Possibilities

#CiscoLive

Troubleshooting the Cisco Meeting Server

2020 Update

Vernon Depee
Technical Leader – Customer Experience

DGTL-BRKCOL-3110

cisco Live!

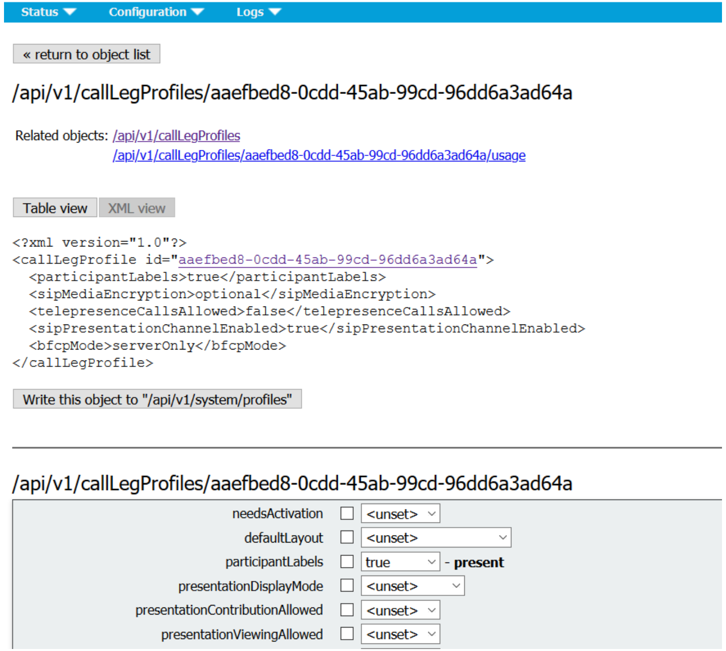
Agenda

- Chapter 1 – Introduction/Agenda
- Chapter 2 - API Explorer
- Chapter 3 - WebApp
- Chapter 4 - Recorder/Streamer

Chapter 2: API Explorer

API Explorer

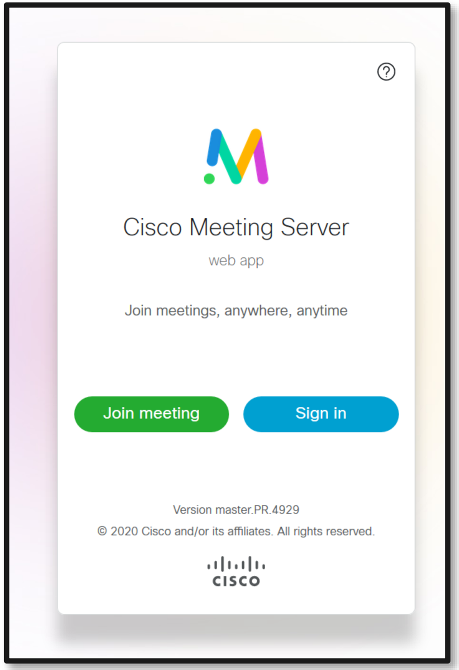
- New feature in CMS 2.9
- API tools no longer required
- In WebAdmin navigate to configuration > API
- Data available as table or xml



Chapter 3: WebApp

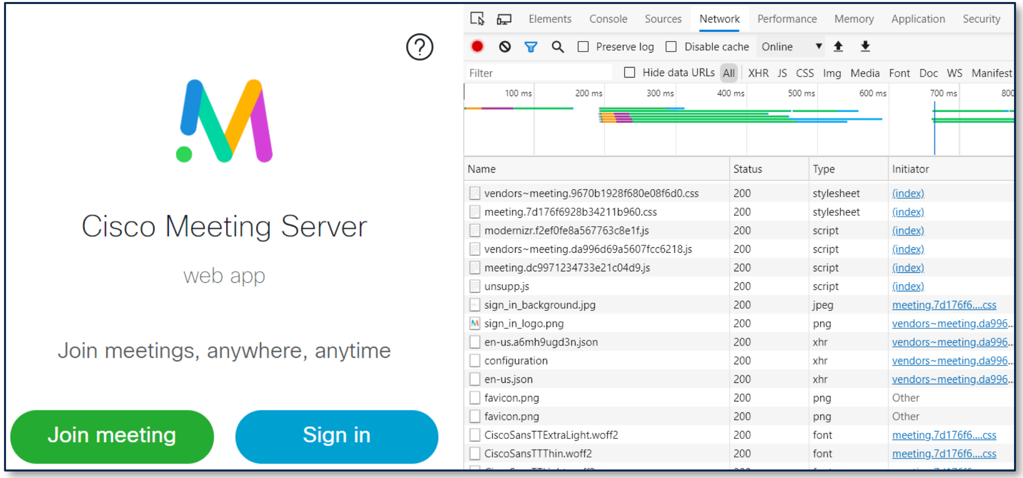
WebApp

- With the removal of the XMPP protocol in CMS 3.0 we've moved to a new protocol – C2W (CallBridge to WebBridge).
- WebBridge was effectively rebuilt form the ground up to provide a far more scalable solution.
- A lot of serviceability work has gone into Web App to make it as easy to troubleshoot as possible.



Client-Side Logging

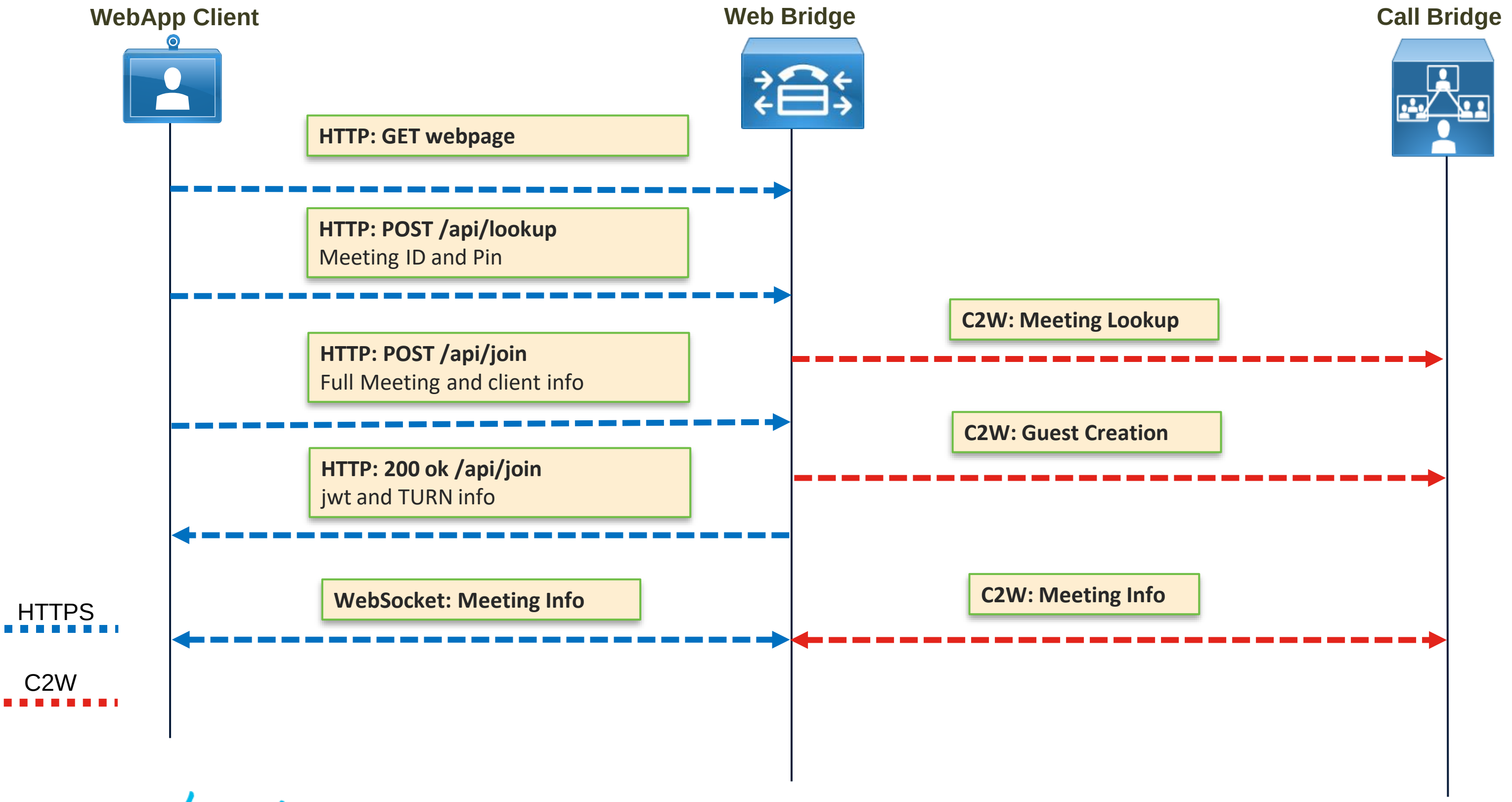
- Most browsers have a way to view network logs.
    - On Firefox press CTRL-SHIFT-E and navigate to the network tab
    - On Chrome/Chromium Edge press CRTL-SHIFT-I and navigate to the network tab
- This will show HTTP(s) communication to and from the browser and include WebSocket messages.
    - These messages can be downloaded as a log (HAR format) from the network tab.
    - At present HAR files from Firefox do not include WebSocket messages, so the use of Chrome/Chromium for log collection is preferred.



Flow - Guest

1. Client sends an HTTP GET for the web interface.
2. Client sends Meeting ID and Passcode as a POST to /api/lookup.
3. The WebBridge Communicates with the CallBridge to validate the information.
   - If Incorrect the POST will receive a response of “400 Bad Request”.
     - The CMS will have a WARNING level entry of “guest lookup request <XXX> failed: invalid passcode”
   - If the meeting information is correct the client will receive a 200 OK and be prompted to join the meeting.
4. When the client clicks join they will send one last POST message with the meeting information, their username, the type of browser they have, and if trace logging is enabled.
5. The response to the join POST will be a 200 ok that includes a jwt and TURN server information.
   - Please note that TURN server information sent to the client will be decipherable by the end client. This is an important reason why TURN credentials should be considered insecure and the same username/passwords used for TURN should not be used for anything secure!
6. The client will open a WebSocket connection with the WebBridge and authenticate using the jwt. From here on out all meeting information will flow through this WebSocket.

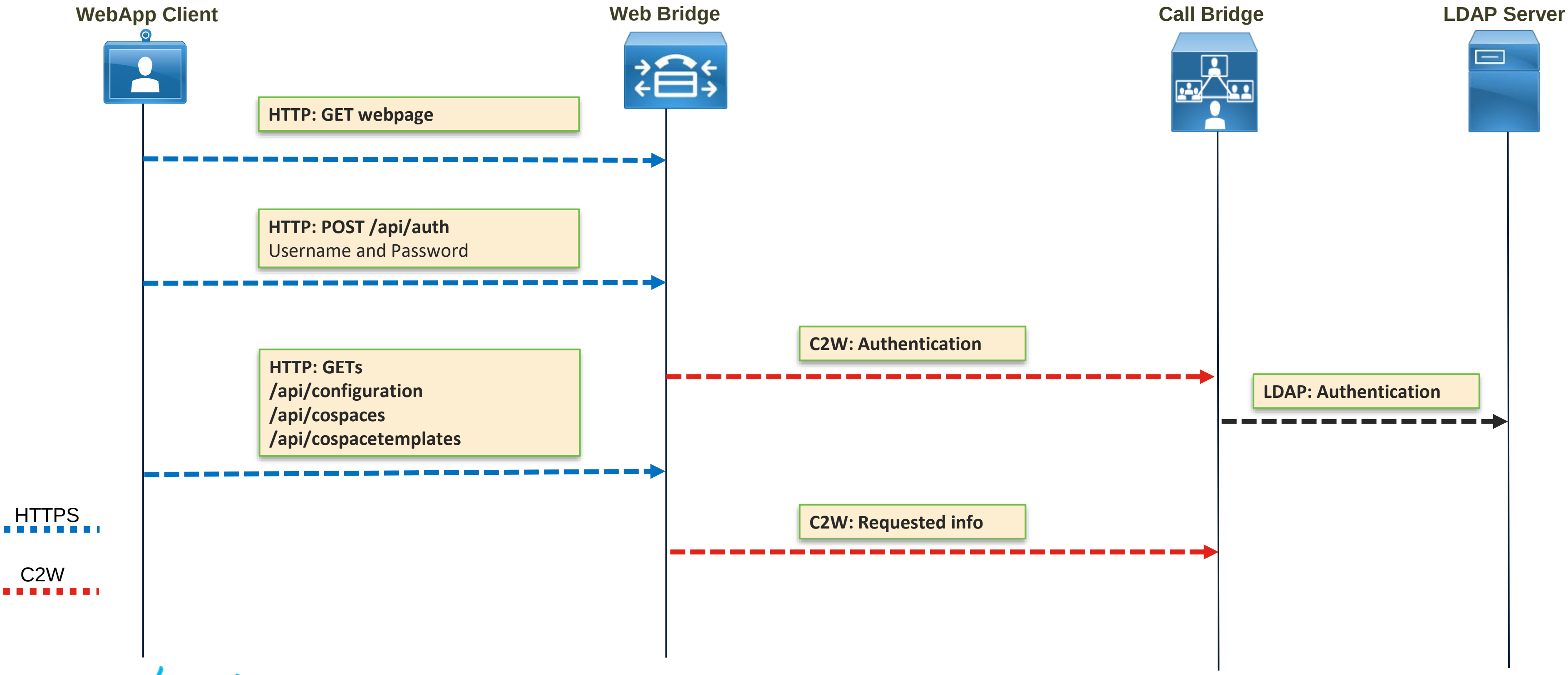
Guest Meeting Join Flow



Flow – Sign In

1. Client sends an HTTP GET for the web interface.
2. Client sends username and password as a POST to /api/auth.
3. The WebBridge Communicates with the CallBridge to validate the information.
   - CallBridge validates credentials against LDAP.
   - If Incorrect the POST will receive a response of “401 Unauthorized”.
     - The CMS will have an INFO level entry of:
     - LDAP authorisation failed for user 'administrator@pod1.local'
     - LDAP failure 49 (invalidCredentials) server diagnostics message: 80090308: LdapErr: DSID-0C09041C, comment: AcceptSecurityContext error, data 52e, v4563
     - unsuccessful login request from administrator@pod1.local
     - If the meeting information is correct the client will receive a 200 OK and be prompted to join the meeting.
4. From here clients download the configuration, list of spaces, and space templates.
5. If they join a meeting the flow is identical to that of a guest join, but they send a guid instead of space information when they POST to /api/join.

Authenticated User Login Flow



Problem Report

- Clients can generate problem reports from the WebApp. The log information is sent via WebSocket from the client and combined with the server's local information. Logs can be downloaded from the WebAdmin.
- Inside of the problem report you will find:
  - messages regarding the call that generated the report
  - SDP
  - RTP statistics
  - Logs from the client's browser

Call diagnostic logs

☐	Log name	Time	Download link
☐	"guest2725741725" diagnostics	2020-08-20 19:09:24.593460 +0000	[download]

Delete | Delete all

Server-Side Logging

- Logging can also be collected from the server side by simply adding trace=true to the URL in the browser.
- If no parameters are already being passed via the URL simply add the following to the end of the URL:
  - ?trace=true
- If parameters are already being passed via the URL (there is already a “?” symbol) simply add the following to the end of the URL:
  - &trace=true
- This must be done on a per-client basis as we want this logging disabled by default and only enabled on clients who are troubleshooting.
- To prevent logs overflowing the WebSocket messages are truncated.

Chapter 4: Recorder/Streamer

SIP Streamer/Recorder

- With CMS version 3.0 we introduced new SIP based recorder and streamer services.
- The removal of XMPP from this flow greatly simplifies things
- sipRecorderUri and sipStreamerURI should be set in the callprofile in the API. This is the URI the CallBridge will dial out to.
- The alias portion of the URI is unimportant – it can be anything, but the domain portion must match an outbound dialplan rule that points to the streamer/recorder server.

Logging

- Recording and Streaming should be running on different servers than the CallBridge and WebAdmin, so logging should be enabled via the CLI.
  - recorder sip trace <1m|10m|30m|24h|on|off>
  - streamer sip trace <1m|10m|30m|24h|on|off>
- Logging will be written to syslog or available to download via SFTP.

Call Routing

- Create outbound rule for SIP domain specified for Recorder/Streamer
- Be sure the SIP proxy matches the listening port for the Recorder/Streamer and the Encryption also matches.

« return to object list

/api/v1/callProfiles/6566153e-bf11-48eb-8870-ba0a7806dbcb

Related objects: /api/v1/callProfiles

Table view | XML view

Object configuration	
recordingMode	automatic
streamingMode	disabled
sipRecorderUri	recorder@recorder.local

Write this object to "/api/v1/system/profiles"

Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	
recorder.local	192.168.1.121:5070		<use local contact domain>	Standard SIP	Stop	1	Unencrypted	no	[edit]

Recorder/Streamer Startup

- The recorder/streamer must be able to bind to the SIP port specified
  - It's important that no other service is using the ports you want to use for SIP.
  - If you would like to use 5060 or 5061 it's important that the CallBridge was never configured on this server. If it was you will likely need to factory reset and reconfigure without ever touching the CallBridge configuration.

```
cms> recorder enable
SUCCESS: Key and certificate pair match
FAILURE: Ports in use by other services
```

- The recorder must be able to communicate with the NFS
  - Upon startup the recorder will establish a connection with NFS. If this fails the recorder will not start.

```
recorder: Failed to mount NFS storage
```

Thank you

Possibilities

#CiscoLive