**.

- Flow Variables as required to read and parse from “BusinessHours\_HolidayName”, “CaseStatements”, “SetVariable”

- TTS SSML sample

```
<speak>  
Thank you for calling ABC. Our office is currently closed for holiday observance. We will resume our regular business  
hours on <say-as interpret-as="date" format="yyyymmdd" detail="1">{{REOPENDATE}} </say-as>, from 8 a.m. to 8 p.m.  
Eastern Standard Time.
```

```
<break time="350ms"/>
```

```
During this time, we invite you to leave a brief message with your name, number, company, the event you're inquiring  
about and reason for calling. A member of our service team will return your call.<break time="200ms"/>Thank you and  
have a great day.
```

```
</speak>
```

## Flow Designer

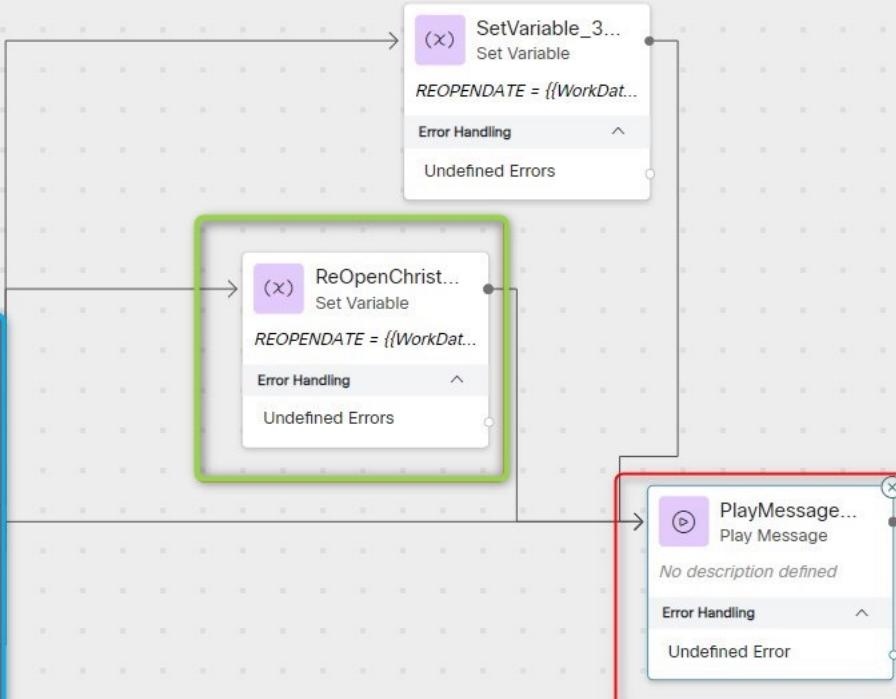
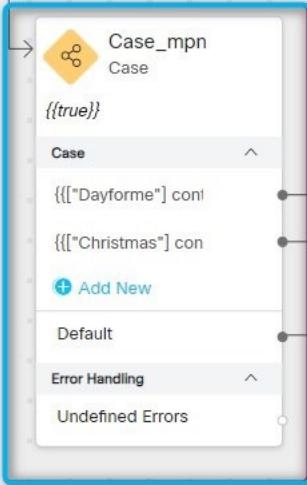
Main Flow

Event Flows

Help

Search Flow

Ctrl K



Autosave Enabled

Application Version: 0.4.415-0

Debug

PlayMessage\_j19

Play Message Activity Settings

in a sequence.

### Text-to-Speech Message

<speak>

Thank you for calling ABC. Our office is currently closed for holiday observance. We will resume our regular business hours on <say-as interpret-as="date" format="yyyyymmdd" detail="1">

<{{REOPENDATE}}>, from 8 a.m. to 8 p.m.  
Eastern Standard Time.

<break time="350ms"/>

During this time, we invite you to leave a brief message with your name, number, company, the event you're inquiring about and reason for calling. A member of our service team will return your call.<break time="200ms"/> Thank you and have a great day.

Add Audio File

Add Audio Variable

Autosave

Validation: Off

Publish Flow

# Common issues & workaround



**Siva**



**When customer sends an email with Plain Text format instead of HTML format, the agent never receives the contact which can impact end user experience and the business as well.**

TRANSACTION LOGS ff520773-6014-4100-9f39-fe919b6dea3f\_31476\_153698 X HELP G X

Trans ID : ff520773-6014-4100-9f39-fe919b6dea3f\_31476\_153698 Time Elapsed (HH:mm:ss.ms) : 00 : 00 : 00 . 902 Invoked by : Rule 31476

NODE ID	↑↓ NODE	↑↓ OUTCOME	↑↓ TIME TAKEN(HH:mm:ss.ms)	↑↓ DETAILS
2	Configure Email Event	onBegin	00 : 00 : 00 . 002	Request Method : GET Request URL : http://integrations-us.imiconnect.io/v1/integration/cntwct Request Data : { "comciscoorgId": "6608122d-50f5-4558-be72-4e9f26c6a42f", "datacontenttype": "application/json", - "data": { "reason": "Create conversation failed:4616\nError Description: The request is invalid.\nError Details: List(The Body field is required.)" }, "transId": "ff520773-6014-4100-9f39-fe919b6dea3f", "id": "ff520773-6014-4100-9f39-fe919b6dea3f", "reasonCode": 1001, "operation": "created", "orgId": "6608122d-50f5-4558-be72-4e9f26c6a42f" }, ResponseContent : {}
9	Parse Variables	success	00 : 00 : 00 . 001	
1894	Resolve Conversation	accepted	00 : 00 : 00 . 463	
1965	End	waiting	00 : 00 : 00 . 000	
1894	Resolve Conversation	taskFailed	00 : 00 : 00 . 123	
1832	Email	onsuccess	00 : 00 : 00 . 011	
1957	End	Success	00 : 00 : 00 . 000	

Resolve Conversation node ends with the error:

**"Create conversation failed:4616\nError Description: The request is invalid.\nError Details: List(The Body field is required.)". The outcome of resolve conversation node is taskFailed.**

## **What are the options/suggestions to avoid this issue?**

1. Can we ask end customer to send an email with HTML format?
2. No, we cannot ask the end customer to send emails in a particular format. So, what else can be done?
3. In Webex Connect workflow, we are checking for HTML format. Instead of checking for the HTML format, we can use the same message type with which the email was sent. This will solve the problem for any email format.

# Plain Text Email - Solution

New Auth Simple Email  
Working Draft

Build Analyse ? Save Make Live

Utilities Channels Integrations Q main

Parse Variables

Configuration Transition Actions (Optional)

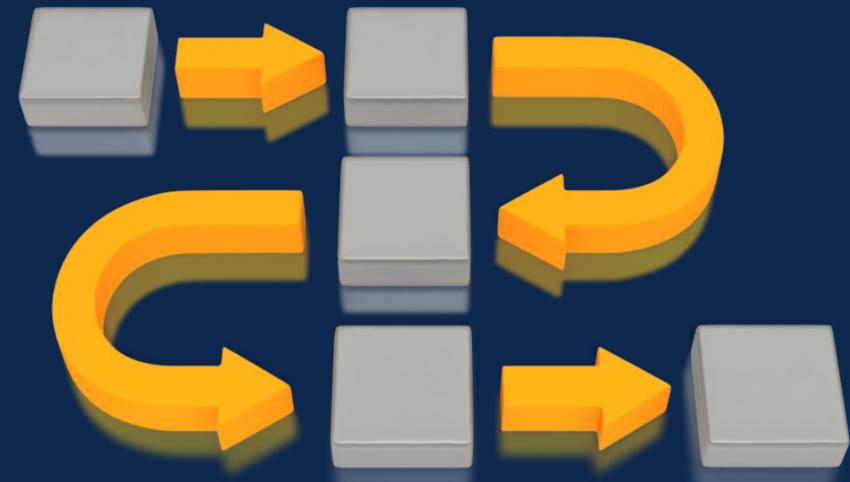
Enter JavaScript to perform logic and return values as output. Configured script outputs will be available as node outcome

```
168 var messageDetails = "{" +  
169 " \\"appID\\": \\" + appId + "\\", "  
170 " \\"assetType\\": \\" + assetType + "\\", "  
171 " \\"senderName\\": \\" + senderName + "\\", "  
172 " \\"recipient\\": \\" + bizEmailId + "\\", "  
173 " \\"businessEmailId\\": \\" + bizEmailId + "\\", "  
174 " \\"customerEmailId\\": \\" + customerEmailId + "\\", "  
175 " \\"customerName\\": \\" + senderName + "\\", "  
176 " \\"toAddresses\\": " + JSON.stringify(toAddresses) +  
177 " \\"ccRecipients\\": " + JSON.stringify(ccRecipients) +  
178 " \\"bccRecipients\\": " + JSON.stringify(bccRecipients) +  
179 " \\"subject\\": \\" + subject + "\\", "  
180 " \\"inReplyTo\\": \\" + inReplyTo + "\\", "  
181 " \\"headers\\": " + JSON.stringify(headers) + ", " +  
182 " \\"message\\": \\" + messageText + "\\", "  
183 " \\"htmlMessage\\": \\" + messageHTML + "\\", "  
184 " \\"strippedText\\": \\" + strippedText + "\\", "
```

The **messageDetails** JSON prepared in Parse Variables node has **htmlMessage** value as **"htmlMessage"**. Replace **"htmlMessage"** with **"messageHTML"** to fix the issue.

This screenshot shows a workflow editor interface for a 'New Auth Simple Email' project. The main canvas displays a process flow starting with a 'Configure Email Event' node, followed by a 'Parse Variables' node (which is highlighted with a red border), a 'Resolve Conversation' node, another 'Email' node, a 'Queue Task' node, and a final 'Email' node. Various transition events like 'onBegin', 'onSuccess', 'onError', 'onTimeout', 'onReopen', 'onAccepted', 'onQueued', and 'onSuccess' are shown connecting the nodes. On the left, a sidebar lists utility nodes such as Evaluate, Branch, HTTP Request, Delay, Data Parser, Data Transform, Call Flow, Page Connector, and Profile. A tooltip for the Parse Variables node shows its configuration code, which includes a JSON object with various fields like appID, assetType, and messageText. A callout box highlights a section of the code where 'htmlMessage' is used, with a note suggesting it should be replaced with 'messageHTML'. The top right of the screen includes standard application controls for Build, Analyse, Save, and Make Live.

# Empty Email Subject



When customer sends an email to business without the email subject, the agent never receives the contact which can impact end user experience and the business as well.

TRANSACTION LOGS		52570e2e-5657-4c8e-a029-2cce2f643b84_2455_9640			
				Time Elapsed (HH:mm:ss.ms) : 00 : 00 : 01 . 616	
NODE ID	↑↓ NODE	↑↓ OUTCOME	↑↓ TIME TAKEN(HH:mm:ss.ms)	DETAILS	
2	Configure Email Event	onBegin	00 : 00 : 00 . 000	amp	
9	Parse Variables	success	00 : 00 : 00 . 001	Request URL :	http://integrations.imiconnect.ca/v1/integration/cntwct
1894	Resolve Conversation	accepted	00 : 00 : 01 . 378	Request Data :	{ "comciscoorgId": "8379be82-65a4-453c-b67e-1d79d416c628", "datacontenttype": "application/json", "data": { "reason": "Create conversation failed:4616\\nError Description: The request is invalid.\\nError Details: List(The Subject field is required.)", "transId": "52570e2e-5657-4c8e-a029-2cce2f643b84", "id": "52570e2e-5657-4c8e-a029-2cce2f643b84", },
1949	End	waiting	00 : 00 : 00 . 000		
1894	Resolve Conversation	taskFailed	00 : 00 : 00 . 001		
1832	Email	onerror	00 : 00 : 00 . 001		
1833	End	Error	00 : 00 : 00 . 000		

Resolve Conversation node ends with the error:

**"Create conversation failed:4616\\nError Description: The request is invalid.\\nError Details: List(The Subject field is required.)". The outcome of resolve conversation node is taskFailed.**

# What are the options/suggestions to avoid this issue?

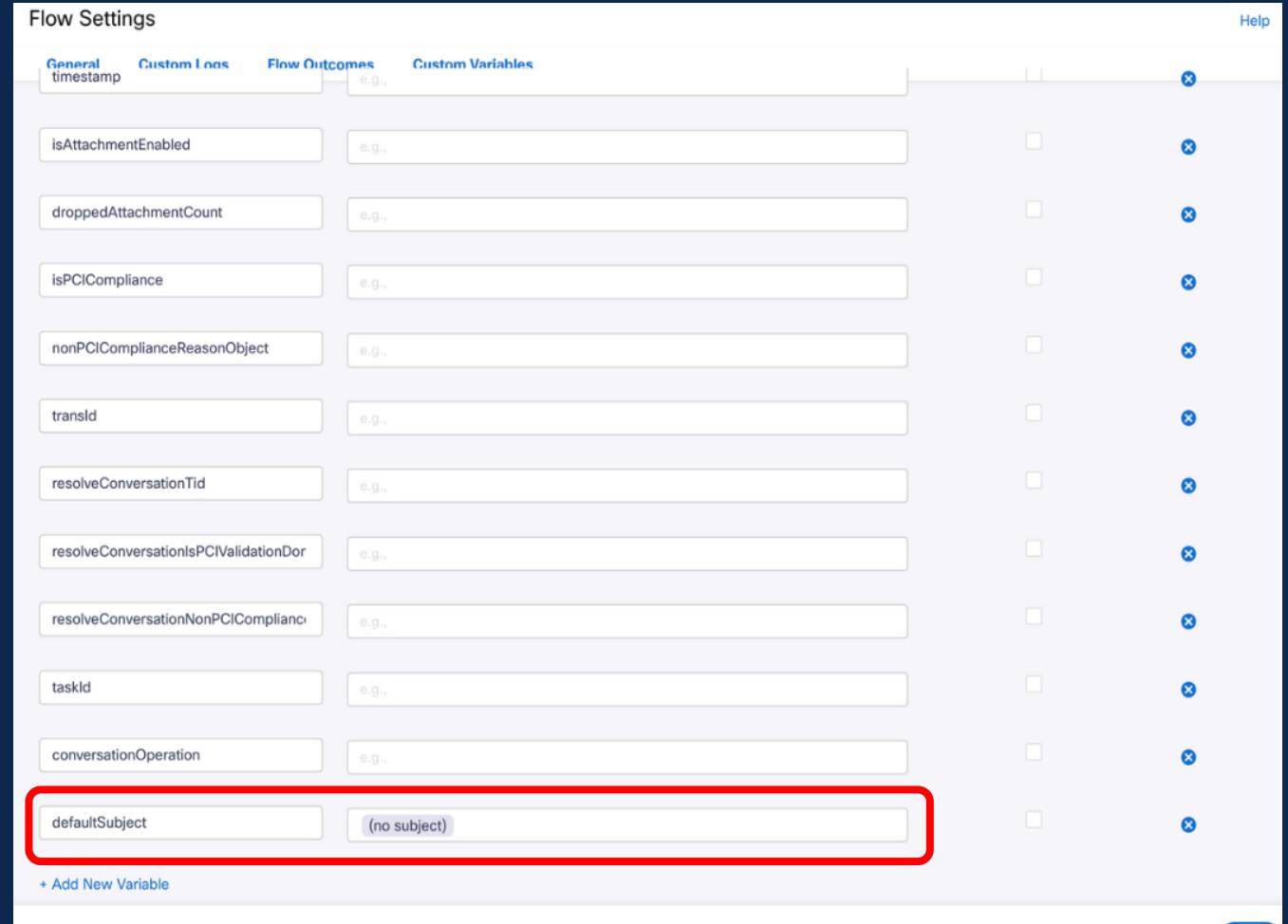
1. Can we ask end customer not to send emails with empty subject?
2. No, we cannot. So, what else can be done?
3. Add a default subject line and use it if the one in email is empty.
4. How can we make the default subject line unique?
5. Add Timestamp to avoid Append conversations.

# Empty Email Subject - Solution

## Step1:

Add a new custom variable '**defaultSubject**' in flow settings.  
Assign value '**(no subject)**' to the variable.

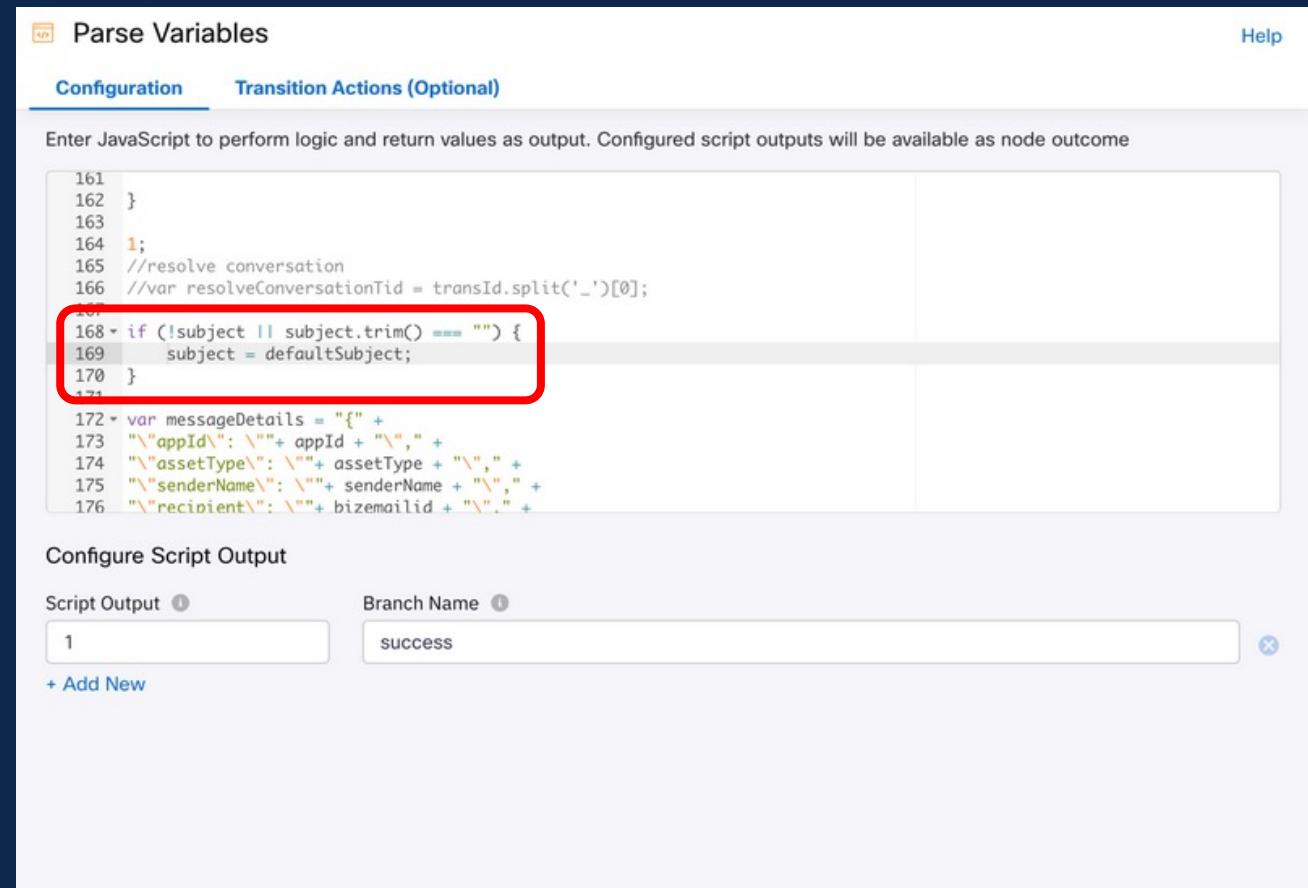
We can assign any standard values for the variable '**defaultSubject**'.



# Empty Email Subject - Solution

## Step2:

The messageDetails JSON prepared in **Parse Variables** node will check for null/empty/whitespace subject. If any of the **condition holds true**, Javascript code will assign '*subject*' value as '*defaultSubject*' from **Custom Variable** in-order to avoid taskFailed outcome of Resolve Conversation. The subject value assigned in this node will be picked up as default value of subject if it missing.



The screenshot shows the 'Parse Variables' configuration screen. The 'Configuration' tab is selected. In the 'Script' section, there is a block of JavaScript code. A red box highlights the following code snippet:

```
161
162 }
163
164 1;
165 //resolve conversation
166 //var resolveConversationId = transId.split('_')[0];
167
168 if (!subject || subject.trim() === "") {
169     subject = defaultSubject;
170 }
171
172 var messageDetails = "{" +
173     "\"appId\": " + appId + "\", " +
174     "\"assetType\": " + assetType + "\", " +
175     "\"senderName\": " + senderName + "\", " +
176     "\"recipient\": " + bizEmailId + ", "
```

Below the script, there is a 'Configure Script Output' section with 'Script Output' set to '1' and 'Branch Name' set to 'success'.

If the **condition is false**, no re-assignment to subject variable.

## Code Snippet:

```
if (!subject || subject.trim() === "") {
    subject = defaultSubject;
}
```

# Empty Email Subject - Solution

## Step3:

In the flow, all email node's subject should be configured to **`$(subject)`**.

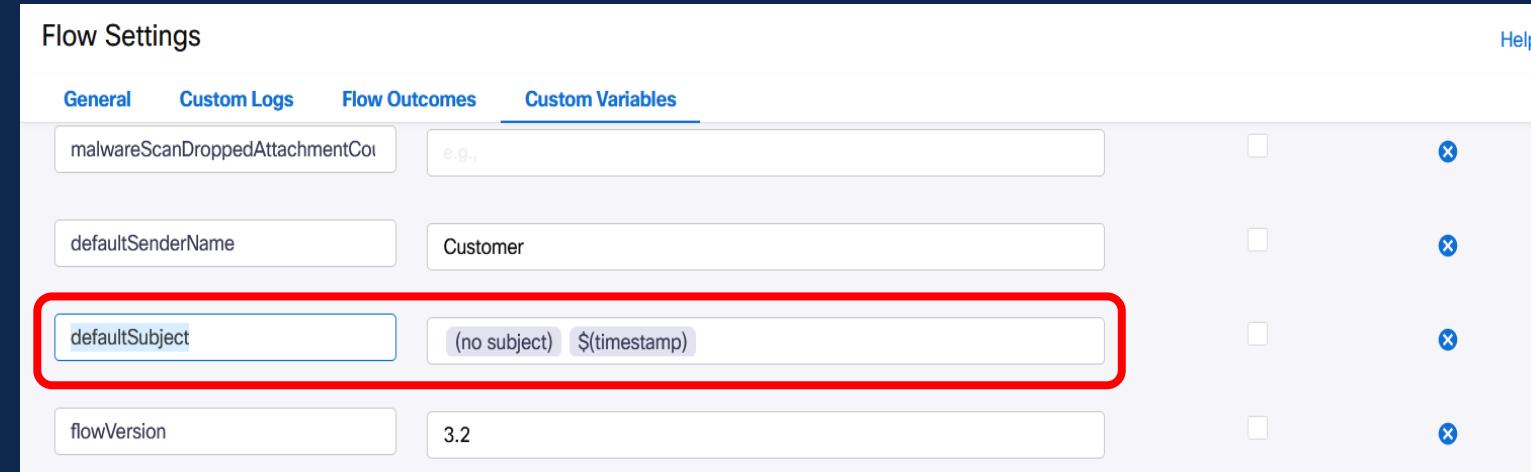
This step should be performed in all the email nodes in the flow to avoid errors.

The screenshot shows the configuration of an Email node in MuleSoft Anypoint Studio. The 'Configuration' tab is selected. In the 'Message Source' section, the 'Subject' field is highlighted with a red box and contains the expression `$(subject)`. Other fields in the configuration include 'Destination Type' set to 'Email Id', 'From Email' set to 'Enter from email address', 'CC Recipients' containing 'example1@abc.example2@abc', 'BCC Recipients' containing 'example1@abc.example2@abc', 'Email Type' set to 'Text', and a 'Message' box containing 'Message is resolved.' The 'Input Variables' panel on the right lists variables such as 'Custom Variables', 'Start', 'Evaluate', and 'Resolve Conversation' with their respective Node IDs.

# Empty Email Subject - Solution

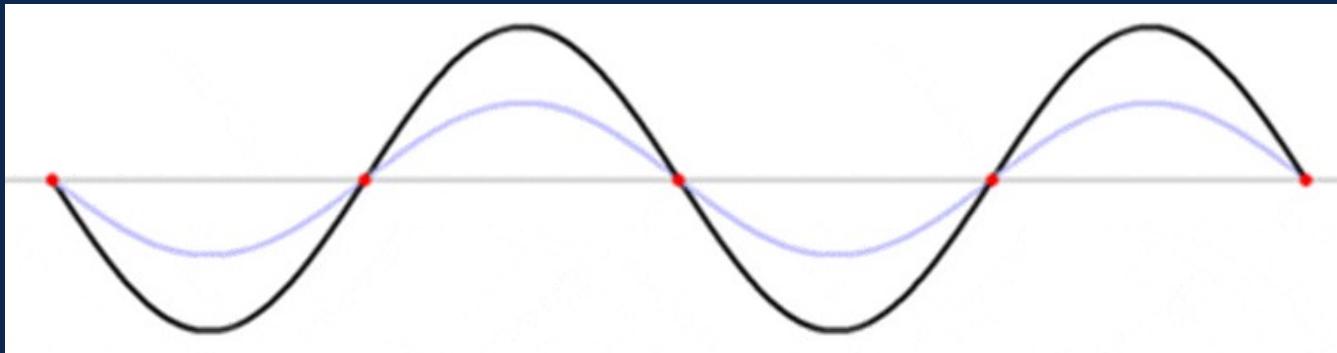
## Note:

If Customer sends an email to the business email id without any subject and if same or other Customer sends another email to the same business email id without any subject, then the second email will be appended to the first email thread.



To avoid this, in the custom variable flow settings, add ***\$(timestamp)*** in defaultSubject value. This will add the current timestamp after the default subject.

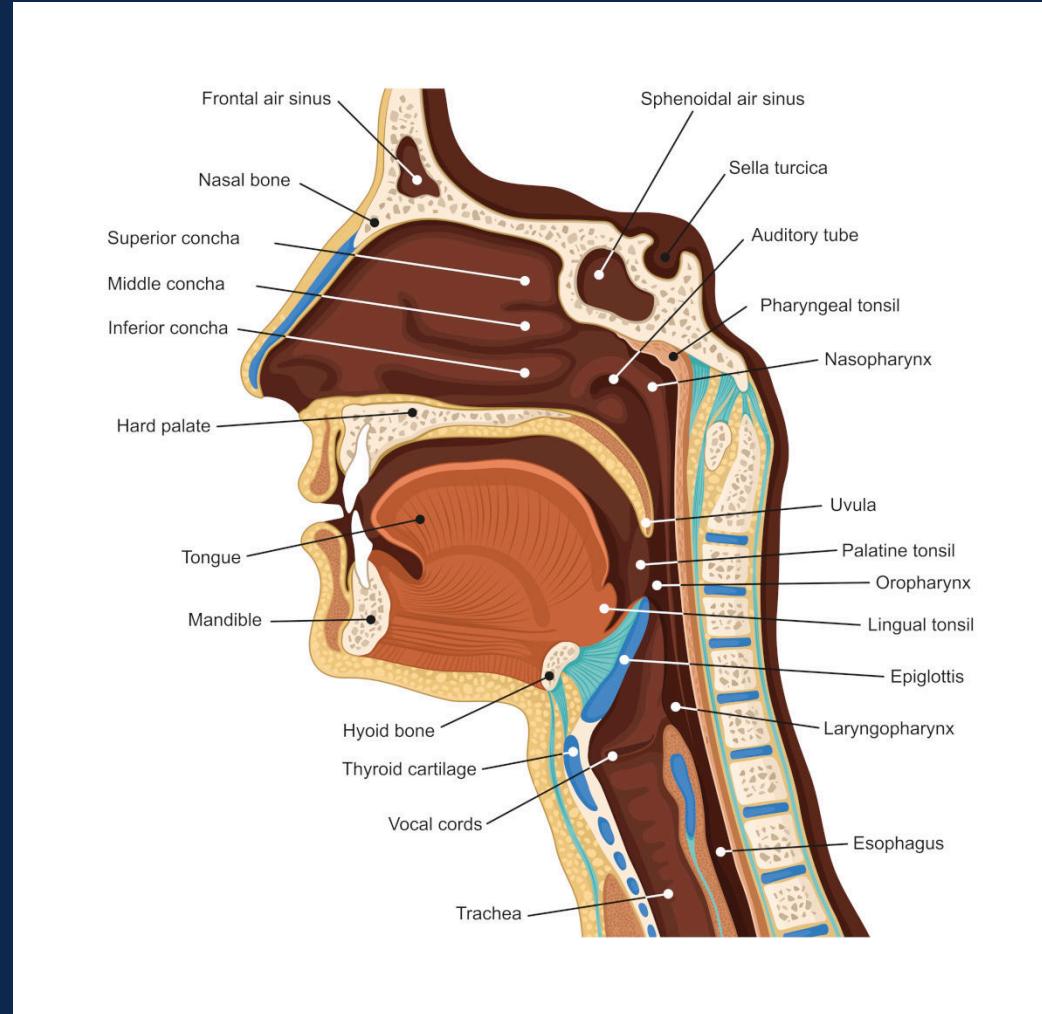
# Voice Biometrics



**Holger Esser**



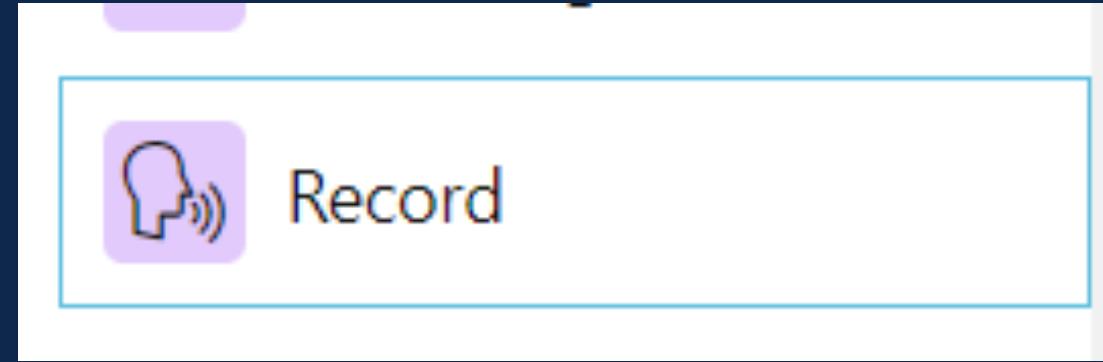
# Voice Biometrics Background



# Voice Biometrics Challenges Brainstorming



# Our new Recording Node (star of the show)



# Voice Bio (supporting cast)

HTTPRequest\_8f0\_wzs

HTTP Request Activity Settings

**General Settings**

Activity Label: HTTPRequest\_8f0\_wzs

Activity Description: Check for enrollment call

**HTTP Request Settings**

Use Authenticated Endpoint:

Request URL: https://demointeractive.org:5000/isSpeakerEnrolled

Method: POST

Query Parameters

Key	Value

isSpeakerVerified

Agent Viewable

Variable Type: String

Default Value

Contains Sensitive Information: No

Desktop Label: Authentication

Agent Editable: No

Parse Settings

Content Type: Select

Output Variable: isSpeakerVerified

Path Expression: \$result.decision

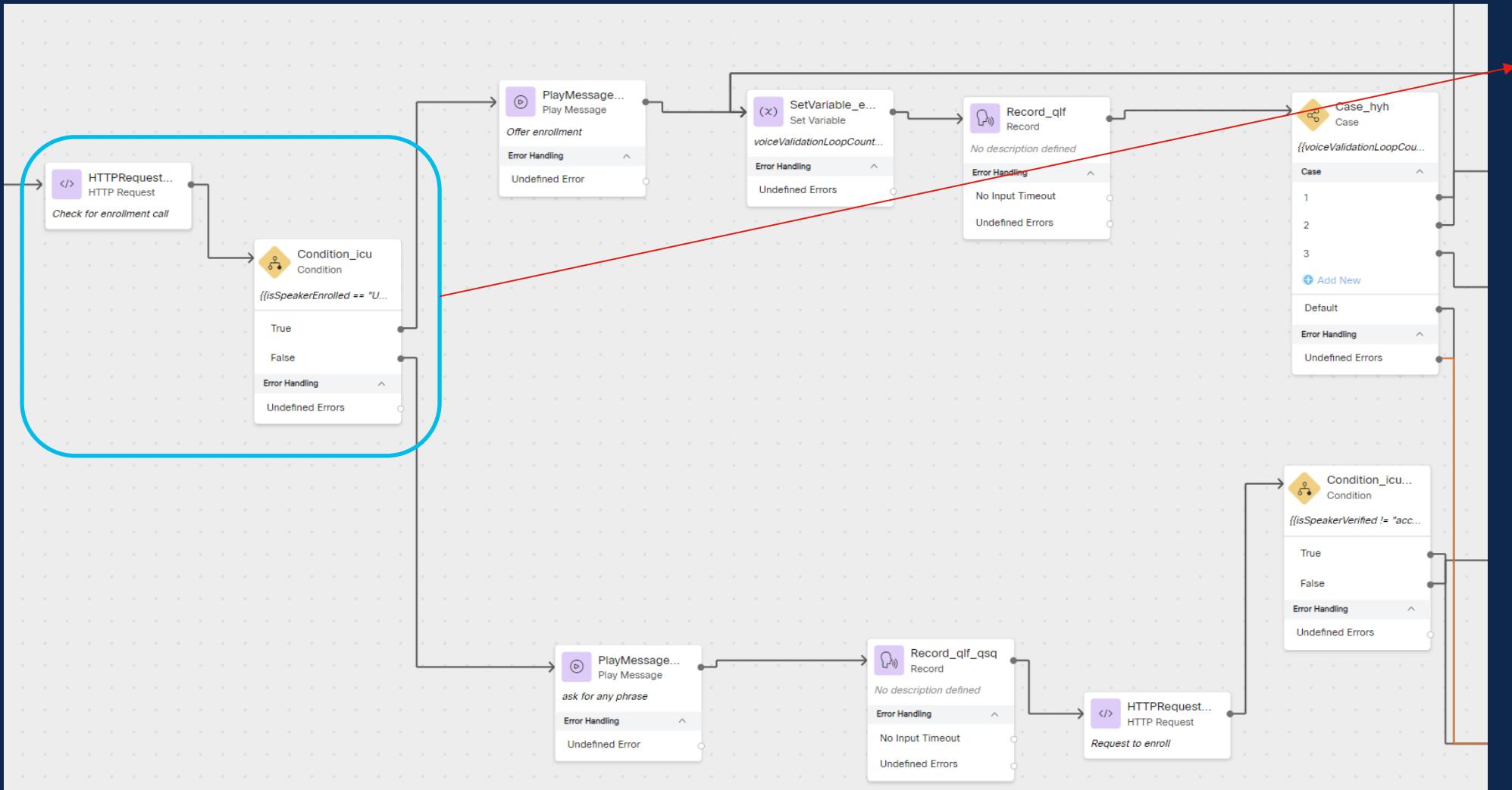
Add New

Activity Output Variables

Phone Number	+12143265416
Queue	hoesser_queueVoice
Authentication Status	accepted

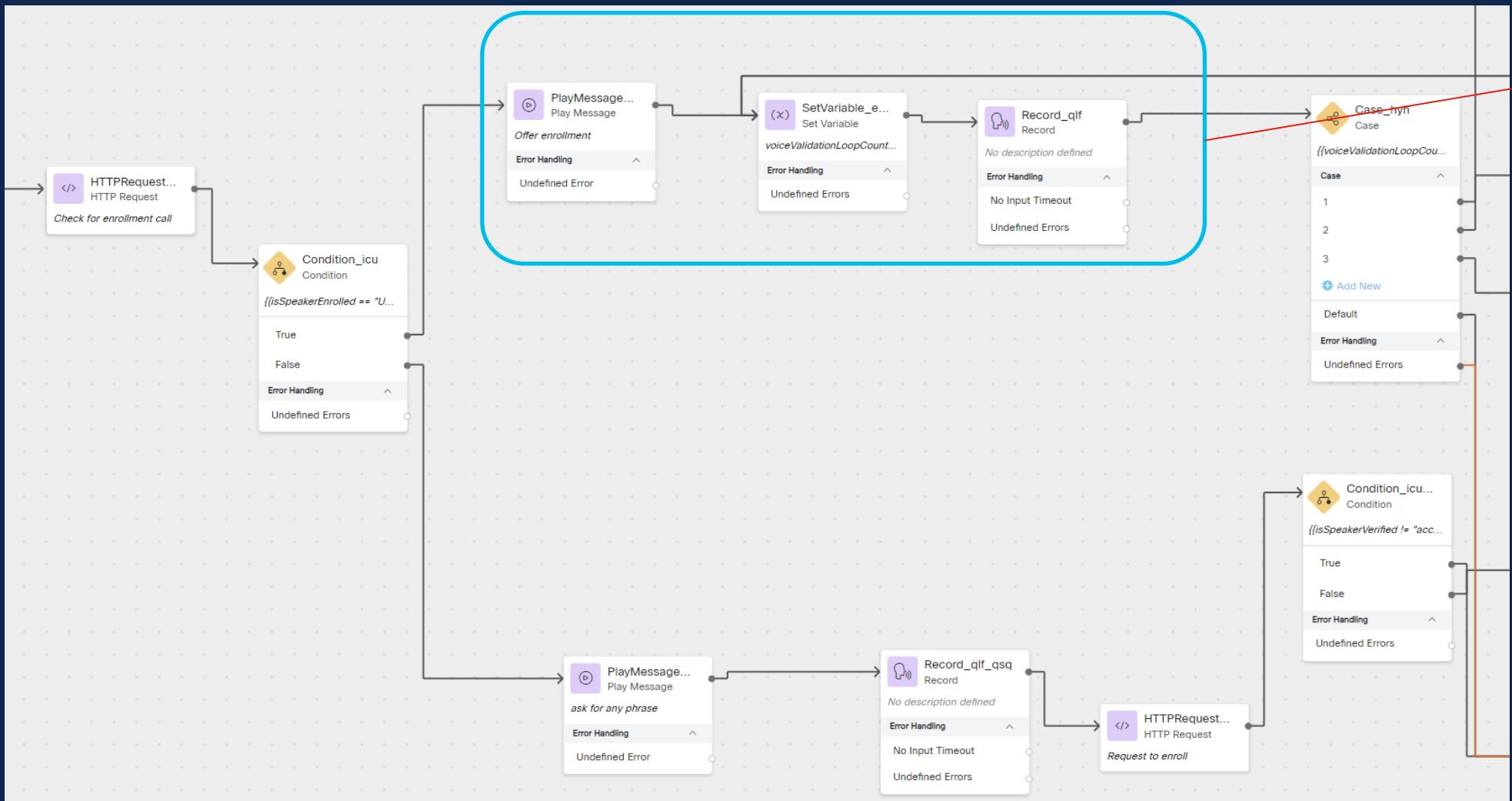
# DEMO

# Flow



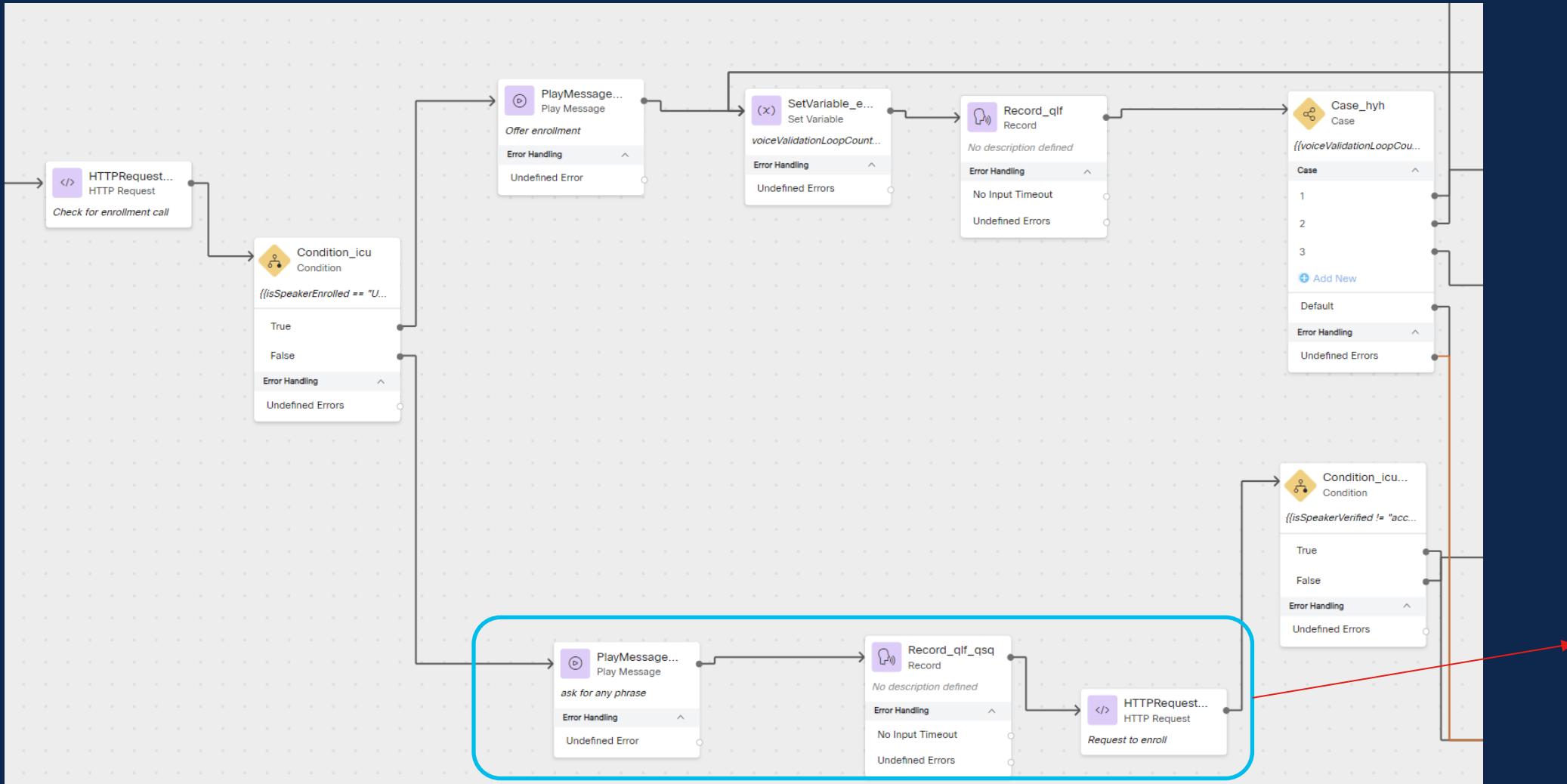
Enrollment  
Check

# Flow



Initial  
Enrollment

# Flow



Voice Print  
Validation



# Call Deflection with context



Marco & Carles

# What is Call Deflection?

- Call Deflection is reducing the number of calls that come into your contact center, help desk, or other areas of your business by diverting calls to alternate means of answering customer questions, such as self-service or digital channels.

# Why Call Deflection?

## 1. Digital channels

- Voice is an “Expensive” channel
- Agents can multitask Digital Channels

## 2. Self-service

- Chat bots allow for more complex self-service scenarios like product ordering, appointment management or flight booking

# How to Call Deflect?



### VirtualAgent

Virtual Agent Activity Settings

**Virtual Agent**

InsuranceAgent

**Make Prompts Interruptible**  
Allow the customer to interrupt the Virtual Agent to speak a new request and change the conversation.

**Override Default Language & Voice Settings**  
Enable this feature to configure custom language and voice settings for this virtual agent. If not enabled, the Global\_Language and Global\_VoiceName variables are referenced to define the default virtual agent settings.

**Input Language**  
en-US  
Select a language code that is supported by the virtual agent.

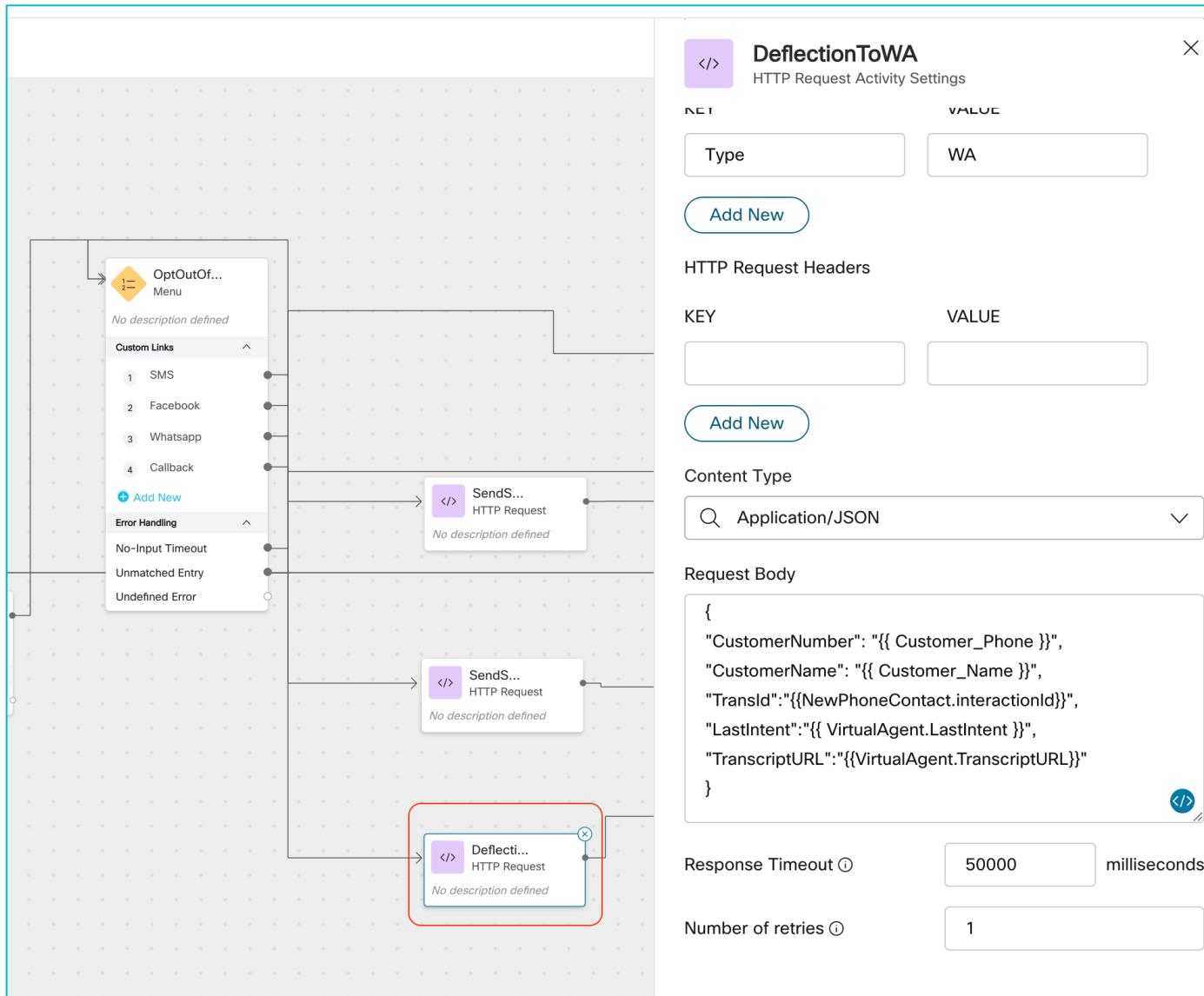
**Output Voice**  
en-GB-Wavenet-A  
By default, the virtual agent uses the Automatic Voice associated with the selected Input Language. If desired, select a different voice name.

**Variable Passing**

**Advanced Settings**

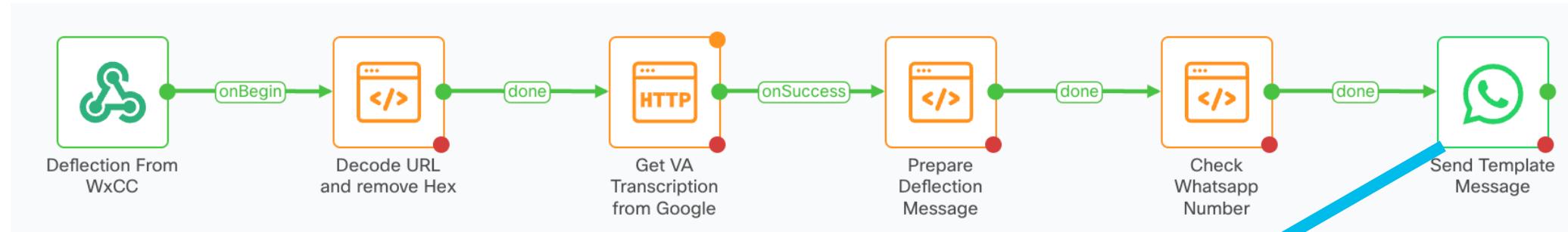
**Activity Output Variables**

VirtualAgent.LastIntent  
**VirtualAgent.TranscriptURL**  
VirtualAgent.ErrorCode



# What about context?





### Send Template Message

[Configuration](#) [Transition Actions \(Optional\)](#)

Parameter Type: Text  
Value: \${n2.inboundWebhook.CustomerName}

Body Parameters

Parameter Name: variable2  
Parameter Type: Text  
Value: \${botsummary}

Buttons

Button Text: Yes  
Payload: Deflection \${botsummary}

Button Text: No  
Payload: I need to book a flight

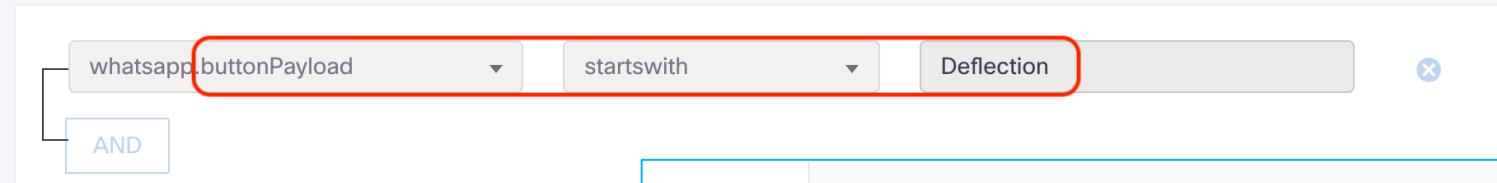
## Configure WhatsApp Event

[Help](#)[Configuration](#)    [Transition Actions \(Optional\)](#)

Choose an event and you can configure conditions on the event variables.

 Conditions

Flow will invoke only when these conditions are met

**Trigger Whatsapp Flow****Intercept deflection with chat bot and escalate directly**

**DEFLECTION**

Add new utterances which determine the intent, link entities which can be recognized in the intent and configure template keys which will be triggered for required entities.

**Intent name \***  
DEFLECTION

**All utterances**

Add utterance

Deflection You asked for a flight from **Rome** to **London** departing on **Wed, 15 Feb 2023 12:00:00 GMT**

**Slots**

Entity name	Required
Orig-city	<input type="checkbox"/>
Dest-city	<input type="checkbox"/>
Dept-date	<input type="checkbox"/>

**Context** ⓘ

Entry context

Add input context

Exit context

Add output context

**Response**

Agent handover



# Dynamic business hour announcement



**Siva Venkatesh Arcot**  
**(US)**  
siarcot@cisco.com

# Problem Statement

The current closure of ABC office for a holiday poses challenges for customers, the ABC office does not know exactly as the current message does not specify the reopening date. This lack of clarity makes it hard for customers to plan their inquiries, potentially causing disruptions.

Addressing this issue involves communicating the exact date of reopening to facilitate smooth and transparent customer interactions.

## Use Case

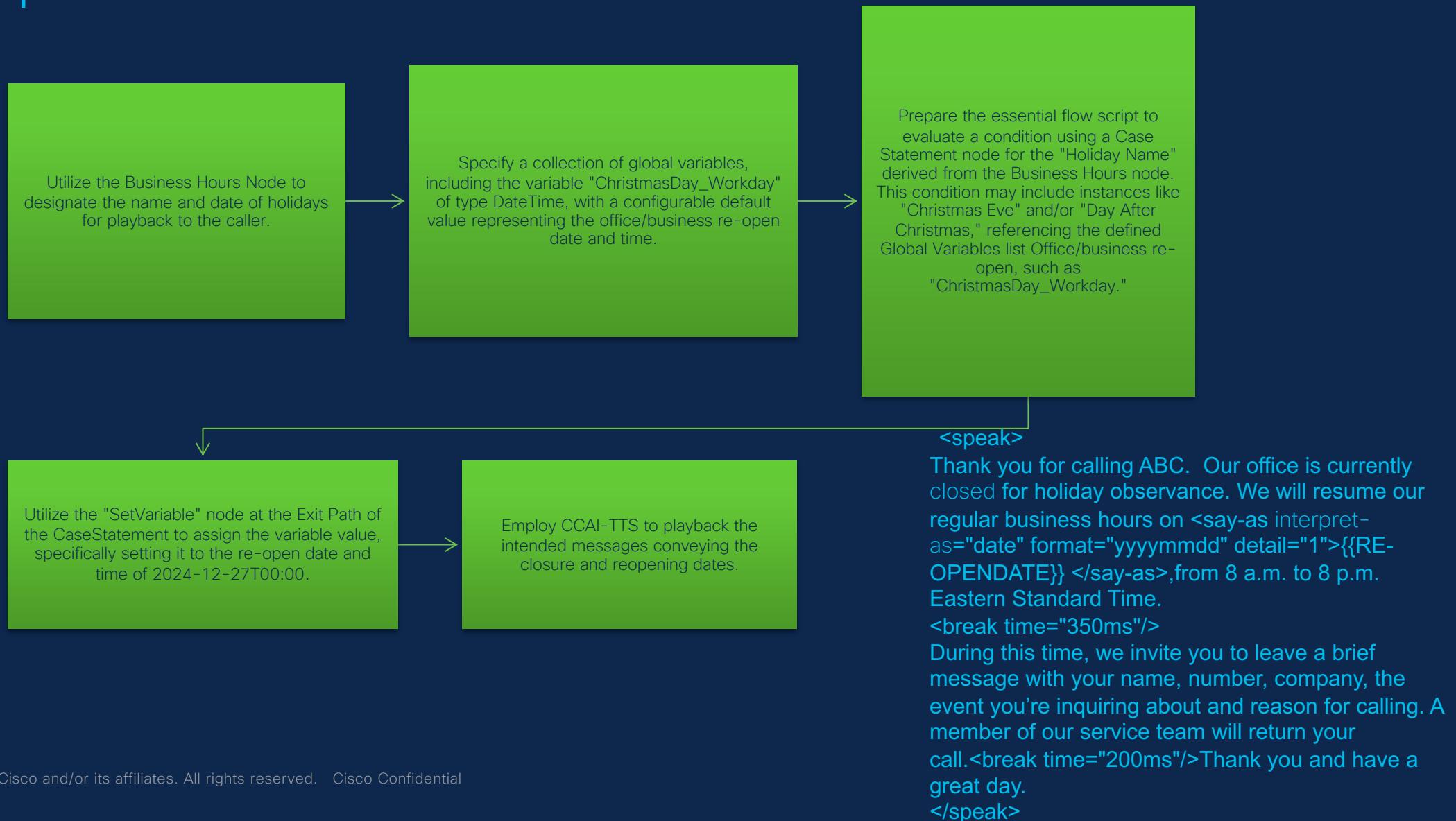
Along with the office closure messages during holiday. Contact Center should be able playback next business “re-open date”.

## Gotcha's

- BusinessHours Node does not offer the capability out-of-box
- Also, WxCCC has no direct integration to any calendar.

So How do you achieve this use case?? Next Slide pls.

# Approach to address the use case



# Use Case Scenario

Along with the office closure messages during holiday. Contact Center should be able playback next business “re-open date” dynamically.

## Pre-requisite:

- Use TTS
- BusinessHours (Holiday List) e.g., Christmas, DayAfterChristmas, Dayforme.
- Define Global Variables; corresponding Re-open day for “Christmas & DayAfterChristmas”, “WorkDate”  
e.g. Variable Name= **WorkDate** , Variable Type: **Date Time** , Default Value= **“Re-open Day”** Supported input formats; YYYY-MM-DD or YYYY-MM-DDThh:mm.

- Flow Variables as required to read and parse from “BusinessHours\_HolidayName”, “CaseStatements”, “SetVariable”

- TTS SSML sample

<speak>

Thank you for calling ABC. Our office is currently closed for holiday observance. We will resume our regular business hours on <say-as interpret-as="date" format="yyyymmdd" detail="1">{{REOPENDATE}} </say-as>, from 8 a.m. to 8 p.m. Eastern Standard Time.

<break time="350ms"/>

During this time, we invite you to leave a brief message with your name, number, company, the event you're inquiring about and reason for calling. A member of our service team will return your call.<break time="200ms"/>Thank you and have a great day.

</speak>

# Config's

The screenshot shows a Cisco Contact Center Express Flow Designer interface. On the left, a flow diagram is displayed with several activities:

- A green Start Flow icon.
- A Case Statement & Expressions activity with a condition `((true))`.
- Inside the Case block:
  - Case `DayforMe`: A Set Variable activity sets `WDHolder = {{WorkDate}}`.
  - Case `Observance of`: A Play Message activity plays a message.
  - Default: A Set Variable activity sets `WDHolder = {{Christmas}}`.
- After the Case block, there are two more Set Variable activities: one setting `WDHolder = {{WorkDate}}` and another setting `WDHolder = {{Christmas}}`.
- Play Message activities are connected to the Set Variable activities.
- Finally, a Disconnect Contact activity is shown.

A large red watermark "Demo Time" is overlaid on the flow diagram.

Two floating windows are shown on the right:

- SetVariable\_369** (Set Variable Activity Settings):
  - General Settings**: Activity Label is "SetVariable\_369".
  - Variable Settings**: Variable is "WDHolder".
- DayforMe Global Flow Properties**:
  - Custom Variables**: Describes Flow Variables as fully configurable local variables.
  - Flow Variables**: Shows "HolidayName" and "WDHolder" selected.
  - Predefined Variables**: Describes Predefined Variables added automatically through Global Variable definition or events and activities.
  - Global Variables**: Shows "WorkDate" and "Christmas" selected.
  - Text-to-Speech Message**: Contains the XML code for a TTS message about office closure during holidays.
  - Text-to-Speech Settings**: Notes that these settings apply to all TTS messages for this prompt.

Page footer: © 2020 Cisco and/or its affiliates. All rights reserved. Cisco 200



The bridge to possible