How to Move Web App Users from Spaces/Meetings in CMS

Contents

Introduction

This document describes the ability to move participants from one meeting to another meeting by Cisco Meeting Management (CMM). CMM administrator can move web app participants between meetings of either same or different call bridges.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Meeting Server (CMS) Basic Knowledge.
- CMM Basic Knowledge.
- CMS web app Basic Knowledge.

Components Used

The information in this document is based on these software and hardware versions:

- CMS Version 3.2.
- CMM Version 3.2.
- CMS web app Version 3.2.
- Web browser chrome 91.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

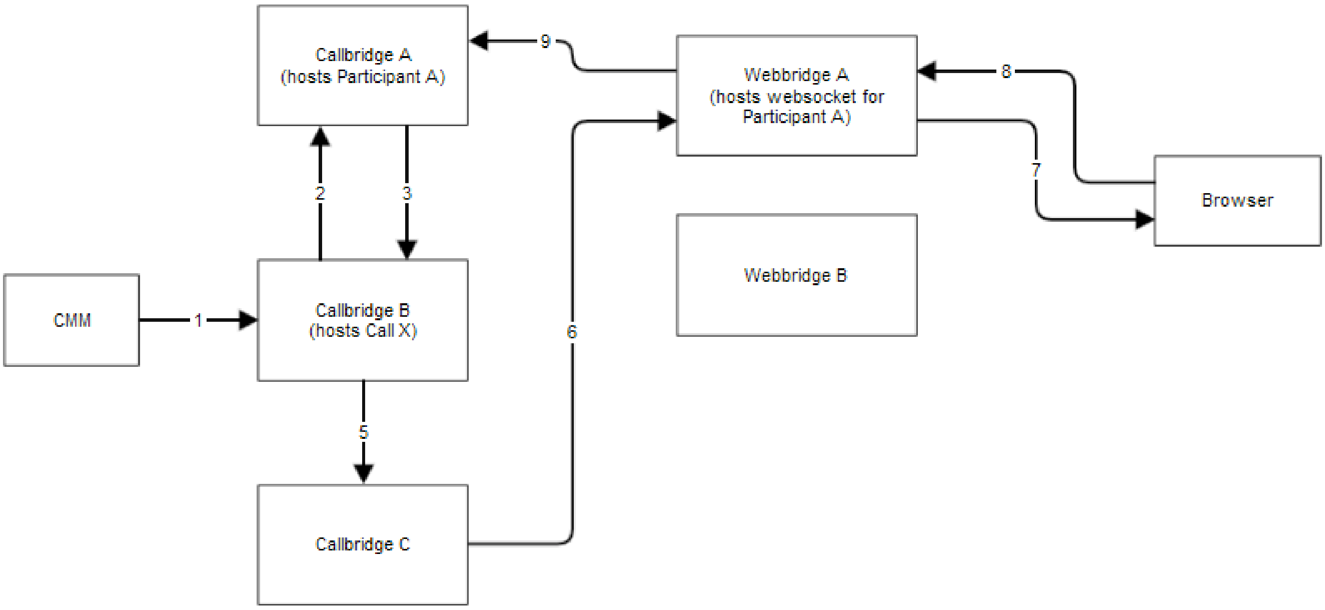
Configure

Background Information

The ability to move participants from one meeting to another meeting by CMM is originally featured in CMS 2.6 but with some restrictions, that is, Meeting App, web app, and Skype for Business (SfB) participants can't be moved. Start with CMS 3.2, CMM administrator can move web app participants between meetings of either same or different call bridges.

> **Note**: This feature does not mean web app participants can invoke a move of other participants. Previously when attempt to move web app participants, CMM would prevent this with an alert. That restriction is automatically detected by CMM is the meeting is hosted on CMS 3.2 and would be allowed to move.

Network Diagram



Configurations

Step 1. CMM makes Application Programing Interface (API) call to Callbridge B with method POST **/calls/<call_X_id>/participants/** with **"movedParticipant"=participant_A_guid**.

Step 2. Callbridge B sends participant move requests to Callbridge A.

Step 3. Callbridge A responds with the move request back to Callbridge B.

Step 4. Callbridge B does load balance and decides to place new participant to Callbridge C.

Step 5. Callbridge B sends request to Callbridge C to create new participant instance and participant. For guest, a new guest id is created. The new participant instance has a new JASON Web Tokens (JWT).

Step 6. Callbridge C sends API **move web socket** message over Call Bridge to Web Bridge (C2W) to Webbridge A.

Step 7. Webbridge A sends the **move web socket** message to the Webbridge Client (WC3) in the browser.

Step 8. WC3 in browser sends **end web socket** message to Webbridge A.

Step 9. Webbridge A forwards **end session** message to Callbridge A.

Step 10. Callbridge A destroys the participant instance and old JWT.

Step 11. WC3 client in browser authenticates in new web socket message to Webbridge A and use a new JWT.

Verify

Below is the sample log messages where the guest web participant is moved from **Space 1** (webapp.com) space to **Space 2** (webapp.com) space. To simplify the flow, the move to different space remains on the same call bridge **cbcms2** (cluster is load balanced).

First, the move flow begins with API POST **/calls/<call id>/participants**.

```
2021-03-04 15:50:03.915 Info API trace 42003: POST for "/api/v1/calls/ae778701-7fed-410c-b3e6-
c2860907a3f4/participants" (from 172.19.233.174) 2021-03-04 15:50:03.915 Info API trace 42003:
content data size 75, type "application/x-www-form-urlencoded": 2021-03-04 15:50:03.915 Info API
trace 42003: movedParticipant=26de0160-30b5-4d7b-8a05-304472a 2021-03-04 15:50:03.915 Info API
trace 42003: f284a& 2021-03-04 15:50:03.915 Info API trace 42003: needsActivation=false
```

Move participant to another call, first creates new guest account (**guest2316075499**).

```
2021-03-04 15:50:03.915 Info move participant operation: moving WC3 participant 26de0160-30b5-
4d7b-8a05-304472af284a (guest921953266) (homed on this callbridge) to call ae778701-7fed-410c-
b3e6-c2860907a3f4 2021-03-04 15:50:03.915 Info guest login request 0: credential storage
scheduled (queue length: 1) 2021-03-04 15:50:03.915 Info created guest account with user ID
"guest2316075499" 2021-03-04 15:50:03.915 Info guest login request 0: credential storage
executed 2021-03-04 15:50:03.915 Info guest login request 0: credential storage in progress
2021-03-04 15:50:03.921 Info guest login request 0: successfully stored credentials 2021-03-04
15:50:03.921 Info replace query for conference c3958a89-3007-4959-99e7-f6ea84609aac: response
from 'cbcms2' (priority: 0, load level: 0, conference is running: 1) 2021-03-04 15:50:03.921
Info replace query for conference c3958a89-3007-4959-99e7-f6ea84609aac: using local server
'cbcms2' (priority: 0, load level: 0, conference is running: 1) 2021-03-04 15:50:03.921 Info API
call leg dd2bc8c6-fa80-495f-9a20-1da19010cfab in call c0cc4e15-bb74-4af3-948b-672c9571c7fc (API
call ae778701-7fed-410c-b3e6-c2860907a3f4) 2021-03-04 15:50:03.922 Info 172.19.233.174: API user
"admin" created new participant dd2bc8c6-fa80-495f-9a20-1da19010cfab, call ae778701-7fed-410c-
b3e6-c2860907a3f4 2021-03-04 15:50:03.922 Info API trace 42003: sending 200 response, size 0
2021-03-04 15:50:03.922 Info API trace 42003: Location: /api/v1/participants/dd2bc8c6-fa80-495f-
9a20-1da19010cfab 2021-03-04 15:50:03.923 Info new session created for user "guest2316075499"
2021-03-04 15:50:03.923 Info instantiating user "guest2316075499"
```

Delete the old user **guest921953266** and tear down the previous call, **call 19**.

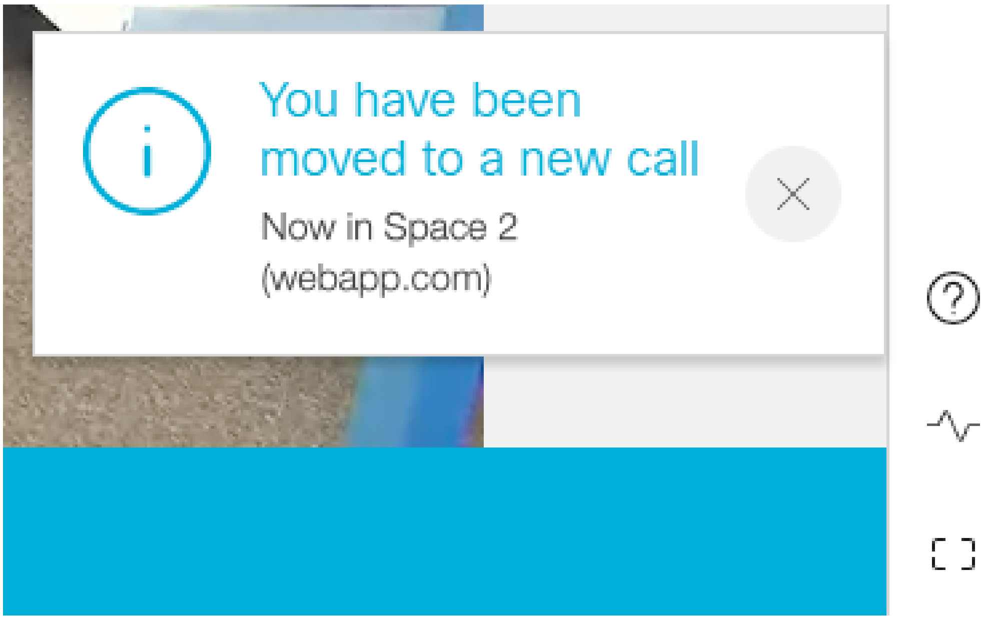
```
2021-03-04 15:50:03.947 Info user "guest921953266": deactivating due to session resource
teardown 2021-03-04 15:50:03.948 Info call 19: tearing down ("guest921953266" conference media)
2021-03-04 15:50:03.948 Info participant "guest921953266" left space 89eae70d-5b67-41fc-97f7-
```

```
38a655fa6467 (Space 1 (webapp.com)) 2021-03-04 15:50:03.948 Info call 19: destroying API call
leg 26de0160-30b5-4d7b-8a05-304472af284a 2021-03-04 15:50:03.948 Info removing guest account
'guest921953266' (name 'User X') on call drop 2021-03-04 15:50:03.948 Info destroying guest
account with user ID "guest921953266"
```

Successfully set up media session for the new call, **call 20**.

```
2021-03-04 15:50:04.106 Info call 20: allocated for guest2316075499 / "User X" conference
participation (Chrome) 2021-03-04 15:50:04.106 Info call 20: removing h264 from video codec
bitmask, because it's Chrome web client and we're using a compatibility profile 2021-03-04
15:50:04.106 Info call 20: configured - API call leg dd2bc8c6-fa80-495f-9a20-1da19010cfab 2021-
03-04 15:50:04.107 Info call 20: setting up combined RTP session for DTLS (combined media and
control) 2021-03-04 15:50:04.108 Info participant "guest2316075499" joined space 59b9e43e-b277-
4d33-a244-e896d20f2049 (Space 2 (webapp.com)) 2021-03-04 15:50:04.108 Info participant
"guest2316075499" (dd2bc8c6-fa80-495f-9a20-1da19010cfab) joined conference c0cc4e15-bb74-4af3-
948b-672c9571c7fc via WB3
```

When the web app receives a move request, it disconnects the current call, and then starts the join process again with the new JWT. After the move, the participant sees the message **You have been moved to a new call** on the lower right corner which indicates that the call is now in a new meeting as shown in the next image. The text after the **Now in** message, is the space name in this case **Space 2**.



You have been moved to a new call

Now in Space 2 (webapp.com)

Some local web app meeting state such as mute and layout are carried over from the previous call. For example, if participant mutes locally then it remains muted in the new call.

The next features are not carried over to the new call:

- **Presentation** – When participant is moved, active presentation is dropped. In the new meeting after the move, the participant does not share.
- **Chat messages** – Previous chat messages are removed from the chat and does not transfer over to the new meeting.

Troubleshoot

Issue: Web app participant doesn’t get moved.

It could mean many things:

1. Nothing happened. The call is still connected to the first call.
2. Dropped but not reconnect. The call is dropped but doesn’t connect to the second call.
3. Connect to the wrong meeting.

For scenario **a. Nothing happened**:

- Ensure the call bridge receives the request to move from CMM. See the CMS Log Messages for specific keyword like **move participant operation**. If CMS doesn’t receive API from CMM then do basic troubleshooting between CMM and CMS include enabled API trace on both side, Domain Name Service (DNS) checks, connectivity check.
- See if the **canMove** parameter in **/participants/<participant id>** or **/callLegs/<callLeg id>** is set to **true**.

For scenario **b. Dropped but not reconnect**:

- Ensure disconnect is due to a move, that is, look for **move participant operation** in the log.
- In CMS logs, look for resource **errors/blockage** on the call bridge that could prevent the participant creation process to take place.
- Does the participant have permission to join the new cospace?
- Is there error with JWT?
- Try to manually join the meeting.

For scenario **c. Connect to the wrong meeting**:

In the Hyper Text Transport Protocol (HTTP) Archive Format (HAR) file, look at the web socket of the first call, the access method data for the POST **/api/call/session/move** shows the numeric Id that is used to connect to the new call. Ensure this numeric Id is the one that is the intended meeting.