Cisco Meeting Server

Cisco Meeting Server Release 3.12

Scalability & Resilience Server Deployment Guide

June 24, 2026

Contents

What's new 11

1 Introduction 12
1.1 Supported Apps for joining Meeting Server hosted conferences 14
1.2 Using the Cisco Expressway-E as the Edge device in Meeting Server deployments 14
1.3 Using the Cisco Expressway-C with the Meeting Server in the core network 16
1.3.1 Expressway support of Call Bridge Groups 16
1.3.2 Using the Cisco Expressway H.323 gateway component 17
1.4 Using Meeting Server as the Edge device in Meeting Server deployments 17
1.5 How to use this guide 18
1.5.1 Commands 20
1.6 Configuring the Meeting Server 20
1.6.1 MMP and API Interfaces 21
1.6.2 New tools to ease configuring Meeting Server 21
1.7 Meeting Server licensing 24
1.7.1 Licensed features 24
1.7.2 Smart Licensing 25
1.7.3 Smart Account and Virtual Account information 26

2 General concepts for deployment 27
2.1 Web Admin 28
2.2 Call Bridge 29
2.3 Database 29
2.4 Configuring Web Bridge 3 29
2.5 TURN server 29
2.6 Meeting Server Edge 31
2.7 Recording meetings 31
2.7.1 License keys for recording 31
2.8 Streaming meetings 31
2.8.1 License keys for streaming 32
2.9 Hosting branding files locally 32
2.10 On screen messaging 32
2.11 SIP trunks and routing 33
2.12 Support for Lync and Skype for Business 33
2.12.1 Support for Lync and Skype for Business clients 33
2.12.2 Support for Dual Homed Conferencing 34

2.13 Web Scheduler 34
2.13.1 Scheduler in the web app UI 35
2.14 General points about scalability and resilience 35
2.14.1 Example using multiple “combined” servers 36
2.14.2 Database clustering 38

3 Prerequisites 41
3.1 Prerequisites for installing and configuring the Meeting Server 41
3.1.1 DNS configuration 41
3.1.2 Security certificates 41
3.1.3 Firewall configuration 41
3.1.4 Syslog server 42
3.1.5 Network Time Protocol server 43
3.1.6 Call Detail Record support 43
3.1.7 Host name 44
3.1.8 Other requirements 44
3.1.9 Specific prerequisites for a virtualized deployment 45
3.2 Meeting Server Edge hardware configuration 45
3.2.1 Edge server configurations 45
3.2.2 Deployment considerations 47
3.3 Network Planning for Meeting Server Edge 48
3.3.1 Technical description 48
3.3.2 Network planning 49
3.3.3 Deploying Meeting Server web edge 53

4 Configuring the MMP 55
4.1 Creating and managing MMP and Web Admin interface user accounts 55
4.2 Upgrading software 55
4.3 Configuring the Call Bridge listening interface 57
4.4 Configuring the Web Admin interface for HTTPS access 58
4.5 Stage Edge Server instances 59
4.6 Configuring Web Bridge 3 60
4.6.1 Useful information to help configure Web Bridge 3 60
4.6.2 Enabling the Web Bridge 3 Service 62
4.6.3 Configuring Call bridge C2W connections 63
4.6.4 Configure Call Bridge with Web Bridge Addresses 64
4.7 Configuring the Email server for Scheduler 65

4.7.1 Scheduler detailed logging 71
4.8 Configuring the TURN Server 72
4.8.1 Enable the TURN Service 72
4.8.2 Configure Call Bridge with TURN Addresses 74
4.9 LDAP authentication for MMP users 76

5 Configuring the Databases 77
5.1 Database on a Separate Server 78
5.1.1 Requirements for a database on a separate server 78
5.1.2 Deploying a database on a separate server 78
5.2 Deploying and validating Certificates on the Database and Call Bridge Servers 78
5.3 Selecting the Primary Database for a Cluster 80
5.4 Attaching other Database Instances to the Database Cluster 82
5.5 Connecting Remote Call Bridges to the Database Cluster 84
5.6 Upgrading the database schema 85
5.6.1 Verify the database schema upgrade 86
5.7 Further information on database clusters 86

6 Deploying the Call Bridges 87
6.1 Setting up the Call Bridges' certificates 87
6.2 Setting up the Call Bridges 87
6.3 Clustering Call Bridges 88
6.3.1 Call Bridge cluster validation 90
6.3.2 Using DTMF sequences in clustered Call Bridge deployments 91
6.4 Dial Plan Information 91
6.4.1 Setting up dial plan rules for Inter-peer calls 92
6.4.2 Examples 93
6.5 Load Balancing calls across Meeting Servers 95
6.5.1 Call Bridge Groups 95
6.5.2 Configuring Call Bridges for load balancing incoming calls 96
6.5.3 Load balancing outbound SIP calls 99
6.5.4 Load balancing web app calls 101
6.6 Lync Account Information 102
6.7 Choosing Call Bridge mode to connect participants to Lync conferences 102

7 Deploying Web Bridge 3 107
7.1 Deploying multiple Web Bridge 3s 107
7.2 Setting up the Web Bridge 3 certificates 107

7.3 Setting up the Web Bridge 3 107
7.3.1 How to create and apply a web bridge profile example 108
7.4 Web Bridge 3 call flow 111
7.5 Web app information 112
7.6 Enabling HTTP redirect and Web Bridge 3 112

8 Deploying the TURN Servers 113
8.1 Configuring TURN servers 113

9 web app in distributed deployments 115

10 Dial plan configuration – overview 120
10.1 Introduction 120
10.2 Dial plan rules for incoming calls and outbound calls. 122
10.2.1 /outboundDialPlanRules 122
10.2.2 /inboundDialPlanRules 125
10.2.3 /forwardingDialPlanRules 126
10.3 Dial Transforms 127

11 Dial plan configuration – SIP endpoints 130
11.1 Introduction 130
11.2 SIP video endpoints dialing a meeting hosted on clustered Meeting Servers 130
11.2.1 SIP call control configuration 130
11.2.2 Meeting Server configuration 131
11.3 Media encryption for SIP calls 133
11.4 Enabling TIP support 133
11.5 IVR configuration 134
11.6 Next steps 135

12 Dial plan configuration – integrating Lync/Skype for Business 136
12.1 Lync clients dialing into a call on clustered Meeting Servers 136
12.1.1 Lync Front End server configuration 137
12.1.2 Adding a dial plan rule to clustered Meeting Servers 139
12.2 Integrating SIP endpoints and Lync clients 140
12.3 Adding calls between Lync clients and SIP video endpoints 141
12.3.1 Lync Front End server configuration 142
12.3.2 VCS configuration 142
12.3.3 Meeting Server configuration 142
12.4 Integrating Lync using Lync Edge service 144

12.4.1 Lync Edge call flow 145
12.4.2 Configuration on Meeting Server to use Lync Edge 146
12.5 Controlling the bandwidth for sharing content on Microsoft Lync and Skype for Business calls 148
12.6 Direct Lync federation 149
12.7 Calling into scheduled Lync meetings directly and via IVR 150

13 Office 365 Dual Homed Experience with OBTP Scheduling 153
13.1 Overview 153
13.2 Configuration 153
13.3 In-conference experience 153

14 Settings for Web Bridge 3 155
14.1 Web Bridge 3 connections 155
14.1.1 Web Bridge 3 call flow 155
14.2 Web Bridge 3 settings 157
14.2.1 How to create and apply a web bridge profile example 157

15 Recording and Streaming meetings 160
15.1 Feature benefits of the new internal SIP recorder and streamer 160
15.2 Points to note when implementing the new internal SIP recorder and streamer 160
15.3 Recording overview 161
15.3.1 Third-party external SIP recorder support 162
15.3.2 Meeting Server internal SIP recorder component support 162
15.3.3 Deploying a Recorder / Streamer for Scalability and Resiliency 164
15.4 Example of deploying the new internal SIP recorder component on a VM server 166
15.5 Configuring an external third-party SIP recorder 169
15.6 Finding out recording status 169
15.7 Recording indicator for dual homed conferences 170
15.8 Recording with Vbrick 170
15.8.1 Prerequisites for the Meeting Server 171
15.8.2 Configuring the Meeting Server to work with Vbrick 172
15.9 Streaming meetings 174
15.10 Deploying the new SIP streamer component on a VM server 175
15.10.1 Known Limitations 178

16 LDAP configuration 179
16.1 Why use LDAP? 179

16.2 Meeting Server settings 180
16.3 Example 183
16.4 More information on LDAP field mappings 184
16.5 Enforcing passcode protection for non-member access to all user spaces 186

17 Single Sign On (SSO) for Cisco Meeting Server web app 187
17.1 Configuring SSO for use on Meeting Server web app 187
17.1.1 Example 1 config.json file 191
17.1.2 Example 2 Simple service provider metadata file. 192
17.1.3 Example 3 Comprehensive service provider metadata file. 192

18 Support for ActiveControl 194
18.1 ActiveControl on the Meeting Server 194
18.2 Limitations 194
18.3 Overview on ActiveControl and the iX protocol 194
18.4 Disabling UDT within SIP calls 195
18.5 Enabling iX support in Cisco Unified Communications Manager 195
18.6 Filtering iX in Cisco VCS 196
18.7 iX troubleshooting 197

19 Scheduler - Deployment 198
19.1 Deploying the Scheduler 199
19.1.1 Scheduler detailed logging 206

20 Additional security considerations & QoS 209
20.1 Common Access Card (CAC) integration 209
20.2 Online Certificate Status Protocol (OCSP) 209
20.3 FIPS 209
20.4 TLS certificate verification 210
20.5 User controls 210
20.6 Firewall rules 210
20.7 DSCP 211
20.8 Verifying SSH fingerprints 211

21 Diagnostic tools to help Cisco Support troubleshoot issues 213
21.1 SIP Tracing 213
21.2 Log bundle 213
21.3 Ability to generate a keyframe for a specific call leg 214
21.4 Reporting registered media modules in syslog 215

22 Additional licensing information 216
22.1 Licensing 216
22.1.1 How Smart licenses work in Meeting Server – overview 216
22.1.2 Expired license feature enforcement actions 218
22.1.3 How to retrieve licensing information (Smart Licensing) 219
22.1.4 Smart Licensing registration process 219
22.1.5 Multiparty licensing 220
22.1.6 Assigning Personal Multiparty licenses to users 221
22.1.7 How Cisco Multiparty licenses are assigned 222
22.1.8 Determining Cisco Multiparty licensing usage 222
22.1.9 Calculating SMP Plus license usage 223
22.1.10 Retrieving license usage snapshots from a Meeting Server 224
22.1.11 License reporting 224
22.1.12 Legacy licensing file method 224

23 Obtaining information on hosted conferences 226
23.1 Call Detail Records (CDRs) 226
23.2 Events 226

Appendix A DNS records needed for the deployment 228

Appendix B Ports required for the deployment 231
B.1 Configuring the Meeting Server 232
B.2 Connecting services 232
B.3 Using Meeting Server components 233
B.4 Additional ports required for Scalability and Resilience 236
B.5 Ports open on loopback 237

Appendix C Sharing Call Bridge licenses within a cluster 239
C.1 Registering your Cisco Meeting Server activation PAK codes 239
C.1.1 Sharing feature licenses across the cluster 240
C.2 Adding licenses to an existing Call Bridge cluster 245

Appendix D Unclustering 246
D.1 Unclustering Call Bridges 246

Appendix E Call capacities by Cisco Meeting Server platform 247
E.1 Cisco Meeting Server web app call capacities 248
E.1.1 Cisco Meeting Server web app call capacities – external calling 248

E.1.2 Cisco Meeting Server web app capacities – mixed (internal + external) calling 249

Appendix F Activation key for unencrypted SIP media 250
F.1 Unencrypted SIP media mode 250
F.2 Determining the Call Bridge media mode 251

Appendix G Dual Homed Conferencing 252
G.1 Overview 252
G.2 Consistent meeting experience in dual homed conferences 252
G.2.1 Summary of user experiences 253
G.3 Mute/unmute meeting controls in dual homed conferences 254
G.4 Configuring the Dual Homed Lync functionality 255
G.4.1 Troubleshooting 255

Appendix H Using TURN servers behind NAT 257
H.1 Identifying candidates 257
H.1.1 Host candidate 257
H.1.2 Server Reflexive candidate 257
H.1.3 Relay candidate 258
H.2 Checking connectivity 260
H.3 NAT in front of the TURN server 261

Appendix I Web Admin Interface – Configuration menu options 264
I.1 General 264
I.2 Active Directory 264
I.3 Call settings 265
I.4 Outbound calls and Incoming calls 266
I.5 CDR settings 266
I.6 Spaces 267
I.7 Cluster 267
I.8 API 267

Appendix J API Examples 269
J.1 Creating an Outbound Dial Plan Rule for a Specific Call Bridge in a Cluster 269
J.2 Setting up a Web Bridge on the Meeting Server 270
J.3 Creating Web Bridge Customization on a Call Bridge 271
J.4 Setting up the TURN Server and connecting to the Call Bridge 272
J.5 Creating a space and adding members 272

J.5.1 Adding Members to the space 273
J.6 Creating Call Leg Profiles 274
J.7 Applying Access Methods to a space 275

Cisco Legal Information 277

Cisco Trademark 278

What's new

Version	Change
April 03, 2026	Updated for version 3.12. Updated the document for Upgrading the database schema.
October 31, 2025	Updated for version 3.12.

1 Introduction

The Cisco Meeting Server software can be hosted on specific servers based on Cisco Unified Computing Server (UCS) technology or on a specification-based VM server. Cisco Meeting Server is referred to as the Meeting Server throughout this document.

Note: Cisco Meeting Server software version 3.0 onwards does not support X-Series servers.

This guide covers the Meeting Server deployed as a scalable and resilient solution. It discusses the concepts, requirements and how to deploy this type of architecture. By deploying more than one host server, you can configure:

- Several components of the same type to work as one resilient “unit”; for example if Call Bridges are clustered then if one Call Bridge goes down, meetings can be hosted on the other(s).
- Scalability (increased capacity); for example, one meeting can be hosted across multiple Call Bridges if one does not have enough capacity to host all the participants. (As a general principle, when possible, each meeting is hosted on a single Call Bridge).
- Efficiency; the Meeting Server decides which components to use to provide effective and efficient meetings; for example, participants calling into a meeting from different locations can use different components while keeping the user experience of a simpler deployment .
- Each Meeting Server can host all the components (" combined deployment") or be an Edge or Core server (part of a “split deployment”).

Note: All of the Meeting Servers in the deployment must run the same version of software.

Note: Meeting Server 3.0 introduced a mandatory requirement to have Cisco Meeting Management 3.0 (or later). Meeting Management handles the product registration and interaction with your Smart Account (if set up) for Smart Licensing support. For more details on Smart Licensing, see Cisco licensing.

Expressway (Large OVA or CE1200) is the recommended solution for deployments with medium web app scale requirements (i.e. 800 calls or less). Expressway (Medium OVA) is the recommended solution for deployments with small web app scale requirements (i.e. 200 calls or less). However, for deployments that need larger web app scale, from version 3.1 we recommend Cisco Meeting Server web edge as the required solution.

For more information on deploying the Meeting Server web edge solution, see General concepts for deployment.

Figure 1 shows the components available in a combined server deployment. Note that the Recorder, Uploader and Streamer components should be enabled on a separate server to that

hosting the meetings. The Cisco Meeting Server 2000 schematic assumes that Cisco Expressway provides the TURN services.

Figure 1: Combined server deployment

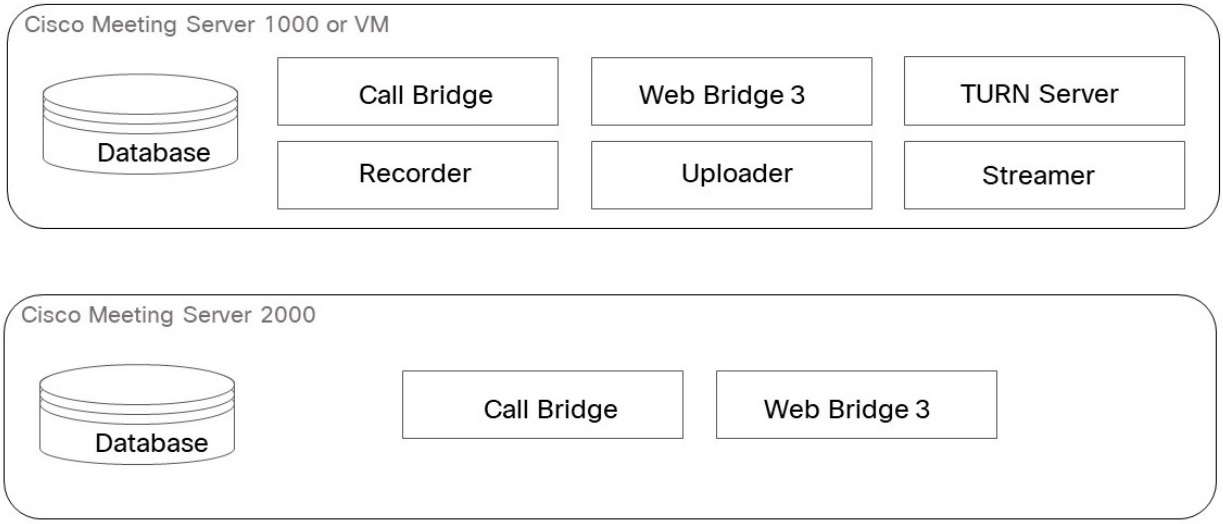
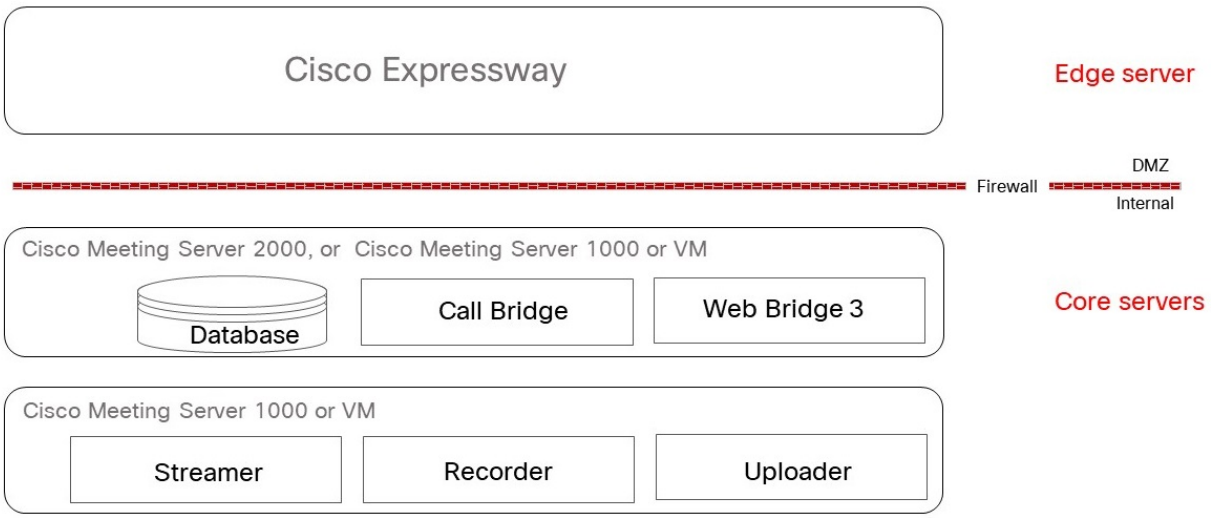


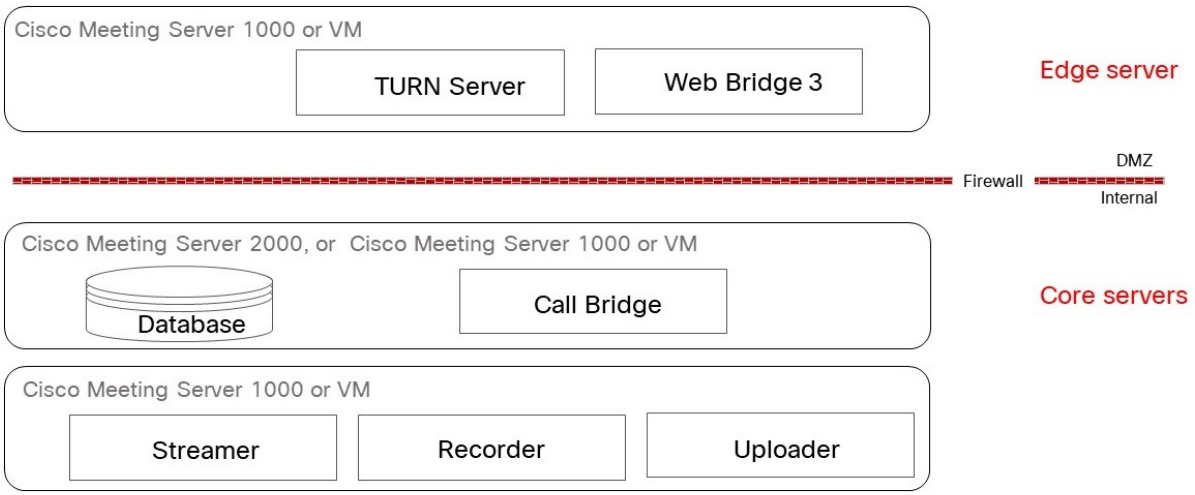
Figure 2 and Figure 3 show the components available in a split server deployment. Figure 2 shows the TURN services provided by Cisco Expressway, and Figure 3 shows the TURN services provided by the Meeting Server using the TURN server component in the Meeting Server. Note that Web Bridge 3 moves to the Edge server if the TURN server component in the Meeting Server is enabled. The Recorder, Uploader and Streamer components should be enabled on a separate core server to that hosting the meetings.

Figure 2: Split server deployment with Cisco Expressway providing TURN services



When used as an edge server, the Meeting Server uses its existing TURN server and web app components (and not the Call Bridge component), as shown in Figure 3.

Figure 3: Split server deployment with Cisco Meeting Server providing TURN services



Not all of these components need to be configured, you only need to configure the components that are appropriate to your deployment. This is discussed in General concepts for deployment.

Important Information: From version 3.0, the XMPP server, Load Balancer, SIP Edge and H.323 Gateway components have been removed from the Cisco Meeting Server software. In addition, new SIP Recorder and Streamer components replace the previous XMPP client versions of the Recorder and Streamer, which have been removed from the server software. The TURN server remains in version 3.0 software and may be used to connect the browser based Cisco Meeting Server web app to Meeting Server conferences. Both native and browser based Cisco Meeting App clients are not supported in version 3.0.

1.1 Supported Apps for joining Meeting Server hosted conferences

The Cisco Meeting Server web app and Cisco Jabber are the supported apps to join Meeting Server hosted conferences. This is in addition to SIP endpoints, and Lync/Skype for Business clients in dual homed conferences.

1.2 Using the Cisco Expressway-E as the Edge device in Meeting Server deployments

Expressway (Large OVA or CE1200) is the recommended solution for deployments with medium web app scale requirements (i.e. 800 calls or less). Expressway (Medium OVA) is the recommended solution for deployments with small web app scale requirements (i.e. 200 calls

or less). However, for deployments that need larger web app scale, from version 3.1 we recommend Cisco Meeting Server web edge as the required solution.

Cisco Expressway software's Edge features have been developed to enable the Cisco Expressway-E to be used as the Edge device in Meeting Server deployments. The Cisco Expressway offers SIP firewall traversal, a reverse web proxy to support external participants joining Meeting Server conferences using the browser-based web app, and TURN Server capabilities to support media traversal for web app and remote Lync and Skype for Business clients.

In addition, the Cisco Expressway-E can be used as a SIP Registrar to register SIP endpoints or to proxy registrations to the internal call control platform (Cisco Unified Communications Manager or Cisco Expressway-C).

CAUTION: Important notes for Expressway users
Deployments of Web Bridge 3 and the web app have been validated on Expressway version X14.3 or later. Earlier versions of Expressway may not be supported by Web Bridge 3.

Note: Cisco Expressway-E can not be used between on premises Microsoft infrastructure and the Meeting Server. In deployments with on-premises Microsoft infrastructure and the Meeting Server, the Meeting Server must use the Microsoft Edge server to traverse Microsoft calls into and out of the organization.

Note: If you are configuring dual homed conferencing between on-premises Meeting Server and on-premises Microsoft Skype for Business infrastructure, then the Meeting Server automatically uses the TURN services of the Skype for Business Edge.

Table 1 below indicates the configuration documentation that covers setting up Cisco Expressway-E to perform these functions. Table 2 below shows the introduction of the features by release.

Table 1: Documentation covering Cisco Expressway as the edge device for the Meeting Server

Edge feature	Configuration covered in this guide
Connect remote browser based Meeting Server web apps	Cisco Expressway Web Proxy for Cisco Meeting Server Deployment Guide
Connect remote Lync and Skype for Business clients	Cisco Meeting Server with Cisco Expressway Deployment Guide
SIP Registrar or to proxy registrations to the internal call control platform	Cisco Expressway-E and Expressway-C Basic Configuration (X14.3)

Table 2: Expressway edge support for the Meeting Server

Cisco Expressway-E version	Edge feature	Meeting Server version
X14.3	Supports Cisco Meeting Server web app. See Cisco Expressway Web Proxy for Cisco Meeting Server (X14.3)	3.8 and later

1.3 Using the Cisco Expressway-C with the Meeting Server in the core network

In addition to deploying Cisco Expressway-E at the edge of the network, Cisco Expressway-C can be deployed in the core network with the Meeting Server. If deployed between the Meeting Server and an on-premises Microsoft Skype for Business infrastructure, the Cisco Expressway-C can provide IM&P and video integration. In addition the Cisco Expressway-C can provide the following functionality:

- a SIP Registrar,
- an H.323 Gatekeeper,
- Call control in Meeting Server deployments with Call Bridge groups configured to load balance conferences across Meeting Server nodes.

Note: If Cisco Meeting Server is integrated with Expressway-E through neighbor zones, Expressway-E cannot replace the calls for load balancing conferences across Meeting Server nodes and this is not a supported deployment.

Table 3:Additional documentation covering Cisco Expressway-C and the Meeting Server

Feature	Configuration covered in this guide
Call control device to load balance clustered Meeting Servers	Cisco Meeting Server Load Balancing Calls Across Cisco Meeting Servers
SIP Registrar	Cisco Expressway-E and Expressway-C Basic Configuration (X14.3)
H.323 Gatekeeper	Cisco Expressway-E and Expressway-C Basic Configuration (X14.3)

1.3.1 Expressway support of Call Bridge Groups

Cisco Expressway running X8.11 or later software supports Call Bridge grouping to load balance incoming and outgoing calls across clustered Call Bridges. Load balancing is achieved by trying to place calls for a single conference onto as few Call Bridges as possible. This reduces the number of distribution links required to connect the participants in the conference, and therefore reduces the overall load across the Meeting Server. For more information see Section 6.5.

1.3.2 Using the Cisco Expressway H.323 gateway component

In line with Cisco's goal of a single Edge solution across the Cisco Meeting Server and Cisco Expressway, Cisco has removed the H.323 Gateway component from version 3.0 of the Meeting Server software. Customers are encouraged to migrate to the more mature H.323 Gateway component in the Cisco Expressway.

Any H.323 endpoints registered to Expressway-E or Expressway-C will not consume Rich Media Session (RMS) licenses when calling into the Cisco Meeting Server from Expressway version X8.10 onwards.

1.4 Using Meeting Server as the Edge device in Meeting Server deployments

The Meeting Server Edge design requires you to deploy Edge instances of Meeting Server where they are reachable by external participants. This can be in your DMZ or public networks. Because this server is exposed to untrusted traffic, only essential services are enabled, and the recommended deployment is for the Edge instance to be deployed in the DMZ behind a NAT or firewall with selective rules allowing only required traffic. The Edge server in the DMZ must be reachable by the Call Bridge servers deployed in the core. We recommend the DMZ/Intranet boundary be access controlled with only required traffic being allowed.

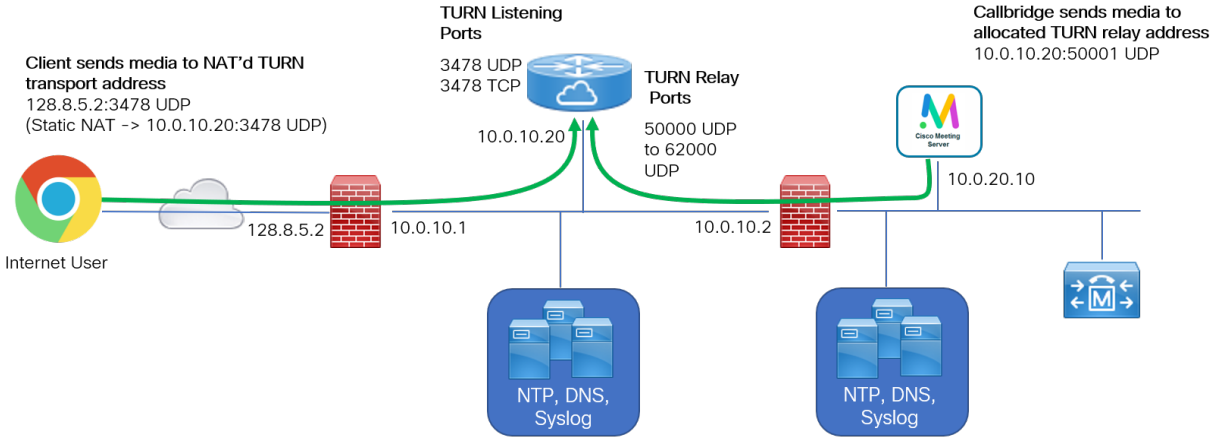
Web app client connectivity is achieved by having the Call Bridge connect outbound to the Web Bridge C2W interface using TLS to establish a secure control channel between the Core and Edge for Web Bridge functions. External browser clients connect to the Web Bridge in the Edge using HTTPS.

Media traffic for external web app clients is handled using a TURN relay setup through the Meeting Server's TURN server. After connecting to Web Bridge and being verified, web clients connect to the TURN server's listening port and request a relay transport address be allocated for them on the TURN server's interface. Using ICE, the client and Call Bridge validate they can send traffic through this relay to each other, and the resulting relay allows both parties to send and receive media across the network boundaries.

Using a TURN relay setup by the external client is the required deployment philosophy for the Edge server to achieve the published call capacities for Meeting Server Edge. Other combinations or scenarios may result in media connectivity being established but can result in reduced capacities and suboptimal media routing and therefore is not advised.

To reduce complexity, this guide only covers the scenario where the remote client establishes the relay.

Figure 4: Example of Meeting Server Edge TURN Server



CAUTION: The Edge Meeting Server must stay within the DMZ and should not be directly connected to networks of different trust levels or security enclaves. The TURN server only needs one interface to perform its relay role.

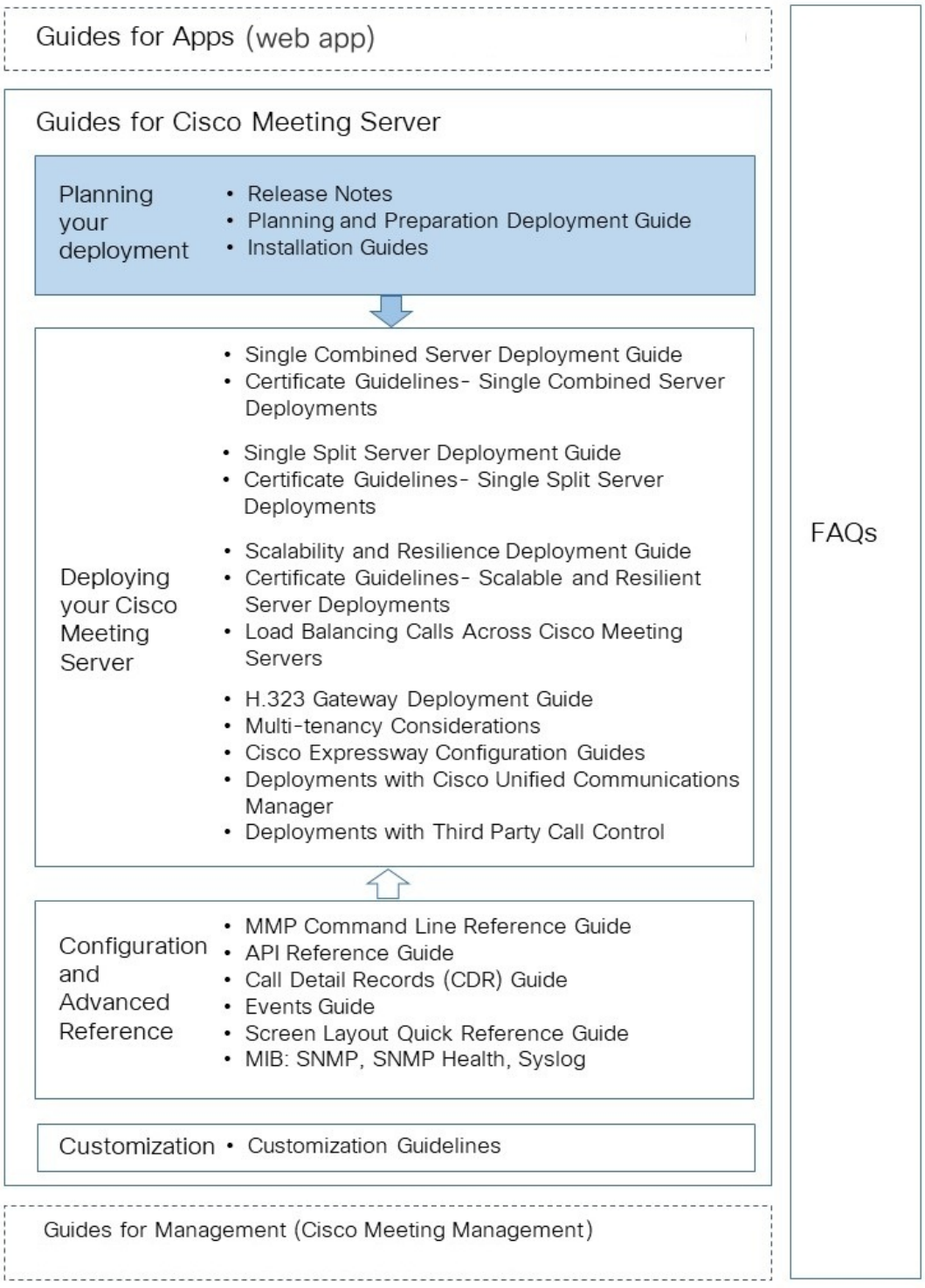
1.5 How to use this guide

This deployment guide follows on from the appropriate Installation Guide for your server, and assumes that you have completed the installation instructions already. This guide should be read and used in conjunction with the appropriate Certificate Guidelines.

In addition to this deployment guide and the Certificate Guidelines, the reference material shown in the figure below can be found on the Cisco Meeting Server documentation page.

Note: Throughout this guide, the term coSpace is referred to as space.

Figure 5: Overview of guides covering the Meeting Server



Note: The address ranges we use in Cisco user documentation are those defined in RFC 5737 which are explicitly reserved for documentation purposes. IP addresses in Meeting Server user documentation should be replaced with correct IP addresses routable in your network, unless otherwise stated.

1.5.1 Commands

In this document, commands are shown in black and must be entered as given—replacing any parameters in <> brackets with your appropriate values. Examples are shown in blue and must be adapted to your deployment.

1.6 Configuring the Meeting Server

There are two layers to the Cisco Meeting Server software: a Platform and an Application.

- **The Platform** is configured through the Mainboard Management Processor (MMP). The MMP is used for low level bootstrapping, and configuration via its command line interface. For example, the MMP is used to enable the Web Bridge, Database clustering, and for various other components.
- **The Application** runs on the MMP platform. Administration of the application level (call and media management) can be done via the Call Bridge's Web Admin interface or through the Application Programming Interface (API) if you prefer. The API uses HTTPS as a transport mechanism and is designed to be scalable in order to manage the potentially very large numbers of active calls and spaces available in a deployment.

  From version 2.9, the application level administration can all be done via the Call Bridge's Web Admin Interface both for single and clustered Meeting Servers.

1.6.1 MMP and API Interfaces

Table 4: Network interfaces configured for the MMP and API on the different Meeting Server platforms

Platform	Access to MMP	Access to Web Admin interface and API
Cisco Meeting Server 2000	Serial over LAN (SoL) connection on blade 1. Note: Before accessing the MMP you need to configure the network settings for the Fabric Interconnect modules	Interface A created during the configuration of MMP. It is a virtual connection that is connected to the external network through uplinks configured on Port 1 of the Fabric Interconnect modules. **Note:** Cisco Meeting Server 2000 platform does not support more than one interface (i.e. configuring 'ipv4 b| c | d' is not supported).
Cisco Meeting Server 1000/ Small and other virtualized deployments	Virtual interface A	One Ethernet interface (A) is created, but up to three more can be added (B, C and D). The Call Bridge Web Admin interface and the API can be configured to run on any one of the A-D Ethernet interfaces.

1.6.2 New tools to ease configuring Meeting Server

The following tools are available to help administrators configure and deploy Meeting Server:

- Installation AssistantSimplifies the creation of a simple Cisco Meeting Server installation for demonstrations, lab environments, or as the starting point for basic installations. From version 3.3 onwards, Installation Assistant is not longer a standalone tool. It is integrated with Meeting Management and can be used from the Meeting Management UI.
- Provisioning Cisco Meeting Server web app users through Cisco Meeting Management, available from version 2.9.
- API access through the Meeting Server web interface. From version 2.9, the Meeting Server API can be accessed via the **Configuration** tab of the Meeting Server Web Admin interface. Some examples in this guide have been changed from using API methods POST and PUT, to using API access through the web interface.

Installation Assistant tool

Use the Installation Assistant to simplify the creation of a single Cisco Meeting Server installation for demonstrations, lab environments, or as the starting point for basic installations. The tool configures Meeting Server based on the best practice deployment described in the Cisco Meeting Server Single Server Simplified Deployment guide. From version 3.3 onwards, it is integrated with Meeting Management to collect information about your setup and then pushes that configuration to the server without you needing to use utilities to access the API, SFTP or the Meeting Server's command line interface. The Installation Assistant can be run from the Meeting Management UI. Refer to the Meeting Management Installation Guide for the software requirements for the client computer, details on installing and running the software, and the steps to configuring a Meeting Server.

Installation Assistant configures Meeting Server to be a SIP MCU capable of making and receiving calls and optionally enables the Cisco Meeting Server web app.

Installation Assistant is intended to be used on an empty, non-configured Meeting Server. It is not a management tool for Meeting Server, nor is it for re-configuring existing Meeting Server installations. The tool is built for configuring Meeting Server virtual machines only. It is not for use with the Cisco Meeting Server 2000 platform.

Using Cisco Meeting Management to provision Cisco Meeting Server web app users

Cisco Meeting Management connected to a Meeting Server or Meeting Server cluster, provides the facility to provision LDAP authenticated Cisco Meeting Server web app users, rather than needing to use the Meeting Server API. The feature also allows admins to create space templates that can be used by web app users to create their own space.

Refer to the Cisco Meeting Management User Guide for Administrators for information on connecting LDAP servers to Meeting Server clusters, how to add one or more user imports, how to create a space template, reviewing and committing the changes and finally running the LDAP sync.

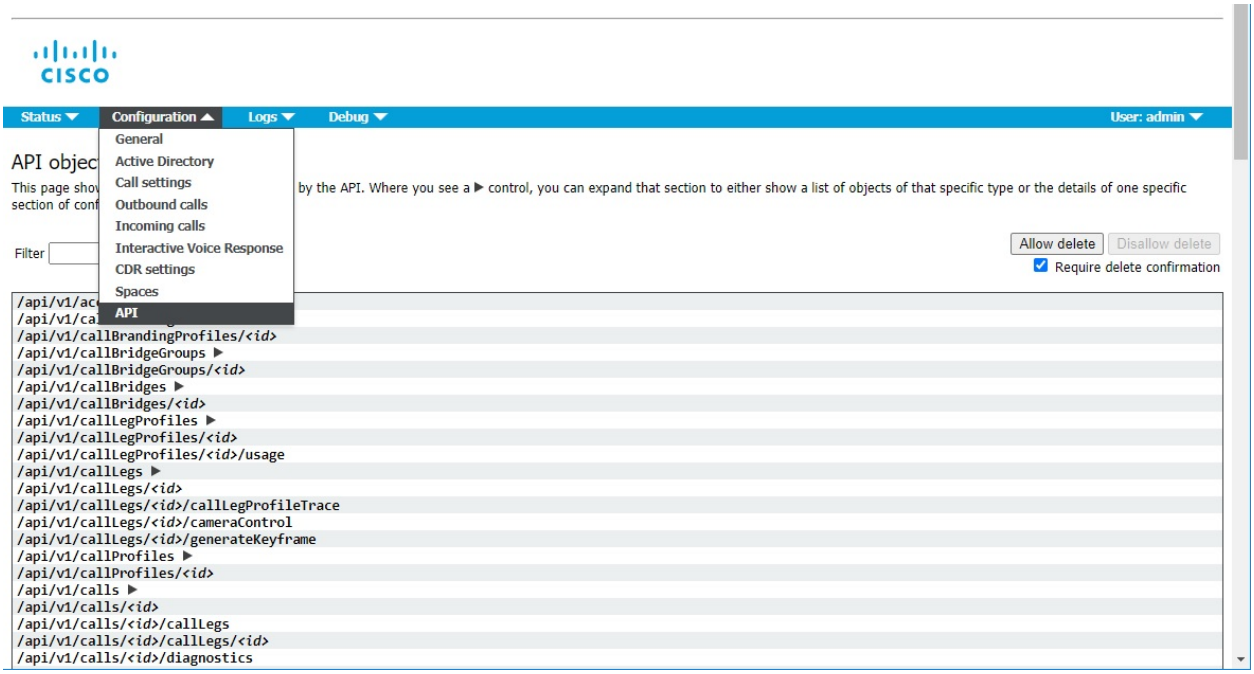
API access on the web interface

To simplify using the Call Bridge API without the need for third-party applications, version 2.9 introduced a user interface for the Call Bridge API that can be accessed via the **Configuration** tab of the Meeting Server web interface, as shown in Figure 6.

The Scheduler APIs introduced in version 3.3 are not supported via this interface. See Scheduler APIs.

Note: To access the API via the web interface you still need to do the initial Meeting Server configuration settings and authentication using the MMP as you would if you were using a third party application. See the MMP Command reference guide for details.

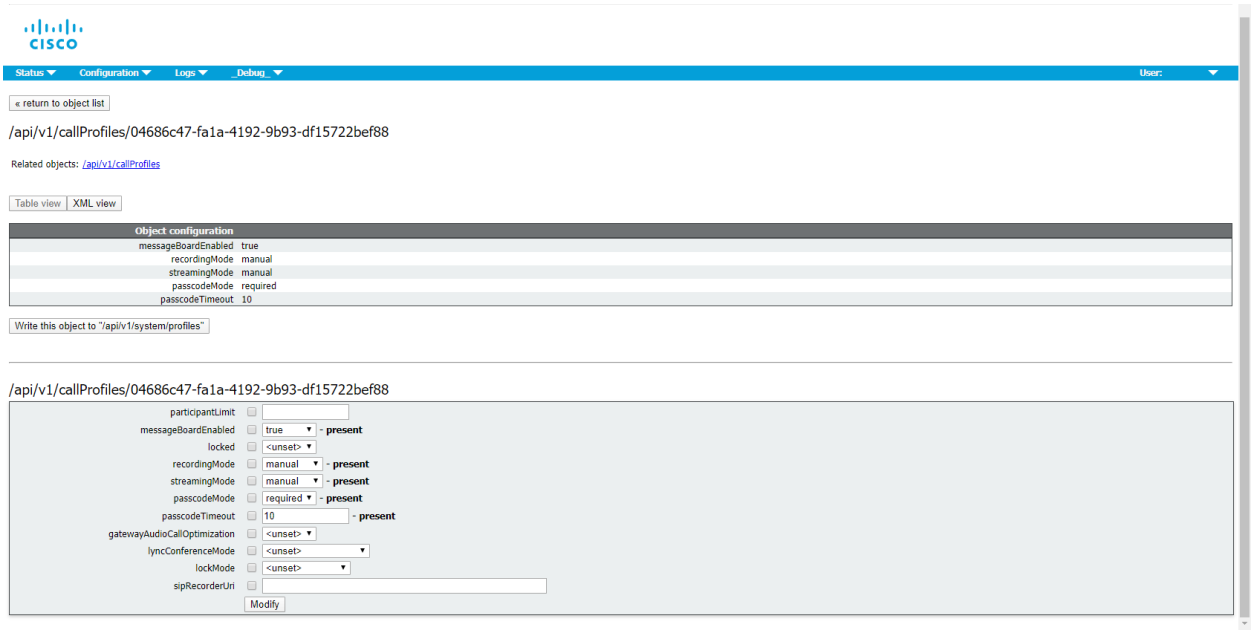
Figure 6: Accessing the Call Bridge API via the Meeting Server web interface



Note: If you wish to delete any configured API objects, select **Allow delete** on the right-hand side of the screen. By default, deletion is disallowed and **Require delete confirmation** is checked to help prevent unintentional deletions.

Using the API via the web interface offers a user-friendly way to work with the API as it gives a more visual approach to configuring your Meeting Server. For example, configuring callProfiles can be achieved using the check boxes and fields shown in Figure 7.

Figure 7: Configuring callProfiles using API access on the web interface



1.7 Meeting Server licensing

You will need licenses to complete a setup of the Cisco Meeting Server. Meeting Server requires license management through the Cisco Meeting Management product and supports Cisco Smart Licensing. From the 3.4 release onwards, Smart licensing is mandatory for Meeting Server. The support for traditional licensing has been deprecated from 3.4 and later releases. Customers are advised to move to Smart licensing.

Note: In an environment where you cannot use Meeting Management or connect to the internet due to security reasons, contact your Cisco Account team for alternate licensing options.

This chapter covers licensed features, Smart licensing, and information about Smart accounts and virtual accounts. You can find more information about licensing in this section.

1.7.1 Licensed features

The following Meeting Server features require a license:

- Call Bridge
- Call Bridge [No Encryption Support]
- Customizations (for custom layouts)
- Recording or Streaming
- Snapshot of participants in the meeting

In addition to feature licenses, user licenses also need to be purchased, there are 2 different types of user licenses:

- Personal Multiparty Plus (PMP Plus)
- Shared Multiparty Plus (SMP Plus)

See Multiparty licensing for more information.

Note: With Cisco Meeting Management, you can use Trial Mode for a 90 day full featured period without licenses.

1.7.2 Smart Licensing

Version 3.0 of Meeting Server introduced support for Smart Licensing on Cisco Meeting Server using Cisco Meeting Management version 3.0 (or later). This transition to the software licensing model, i.e. moving from traditional Product Activation Key (PAK) licenses to Smart Licensing, improves the user experience of license purchasing, registration and software administration. It also aligns Meeting Server with other Cisco products' approach to software licensing and utilizes Cisco Smart Account – a central repository where you can view, store, and manage licenses across your entire organization.

Note: Cisco Smart Licensing Cloud Certificates will be updated in February 2023. After the update, all communications directly with Smart Licensing cloud or through Cisco Smart Software manager (SSM) on-prem will be impacted. It is recommended to upgrade to Meeting Management 3.6 before Feb 2023. SLR/PLR customers should also upgrade to Meeting Management 3.6 for getting new licenses, performing manual sync or adding a new call bridge.

All new license purchases still receive a PAK code – retain for reference – as all licenses will be available in the Smart Account that Meeting Management will sync to.

For further information and to create a Smart Account, go to: https://software.cisco.com and choose Smart Licensing.

The Meeting Server licensing changes from versions prior to 3.0 are:

- Cisco Meeting Management version 3.0 (or later) is mandatory in version 3.0 – Meeting Management reads the Meeting Server license file, and can handle the product registration and interaction with your Smart Account (if set up).
- You can now license multiple clusters with one set of Meeting Server licenses in your Smart Account and you no longer need to load the license file onto each individual Meeting Server instance as was the case prior to 3.0.
- Meeting Management with Smart Licensing tracks how many Call Bridges per cluster, thereby eliminating the need for the R-CMS-K9 activation license.
- For a new deployment with no existing licenses:

    - Newly purchased licenses may be Smart-enabled by default and require a Smart Account – once you have entered the license details into Meeting Management, it will validate the license details against those held in the Smart Account.
- For an existing deployment with a local license file on each Call Bridge:
    - You can move to a Smart Account using the Cisco Smart Software Manager (CSSM) portal and choose the option to convert your existing licenses to Smart.
- SMP Plus and PMP Plus license usage is combined to decide if a day is counted as overage (if either license is over, the whole day is regarded as usage higher than the entitlement). For other feature licenses (for example, recording or custom layout), they are assessed separately and enabled with entitlement via Meeting Management (assuming the license exists in your Smart account).

Note: The term " overage" is used to describe a situation where license usage is higher than the entitlement.

Note: As Meeting Management is required for all 3.0 deployments, for larger customer deployments, Meeting Management can be deployed in new licensing-only mode without active meeting management.

1.7.3 Smart Account and Virtual Account information

Smart Accounts can contain Virtual Accounts which allow you to organize your licenses by any designation of your choice, for example, by department. Here are some important points to note when using a Smart Virtual Account with Meeting Server and Meeting Management:

- Each Meeting Server cluster(s) to a single Meeting Management should be linked to a user-defined Smart Virtual Account.
- Each Virtual Account can only connect with a single Meeting Management server that is configured to handle Smart Licensing.
- Only configure a single Meeting Management to Smart – we recommend you do **not** configure a second redundant Meeting Management for Smart Licensing as double counting of license usage will occur.
- PMP Plus, SMP Plus, and Recording/Streaming licenses can be shared across multiple clusters with a single Meeting Management instance and Smart Licensing in a single Virtual Account.

For more information about licensing, see Additional licensing information.

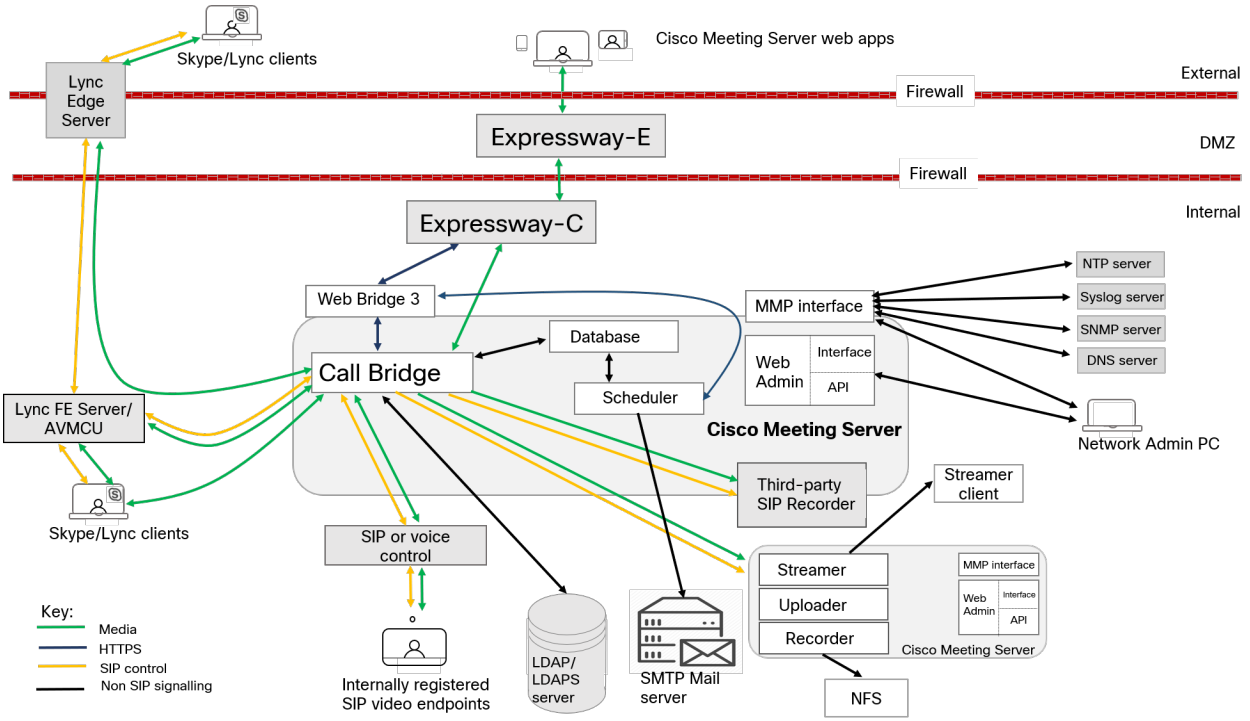
2 General concepts for deployment

This chapter provides an overview of the general concepts for deploying the Meeting Server in a scalable and resilient server deployment. Figure 8 and Figure 9 illustrate typical deployments.

Note: All of the Meeting Server in the deployment must run the same version of software.

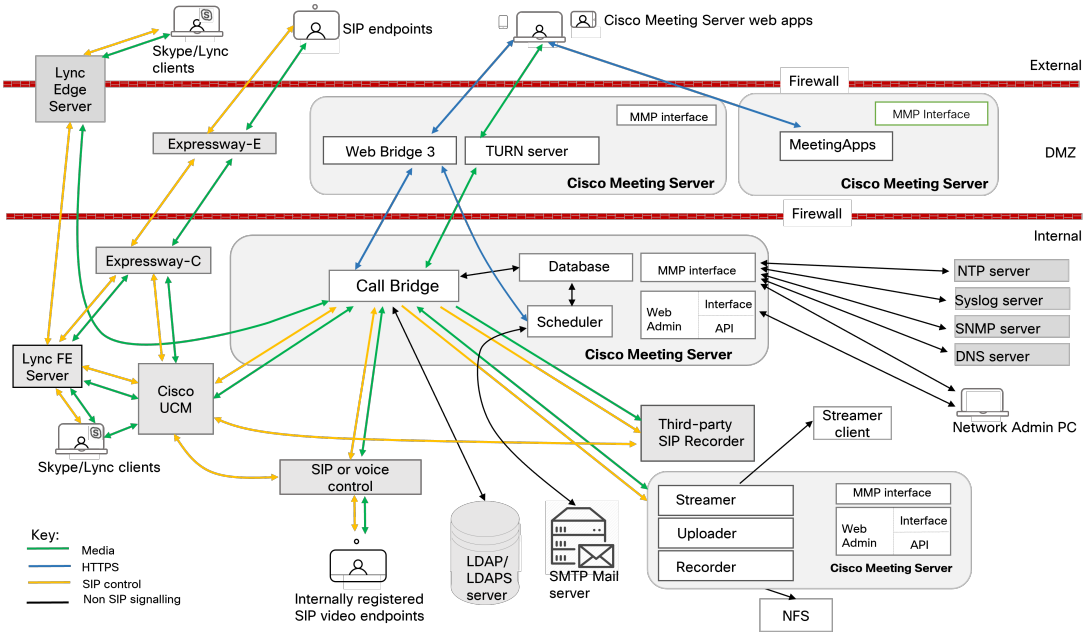
Expressway (Large OVA or CE1200) is the recommended solution for deployments with medium web app scale requirements (i.e. 800 calls or less). Expressway (Medium OVA) is the recommended solution for deployments with small web app scale requirements (i.e. 200 calls or less). However, for deployments that need larger web app scale, from version 3.1 we recommend Cisco Meeting Server web edge as the required solution.

Figure 8: Example of a Meeting Server deployment with Cisco Expressway at the edge



With the greater demand for remote working driving the need for increased web app scale, Cisco Meeting Server version 3.1 has been developed and tested to provide edge support for this increased web app scale. Figure 9 shows an example of how you can deploy the Meeting Server web edge solution to optimize your deployment for larger web app scale.

Figure 9: Example of a Meeting Server deployment using the TURN server component in a split server deployment



Note:

- The Meeting Server includes a Recording facility and a Streaming facility. Enable the Recorder/Streamer on the same server as the Call Bridge only if you are evaluating the features, as this will result in a drop in the connection 15 minutes after the call is initiated. For normal deployment enable the Recorder/Streamer on a different server to the Call Bridge. If you intend to deploy the Recorder and Streamer on the same Meeting Server, you will need to size the server appropriately for both uses. For more information on recording and streaming, see Section 15.

2.1 Web Admin

The Web Admin is a web based interface to configure the Meeting Server.

After configuring the Web Admin Interface for HTTPS access, as described in the Meeting Server installation guide, type the hostname or IP address of the server in a web browser to reach the login screen of the Web Admin Interface. See Web Admin Interface – Configuration menu options for details of the configuration accessible through the Web Admin Interface. From version 2.9, the **API**can be accessed via the **Configuration** tab of the Web Admin Interface.

In addition to providing an administrator web page for Meeting Server, Web Admin also provides the interface for the REST API for Meeting Server. The REST API can be accessed with any conventional REST tool such as Postman or Chrome Poster. Starting with version 2.9, the

Web Admin interface includes an API Explorer interface that allows administrators to work with the Meeting Server API without additional tools/software. The API Reference Guide is available here.

2.2 Call Bridge

The Call Bridge is the component on the Meeting Server that bridges the conference connections, enabling multiple participants to join meetings hosted on the Meeting Serveror Lync AVMCUs. The Call Bridge exchanges audio and video streams so that participants can see and hear each other. The Call Bridge does require licensing to operate.

In a scalable and resilient deployment, Call Bridges can be clustered which allow multiple Call Bridges to operate as a single entity, and scale beyond the capacity of any single Call Bridge. Call Bridges in a cluster can be configured to link peer-to-peer, or for calls to route via call control devices between the clustered Call Bridges. For more information see section *Clustering Call Bridges*.

Note: In deployments involving mainly gateway calls between Lync (or Skype for Business) and SIP, you are advised to use a single standalone Call Bridge to proxy the calls. This is due to the Lync FE only using one Call Bridge, and not implementing a round robin of multiple Call Bridges.

2.3 Database

The Call Bridge reads from and writes to the database storing the space information, for example, the members of spaces, and recent activity within a space.

In a scalable and resilientscalable and resilient deployment, the database can be detached from the Call Bridge and run as a separate component. It can be on the held on the same server as the Call Bridge or on a different server. Multiple instances of the database can be clustered together to provide resiliency in the deployment. See Chapter 5 for information on clustering databases.

2.4 Configuring Web Bridge 3

Web Bridge 3 is a Meeting Server component that enables participants to join meetings using the browser-based Cisco web app client. Web Bridge 3 provides the web server for Cisco Meeting Server web app participants and works in conjunction with the Call Bridge and TURN Server components to support clients. .

Note: If you are not using the web app, you do not need to deploy Web Bridge 3 and can skip this section.

2.5 TURN server

The TURN server provides firewall traversal technology, allowing the Meeting Server to be deployed behind a Firewall or NAT. To connect to the deployment from external web app users or SIP endpoints you need to enable the TURN server, refer to the section on Deploying the TURN Servers . If you are using web apps you also need to configure the Web Admin interface to allow the Call Bridge and external clients to access the TURN server. Using the TURN server does not require a license.

The TURN server listens on port 3478 for UDP connections from the Call Bridge, and is also available for remote connections.

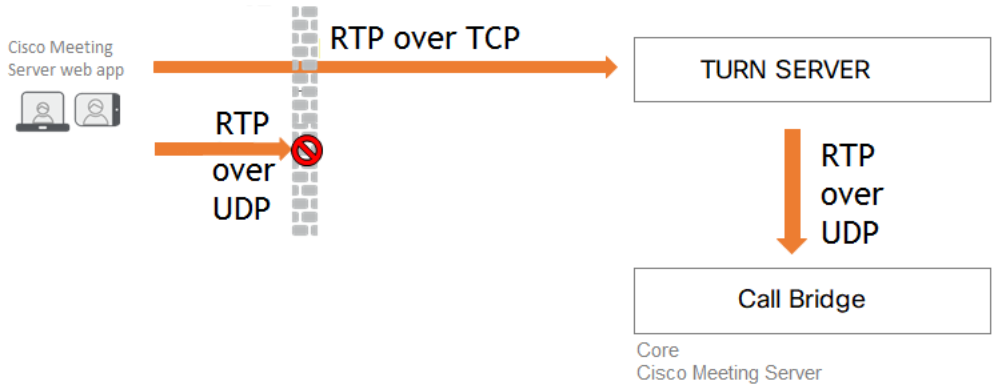
The TURN server can also listen on a second port for TCP from client connections, typically 443 (requires 'turn tls <port>' configuration).

Although the Meeting Server configuration option for enabling TURN TCP is named " tls" , TURN TLS is not used by Meeting Server or web app. Web app uses TCP or UDP and Call Bridge always uses UDP (Media is encrypted using SRTP).

Ensure your firewall rules permit UDP port 3478 from the Call Bridge to the TURN server.

Media sent over TCP is encrypted using SRTP . The TURN server supports TCP to UDP interworking (see Figure 10). A browser can send TCP media to the TURN server which converts it to standard UDP media. This is useful when UDP traffic from browsers is blocked.

Figure 10: TURN server supporting TCP and UDP



The default configuration of the TURN server has changed. How the change impacts your deployment will depend on whether you have deployed combined or split servers.

- A TURN server in a combined server deployment must be configured to listen on the loopback interface. See Section 4 for details.
- A TURN server in a split server deployment now listens on port 3478 for TCP communication from the Call Bridge, instead of port 443 as in previous releases. You need to open ports UDP 3478 and TCP 3478 in your firewall.

2.6 Meeting Server Edge

Meeting Server Edge or CMS Edge is the label used to describe a limited role Meeting Serverr instance deployed in the DMZ or external network to be the point of contact for external web app participants. One or more limited-service instances of Meeting Server are deployed in your DMZ or external network to be the 'Edge' role and work in conjunction with Meeting Server instances deployed in the internal network - the ‘Core’.The CMS Edge should only have Web Bridge 3 and TURN services enabled on it. This deployment scenario is a high-capacity alternative to using the Cisco Expressway as the proxy and TURN Server for external web app participants. The Meeting Server Edge deployment model does not address SIP firewall traversal needs – traversal needs for SIP calling must be addressed separately using Cisco Expressway or other SIP calling technologies. A typical Meeting Server Edge deployment would use Cisco Expressway for SIP calling and the Meeting Server Edge functionality for Cisco web app participants.

2.7 Recording meetings

Prior to 3.0, Meeting Server's internal recorder and streamer components were dependent upon the Meeting Server's internal XMPP server component – in 3.0 this XMPP server is removed. Version 3.0 introduces a new internal recorder and streamer, both SIP-based.

The new internal recorder and streamer components and dialing out to third-party SIP recorders are all configured using SIP URIs, so when recording or streaming is started the administrator-configured SIP URI is called.

The internal SIP Recorder component (from version 3.0) on the Meeting Server adds the capability of recording meetings and saving the recordings to a document storage such as a network file system (NFS).

For more information on recording meetings, see Section 15.

2.7.1 License keys for recording

You will need one or more licenses for recording. One ‘recording’ license supports 1 concurrent streaming or 1 recording, existing recording licenses will allow streaming. Contact your Cisco sales representative or partner to discuss your licensing requirements.

2.8 Streaming meetings

The internal SIP Streamer component (from version 3.0) adds the capability of streaming meetings held in a space to the RTMP URL configured on the space.

An external streaming server needs to be configured to be listening on this RTMP URL. The external streaming server can then offer live streaming to users, or it can record the live stream for later playback.

Note: The Streamer component supports the RTMP standard in order to work with third party streaming servers that also support the RTMP standard. Vbrick is the officially supported external streaming server, however, other servers have also been tested.

Version 3.1 extends the RTMP support in the internal SIP streamer application to RTMPS – essentially RTMP over a TLS connection. Previously all traffic between the streamer and RTMP server was unencrypted, 3.1 RTMPS support allows this traffic to be encrypted.

The existing `tls` MMP command is extended to optionally allow configuration of TLS trusts for RTMPS. This step is optional but recommended. If a TLS trust is not configured then the RTMPS connection will not be secure.

2.8.1 License keys for streaming

You will need one or more licenses for streaming. One 'recording' license supports 1 concurrent streaming or 1 recording, existing recording licences will allow streaming. Contact your Cisco sales representative or partner to discuss your licensing requirements.

2.9 Hosting branding files locally

One set of branding files can be held locally on the Meeting Server. These locally hosted branding files are available to the Call Bridge and Web Bridge once the Meeting Server is operational, removing the risk of delays in applying customization due to problems with the web server. The images and audio prompts replace the equivalent files built into the Meeting Server software; during start up, these branding files are detected and used instead of the default files. Locally hosted branding files are overridden by any remote branding from a web server.

You can change these locally hosted files simply by uploading a newer version of the files and restarting the Call Bridge and Web Bridge. If you remove the locally hosted files, the Meeting Server will revert to using the built-in (US English) branding files after the Call Bridge and Web Bridge have been restarted, providing a web server has not been set up to provide the branding files.

Note: To use multiple sets of branding files, you still need to use an external web server.

For more information on hosting branding files locally, see the Cisco Meeting Server Customization Guidelines.

2.10 On screen messaging

The Meeting Server provides the ability to display an on-screen text message to participants in a meeting hosted on the Meeting Server; only one message can be shown at a time. Using the API, the duration that the message is displayed can be set, or made permanent until a new

message is configured. Use the **messageText**, **messagePosition** and **messageDuration** parameters for API object **/calls**.

For users of SIP endpoints and Lync/Skype for Business clients, the on-screen text message is displayed in the video pane. The position of the message in the video pane can be selected from top, middle or bottom.

On screen messaging is also sent to other devices that are using ActiveControl in the deployment, for instance CE8.3 endpoints, and individual Meeting Servers not in a cluster but with the in-call message feature enabled. Meeting Servers in a cluster also support on screen messaging through a proprietary mechanism.

2.11 SIP trunks and routing

The Meeting Server requires SIP trunks to be set up from one or more of the following: SIP Call Control, Voice Call Control and Lync Front End (FE) server. Changes to the call routing configuration on these devices are required to route calls to the Meeting Server that require the Web Bridge service for interoperability.

2.12 Support for Lync and Skype for Business

2.12.1 Support for Lync and Skype for Business clients

You can use Skype for Business clients, and Lync 2010 and Lync 2013 clients connected to a Skype for Business server, Lync 2010 or 2013 server . From version 2.6, the Meeting Server supports Skype for Business 2019.

The Meeting Server uses:

- the RTV codec transcoding up to 1080p with the 2010 Lync Windows client and 2011 Lync Mac clients,
- the H.264 codec with the 2013 Lync Windows client and Skype for Business client.

The Meeting Server will provide both RTV and H.264 streams when a mixture of clients versions are connected.

Lync 2010 and 2013 clients and Skype for Business clients can share content. The Meeting Server transcodes the content from native Lync RDP into the video format used by other participants in the meeting and sends it as a separate stream. Lync and Skype for Business clients also receive content over a RDP stream and can display it separately from the main video.

The Lync FE Server will need a Trusted SIP Trunk configured to route calls originating from Lync endpoints through to the SIP video endpoints i.e. to route calls with destination in the SIP video endpoint domain through to the Call Bridge.

The SIP Call Control will require configuration changes to route calls destined to the Lync/Skype for Business client domain to the Call Bridge so that SIP video endpoints can call Lync/Skype for Business clients.

The dial plan routes Lync/Skype for Business calls between these two domains in both directions.

The Meeting Server includes support for Lync Edge to enable Lync/Skype for Business clients outside of your firewall to join spaces.

Dual homed conferencing functionality improves how the Meeting Server communicates with the Lync AVMCU, resulting in a richer meeting experience for both Lync/Skype for Business and Cisco Meeting Server web app users. Appendix G describes the dual homed conference experience.

2.12.2 Support for Dual Homed Conferencing

Dual homed conferencing requires the Lync Edge settings to be configured on the Lync Edge server settings on the Meeting Server for conference lookup. If you already have an on-prem Lync deployment or Lync Federation deployment working with the Meeting Server deployment, then no additional configuration is required on the Meeting Server. If this is a new deployment, then you need to setup the Meeting Server to use the Lync Edge server, see Chapter .

For information on the features which improves the experience of participants in Lync/Skype for Business meetings, see:

- FAQ on the improvements in meeting experience for Lync participants,
- FAQ on RDP support,
- FAQ on multiple video encoder support.

2.13 Web Scheduler

The Scheduler is a Meeting Server component that allows end users to schedule meetings via the web app. It is supported on Meeting Server 1000/ Small,Meeting Server 2000 and Meeting Server on VM deployments. For Meeting Server on specification-based VM platforms, an additional 4 GB of RAM is required for running the scheduler component. There is no additional RAM requirement for Meeting Server 1000/ Small and Meeting Server 2000. Scheduler supports sending email notifications via configuration of an SMTP email server. For more information on email server configuration, see Cisco Meeting Server Installation Guides.

One scheduler supports 150,000 meetings; two or three schedulers can be added to provide resiliency but the capacity remains at 150K scheduled meetings. Scheduled meeting data is stored in the Meeting Server database and both clustered and single box database deployments are supported.

For more information, see Scheduler - Deployment.

2.13.1 Scheduler in the web app UI

- The user interface for scheduling meetings will be displayed to web app users, provided at least one scheduler has established a connection to the Web Bridge. If no schedulers are enabled then the web app user will not see the user interface for scheduling meetings.
- When the administrator adds, removes, or changes Web Bridges via the Call Bridge /Web Bridges API, the scheduler does not automatically become aware of those changes. Therefore, the schedulers must be restarted. Similarly, when a scheduler is disabled, the Web Bridges are not aware that the scheduler is purposely disabled rather than just down for some unexpected reason. If the scheduler is intentionally disabled by the administrator, a restart of the Web Bridges is recommended so that the scheduling user interface is not displayed.
- When a scheduler is down due to being disabled or some other issue, the Web Bridge uses a different scheduler if available. Otherwise, an error is displayed to the web app users.

2.14 General points about scalability and resilience

Note: Any references to Web Bridge in this section applies to Web Bridge 3.

For scalability and resilience the Meeting Server can be deployed with:

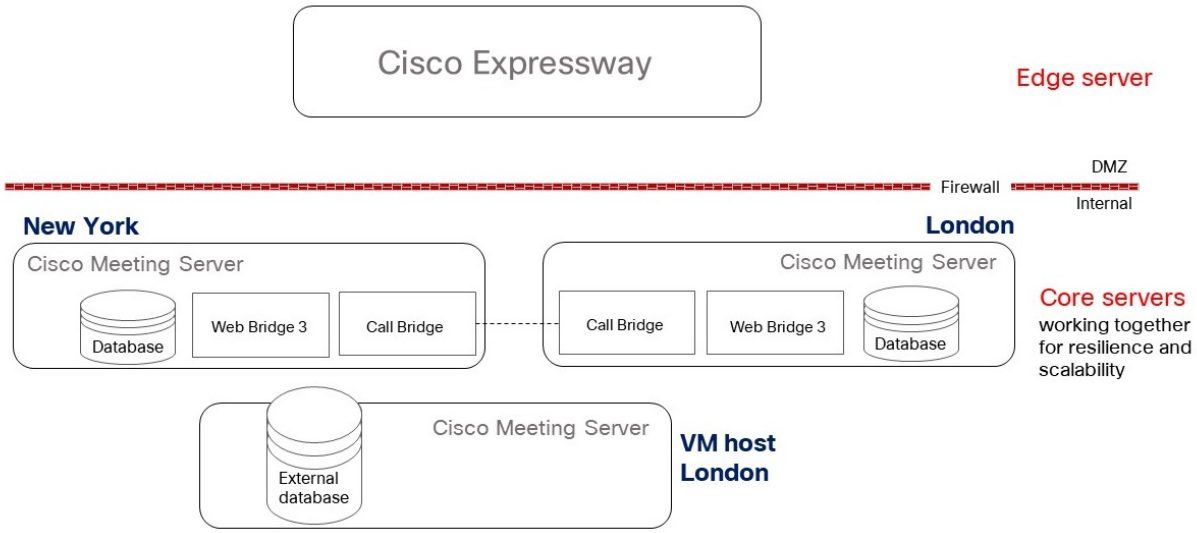
- Multiple combined servers (each server having the appropriate components enabled). The location of these servers will depend on your requirements, possibly one at each point of presence—although this is not a requirement.

  When scaling to a large deployment, it is not necessary (and not always recommended) to have every component enabled on every instance: this is discussed more fully later.
- Multiple Edge and/or Core servers (Edge servers will have edge components enabled, Core servers will have core components enabled, Call Bridge and the database are always core components). The location of these Edge and Core servers will depend on your requirements, possibly co-locating Core and Edge servers in the same data center — although this is not a requirement.
- In a large split deployment it is not necessary, or even desirable, to have the same number of Edge and Core servers. For example, one Call Bridge can manage multiple Web Bridges; those Web Bridges can be reachable externally with a single DNS name resolving to potentially multiple separate units.

2.14.1 Example using multiple “combined” servers

There are many topologies in which to deploy the Meeting Server but a simple example is shown below: this provides resilience and double the capacity of a single host server solution.

Figure 11: Simplest scaled and resilient configuration



This deployment shows two host servers each with all the components enabled, and a third host server with just a database that is likely to be a virtualized (VM) host. Ideally this third host would be located at a different site to either of the other servers. This allows for a total outage of either site to be handled. For a database VM host we recommend:

- Enabling hyper-threading
- Not changing any of the default ESXi system parameters

Note: Do not create a database cluster of 2 nodes, as it reduces resilience rather than increases it. Using an odd number of nodes aids resiliency in the case of network partitions, and Cisco recommends running at least 3 database nodes.

Such an implementation can provide:

- Consideration of geographic location
- Resilience because if any one component is unavailable at the time that a call starts, its “partner” will be used

  Similarly, if a component becomes unavailable during a call, while the call will drop for any PC/WebRTC Client using it – if the participant calls in again, a new call with a new route will

be established and the participant can re-join the call remaining unaware of the new route.

- Ability to scale by using both Call Bridges seamlessly

Points to note in Figure 11:

- The three database servers are clustered using the MMP, as described in Section 5.3. Clustered databases have their contents synchronized.
- Each database can be on the same server as one of the Call Bridges (recommended in most deployments), on a separate virtualized server or as shown in the previous figure, on a combination

  Note:
  In a large deployment with several Core servers, it is not necessary to have a database instance for every Call Bridge; rather we recommend one at every point of presence (POP). (For example, you may want the database in a local data center where you can control physical access but require Call Bridges around the world.

- The two Call Bridges are clustered using the Web Admin Interface as described in *Deploying the Call Bridges* . In addition, they are aware of Web Bridge 3 on the other host server. They also connect to the database cluster to read from and write to it

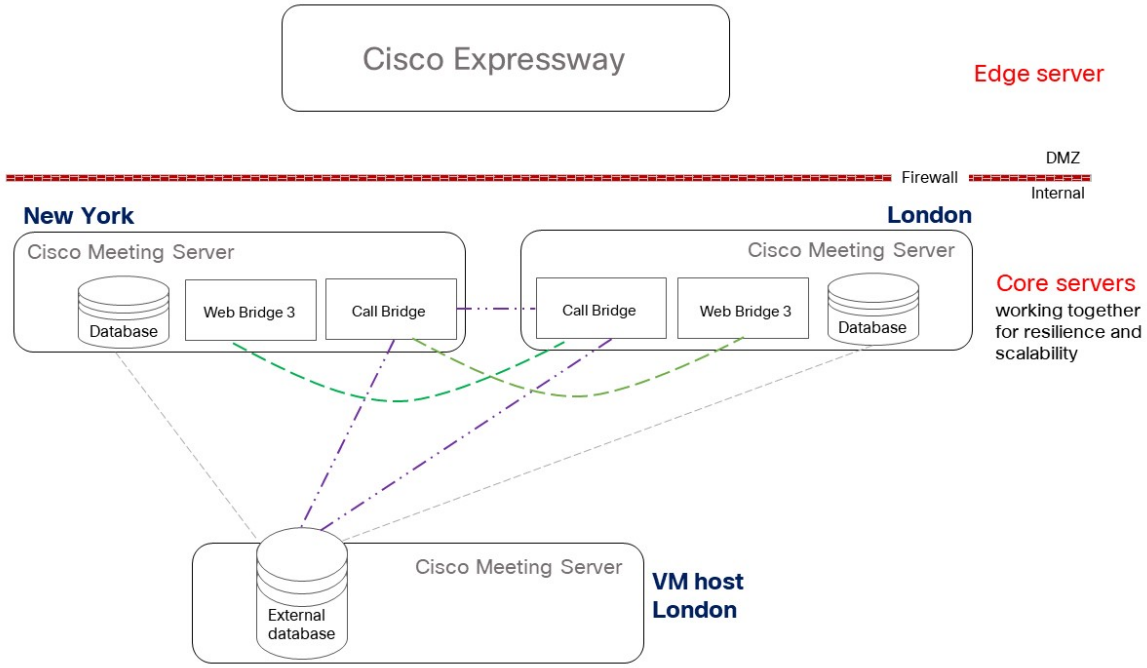
  Each Call Bridge provides CDRs for the call legs that it is hosting. Each CDR identifies the space ID so you can identify the same meeting on different Call Bridges by collecting together calls with the same space ID

  Note: Clustered Call Bridges cannot use the same database (or database cluster) as a non-clustered Call Bridge.

- The two Web Bridges are configured using API configuration through the Web Admin Interface, as described in Appendix 1.2
  - The Web Bridge services can be configured to have a single DNS A record externally. However, when configuring the Web Bridges on the Call Bridge(s) there must be a unique hostname or IP address for each Web Bridge configured on the Call Bridge(s). This enables, each Web Bridge to be uniquely identifiable by every Call Bridge

Figure 11 transforms into Figure 12 when connections are shown.

Figure 12: Simple deployment showing connections



2.14.2 Database clustering

Points to note relating to a database cluster:

- There is a network latency limitation (or Round Trip Time) of 200 ms or less between the database servers, and between the Call Bridge and the primary database.
- All inter-database communication between database cluster peers is handled through SSL for security.
- Within a database cluster, only one database is used at any time by all the Call Bridges; this is the “primary” database. All reads and writes are performed on this database instance.
- This primary database’s contents are replicated to the “replicas/hot-standbys” for resilience: this is indicated in the figures in Chapter 5.
- In case of failure of the primary database, a replica database will be " promoted" to being the new primary, and other replicas will reregister with the new primary database. After the failure has been corrected, the old primary database will assign itself as a replica and will also register with the new primary database:
  - Loss of power to the primary database results in that database reverting to being a replica on startup.

  - Loss of all network connectivity to and from the primary database results in that database becoming a replica when connectivity is restored.

- In cases where a network partition occurs, only databases that can see more than half of the total number in the cluster are considered for promotion to being a primary database. Likewise, any existing primary that cannot see more than half of the database cluster will be demoted to a replica. This ensures that multiple primarys are not created, and that the contents of the database remain consistent across the cluster.

- If the network is split so that no part contains more than half of the servers; then for safety, the database cluster reverts to containing no primary databases. This situation can also occur if the cluster contains an even number of nodes, or if the network is partitioned into three or more disconnected pieces.

  CAUTION: In cases where no primary database can be elected, the system administrator must reinitialize the cluster. This can be done by following the steps on initialization and attachment described in Configuring the Databases. For this reason, we recommend having at least 3 databases when using database clustering.

- When a Call Bridge can only see replica databases it continues to operate, but will not be able to read or write any database. This includes modification operations via the API, modification of spaces via a Cisco Meeting Server web app and LDAP sync. Only active calls that are connected to the Call Bridge work while no new calls are supported.
- In order for a Call Bridge and a database to communicate, the two must be running the compatible database schemas. In a single-node (non-clustered) system, the Call Bridge automatically upgrades the database schema to the latest version when it first boots. However in the clustered scenario, this process has been made manual to allow greater control of when the upgrade occurs — as described in Upgrading the database schema
- There are two important time factors:
  - Time after becoming isolated for a primary database to revert to a replica: 5-6 seconds
  - Time after the primary database goes down for a replica to become the primary: 10-15 seconds
- From version 2.7, database clusters require client and server certificates signed by the same CA configured in each Meeting Server holding or connecting to a database in the cluster. Enforcing the use of certificates ensures both confidentiality and authentication across the cluster.
- CAUTION: If a database cluster was configured without certificates using an earlier version of Meeting Server software which did not require certificates, then on upgrading to version

2.7 the database will stop and remain unreachable until certificates are configured and the database cluster is recreated.

- CAUTION: The nodes forming a database cluster must be configured with a trusted root CA certificate so that only legitimate nodes can connect to the cluster. The nodes will trust connections that present a certificate chain that ends with a trusted root certificate. Therefore each database cluster must use a dedicated root certificate, the root certificate or intermediate certificates must not be used for any other purpose.

- CAUTION: From version 3.5, before upgrading your Meeting Server, uncluster the nodes using the **`database cluster remove`** command. Users must uncluster the nodes, upgrade the Meeting Server and cluster the nodes back using the MMP commands. See Chapter 5 for steps on the clustering databases.

3 Prerequisites

3.1 Prerequisites for installing and configuring the Meeting Server

This chapter describes the changes to your network configuration that you need to consider before installing and configuring the Meeting Server; some of these items can be configured beforehand.

3.1.1 DNS configuration

The Meeting Server needs a number of DNS SRV and A records. See Appendix A for a full list, but specific records are also mentioned elsewhere.

DNS names can be configured to resolve to multiple IP addresses with priority and weighting to each. In advanced configurations, the result of the DNS resolution can be set up to be dependent on the location of the requestor.

3.1.2 Security certificates

You will need to generate and install X.509 certificates and keys for services which use TLS; for example, Call Bridge, Web Admin Interface (the Call Bridge's interface), Web Bridge 3, TURN server, and the Network Load Balancer (if used).

The Certificates Guidelines for scalable and resilient deployments contains both background information on certificates and instructions, including how to generate self-signed certificates using the Meeting Server's MMP commands. These certificates are useful for testing your configuration in the lab. However, in a production environment we **strongly recommend** using certificates signed by a Certificate Authority (CA).

Instructions that were previously in this guide concerning certificates have been removed and replaced by a single step referencing the Certificate Guidelines.

Note: If you self-sign a certificate, and use it, you may see a warning message that the service is untrusted. To avoid these messages re-issue the certificate and have it signed by a trusted CA: this can be an internal CA unless you want public access to this component.

3.1.3 Firewall configuration

See Appendix B for the list of ports which need to be opened on your firewall, and Section 20.6 for advice on creating Firewall rules.

3.1.4 Syslog server

The Meeting Server creates Syslog records which are stored locally and can also be sent to a remote location. These records are useful when troubleshooting because they contain more detailed logging than is available on a Meeting Server's own internal log page. Internal syslog messages can be downloaded over SFTP, however Cisco recommends that the host servers (combined, Edge and Core) are configured to send debug information to a remote Syslog server. This can be to a single Syslog server or to multiple servers; however, if you are using any form of clustering, using the same Syslog server for all servers can simplify troubleshooting. Remember to look in the logs for all the Meeting Servers involved in your issue.

Note: The Syslog server must use TCP, not UDP. Check that your Syslog server is configured to use TCP.

Follow the instructions below on each Meeting Server to define a Syslog server.

1. SSH into the MMP and log in.
2. Enter the following command, **`syslog server add <server address> [port]`**

   Examples:

   ```
   syslog server add syslog01.example.com 514
   syslog server add 192.168.3.4 514
   ```

3. Enable the Syslog server by entering:

   ```
   syslog enable
   ```

4. Optionally, if you want to send the audit log to a Syslog server follow these steps.

   (The audit log facility records configuration changes and significant low-level events. For example, changes made to the dial plan or configuration of a space via the Web Admin Interface or the API, are tracked in this log file, and tagged with the name of the user that made the changealong with the respective source IP address and SSH port. This enables identifying the source of events, especially in concurrent sessions . The file is also available via SFTP.)

   a. Create a user with the audit role.

      ```
      user add <username> (admin|crypto|audit|appadmin)
      user add audituser audit
      ```

   b. Log out of the MMP and log back in with the newly created user account.

   c. Enter the command (this command can only be run by a user with the audit role):
      `syslog audit add <servername>`
      `syslog audit add audit-server.example.org`

Note: Normally local Syslog files are overwritten in time, but you can permanently store system and audit log files using the **`syslog rotate <filename>`** and **`syslog audit rotate <filename>`** commands. These files can also be downloaded over SFTP. See the MMP Command Reference.

3.1.5 Network Time Protocol server

Configure one or more Network Time Protocol (NTP) servers to synchronize time between the Meeting Server components.

Note: Sharing a common view of time is important for multiple reasons, it is necessary when checking for certificate validity and to prevent replay attacks. It also ensures that timings in the logs are consistent.

On each Meeting Server:

1. If necessary, SSH into the MMP and log in.
2. To set up an NTP server, type:

   `ntp server add <domain name or IP address of NTP server>`

To find the status of configured NTP servers, type **`ntp status`**

See the MMP Command Reference for a full list of **`ntp`** commands.

3.1.6 Call Detail Record support

The Meeting Server generates Call Detail Records (CDRs) internally for key call-related events, such as a new SIP connection arriving at the server, or a call being activated or deactivated. It can be configured to send these CDRs to a remote system to be collected and analyzed. There is no provision for records to be stored on a long-term basis on the Meeting Server, nor any way to browse CDRs on the Meeting Server.

The core servers in a scalable server deployment supports up to four CDR receivers, enabling you to deploy different management tools such as Meeting Management, or more than one instance of Meeting Management for resiliency. In a resilient deployment, each of the Core servers generates separate CDRs. To get a consistent picture of the whole deployment the Core servers should use the same CDR receivers.

For more information on setting up Meeting Management as a CDR receiver, see the Cisco Meeting Management Admin Guide.

You can use either the Web Admin Interface or the API to configure each core Meeting Server with the URI of the CDR receivers. If you are using the Web Admin interface go to **Configuration > CDR settings** and enter the URI of the CDR receivers. Refer to the Call Detail Records Guide or the API Reference guide for details on using the API to configure the Core Meeting Servers with the URIs of the CDR receivers.

Note: The list of CDR receivers is held locally to an individual Call Bridge, it is not stored in the database shared between clustered Call Bridges.

3.1.7 Host name

Cisco recommends that each Meeting Server is given its own hostname. This allows for easier diagnostics of issues in clustered deployments.

1. If necessary, SSH into the MMP and log in.
2. Type:
   ```
   hostname <name>
   hostname london1
   hostname mybox.example.com
   ```
3. Type:
   ```
   reboot
   ```

Note: A reboot is required after issuing this command.

3.1.8 Other requirements

- Access to an LDAP server to import users. This can be a Microsoft Active Directory (AD) server or an OpenLDAP server.

  If you plan for users to utilise the web apps to connect to the Meeting Server, then you must have an LDAP server. User accounts are imported from the LDAP server. You can create user names by importing fields from LDAP as described in LDAP configuration. The passwords are not cached on the Meeting Server, they are managed centrally and securely on the LDAP server. When a web app authenticates, a call is made to the LDAP server.
- Decision on a dial plan to use to reach calls hosted on the Call Bridge. The dial plan will depend on your environment; that is whether you are making one or more of the following types of call: Lync, SIP (including voice) or web app calls. Instructions for deploying this dial plan are given in *Dial plan configuration – overview Dial plan configuration – overview*. Dial plans for scalable and resilient deployment must be set up via the API.
- Access to one or more of the following to test the solution: Lync clients, SIP endpoints, SIP phones and/or web apps as appropriate.
- Access to a SIP Call Control platform if you intend to make SIP calls. Chapter 11 and Chapter explain how to set up a SIP trunk to the Cisco VCS and summarizes the required dial plan configuration changes. Information on setting up the SIP Trunk to a Cisco Unified Communications Manager (CUCM), the Avaya CM and Polycom DMA is provided in the Cisco Meeting Server Deployments with Call Control guide; you can use other call control devices not listed in the guide.

- If you intend to integrate the Meeting Server with an audio deployment, the Meeting Server must connect to a Voice Call Control device attached to a PBX; it is not possible to connect a Meeting Server directly to a PBX.
- If deploying in a Lync environment, access to the Lync Front End (FE) server to make dial plan configuration changes there. The changes required are given in this document.

  The Meeting Server integrates with more than one Lync Front End (FE) server: for the incoming (Lync to Meeting Server) direction, each Lync FE pool can be configured to point to a DNS record that resolves to multiple Call Bridges. Failover happens via DNS; Lync will try each result in turn. For geographic distribution, we assume that the Lync FE pools are geographically distributed and the most logical approach is to point each Lync FE pool to a different set of Call Bridges co-located in the same region.

  For the outgoing direction, via DNS the Meeting Server dial plan can be configured to resolve to multiple Lync FE servers or the Lync Director. Each region can be configured to point to a different FE pool.

  Any production environment which also has a Lync deployment requires certificates that are trusted by the Lync FE server.

3.1.9 Specific prerequisites for a virtualized deployment

- A host server that complies with the resources specified in the Installation Guide for Cisco Meeting Server Virtualized Deployments.

3.2 Meeting Server Edge hardware configuration

The Meeting Server Edge role can be deployed as a single server or as multiple servers. The choice is driven by the concurrent call capacity needed for external web app participants. If a high percentage of your participants are expected to be external web app participants, Cisco recommends deploying Edge servers so their capacity matches or exceeds the Call Bridge capacity in the core. Note that excess Edge capacity will not enable more participants to connect than what the core Call Bridge deployment supports. Edge provides the Web Bridge and TURN capacity for a participant; the core must still provide the Call Bridge capacity for the web app participant.

3.2.1 Edge server configurations

Two virtual machine hardware configurations are supported for the Edge server role. These configurations define the supported minimum hardware requirements and capacities they support.

" Small" Edge Server

1 x Cisco Meeting Server VM with the following specification for supported Cisco hardware

- 4 GB RAM
- 4 vCPUs
- 1Gbps network interface

" Large" Edge Server

1 x Cisco Meeting Server VM with the following specification for supported Cisco hardware

- 8 GB RAM
- 16 vCPUs
- 10Gbps network interface

Recommended processor specifications:

We recommend processor specification such as Intel Xeon E5 2600 running at 2.5GHz or higher. We recommend 1 vCPU to 1 physical CPU.

NIC requirement:

Cisco has tested and validated Split-server deployment using single NIC configuration for the TURN Servers. Hence, from version 3.0, we recommend you configure listening ports for a TURN Server only on one interface.

Co-residency support:

The Edge server can be co-resident with other VMs. However, each 4 vCPU VM has a 1 Gbps NIC requirement and each 16 vCPU has a 10Gbps NIC requirement. The VM host will need sufficient NIC capacity for all applications.

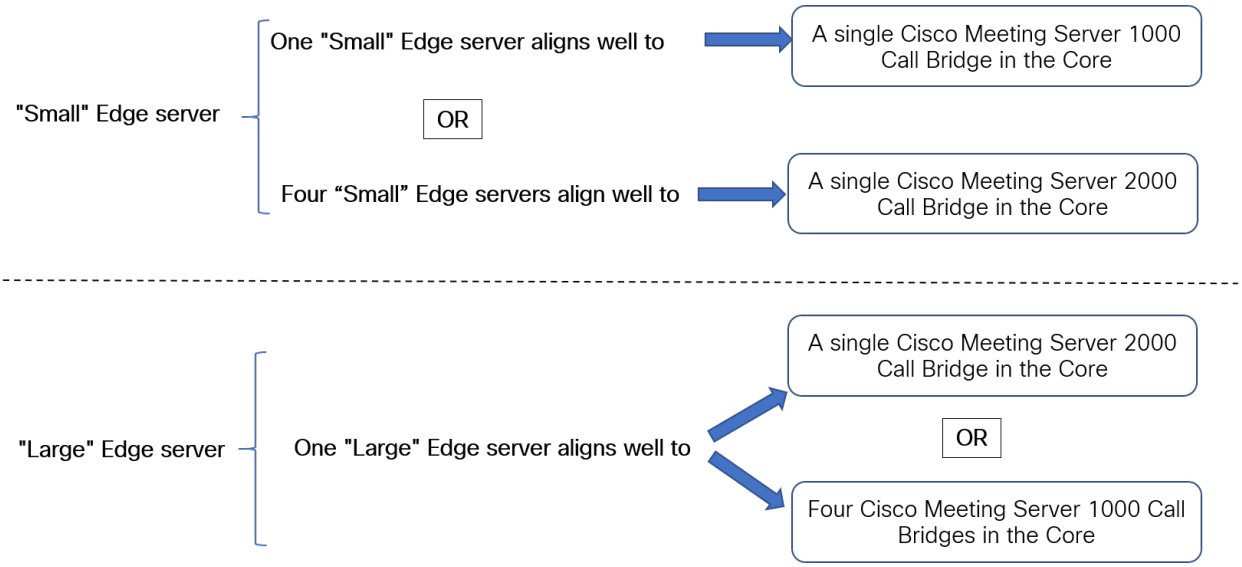
Note:

- Meeting Server M5 onwards hardware supports 10Gbps NIC.
- The CMS 2000 is not suitable as a Meeting Server Edge instance.

Table 5: Edge Server web app call capacities

Type of calls	Small Edge VM Call Capacity	Large Edge VM Call Capacity
Full HD calls 1080p30 video	100	350
HD calls 720p30 video	175	700
SD calls 448p30 video	250	1000
Audio calls (G.711)	850	3000

The two Edge server configurations provide capacities that simplify matching Edge capacity to Core Call Bridge capacity when using Cisco Meeting Server appliances for Call Bridge.



Determine the number of Edge servers needed by reviewing the Call Bridge call capacity the core Call Bridge supports, and the Edge server hardware configuration being used.

3.2.2 Deployment considerations

- We recommend that all edge servers serving the same Call Bridge or Call Bridge Group be the same capacity, i.e. all 4 vCPUs or all 16 vCPUs, not a mix of both.
- For scalable or resilient deployments, we recommend that you configure Call Bridge groups. This allows you to assign a unique group of TURN servers to each Call Bridge group which is useful for helping with load balancing and keeping TURN servers sensibly geolocated with Call Bridges.
- For web app to match SIP scale (up to 24 Call Bridges per cluster), we support multiple edge servers. However, Call Bridge groups only support up to 10 Edge servers per group. For scalable or resilient deployments needing more than 10 Edge servers, more than one Call Bridge group will be necessary.
- To support the Meeting Server Edge solution, a new MMP command **`turn highcapacity-mode (enable|disable)`** is introduced that enables TURN scalability mode. This setting is enabled by default.

3.3 Network Planning for Meeting Server Edge

3.3.1 Technical description

The Meeting Server Edge design requires you deploy Edge instances where they are reachable by external participants. This can be in your DMZ or public networks. The recommended deployment is for an Edge instance to be deployed in the DMZ behind a NAT or firewall with selective rules allowing only required traffic. The Edge server in the DMZ must be reachable by the Call Bridge servers deployed in the core. We recommend the DMZ/Intranet boundary be access controlled with only required traffic being allowed.

Web app client connectivity is achieved by having the Call Bridge connect outbound to the Web Bridge C2W interface using TLS to establish a control channel between the core and edge for Web Bridge functions. External clients connect to the Web Bridge listening port using HTTPS.

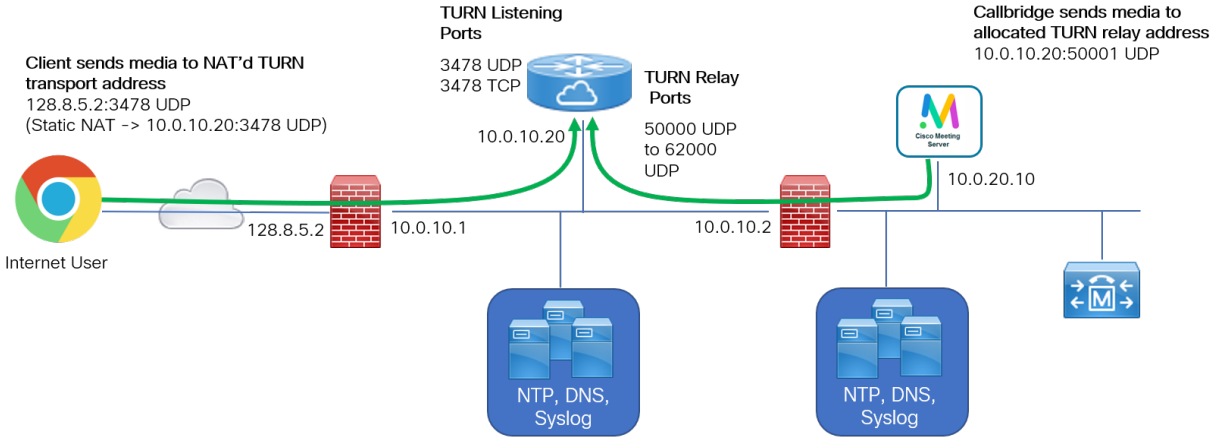
Media traffic for external web app clients is handled using the TURN Server as a relay. Authenticated web clients connect to the TURN Server's listening port and request a relay transport address be allocated for them on the TURN Server's interface.Using ICE, the client and Call Bridge validate they can send traffic through this relay to each other and if it is their best route. The Call Bridge can send media outbound to the allocated relay address, which is sent onward (or 'relayed') to the external client by the TURN Server. Traffic from the client is sent to the TURN Server listening address and relayed using the relay transport address as its source, back to the Call Bridge. The UDP based media can reach the Call Bridge in the core by the firewall allowing symmetric UDP traffic back to the originating connection.

Note: From version 3.0, we recommend you configure the listening ports for a TURN Server on a single interface.

Using a TURN relay setup by the external client is the required deployment model for the Edge server to achieve the published call capacities. Other combinations or scenarios may result in media connectivity being established but can result in reduced capacities and suboptimal media routing and therefore are not advised.

The recommended deployment for Meeting Server Edge enables external web app participants to connect to the Edge instance using TURN over UDP and the Call Bridge connecting to the TURN relay via UDP. This configuration is optimal for balancing security and performance. To improve compatibility with restrictive client networks, an optional scenario to add a second DMZ interface to move TURN to its own interface to support TURN over TCP 443 is also covered. While other combinations of network paths and service configurations are technically feasible, they are not documented by Cisco as they may incur other security risks, capacity impacts, or are simply excluded from this guide to reduce variations.

Figure 13: Sample Diagram of TURN using UDP



3.3.2 Network planning

This section outlines the network requirements to operate Meeting Server Edge instances in a DMZ network. The nomenclature used assumes there are three security levels of network. The Internet, a DMZ, and an Intranet. The scenarios outlined include multiple meeting server instances and TCP fallback. Destinations are labeled based on their role and may map to multiple addresses in your environment.

3.3.2.1 DMZ to Internet boundary

The DMZ by default should only accept incoming connections from the Internet for approved traffic and services. Because we do not know where participants may connect from, connections to these services must be accepted from all source IPs.

Note: The DMZ network maybe NAT'd or directly routable from the public internet. The examples here assume the DMZ is NAT'd.

To support web app, the firewall must accept incoming TCP connections from the Internet to port 443 for the Meeting Server Edge servers hosting the Web Bridge 3 service. Optionally you can enable TCP port 80 if you wish to enable HTTP redirect so users who attempt a HTTP connection will automatically redirect to HTTPS. Participants are not allowed to use HTTP for calls; the port only supports redirecting to HTTPS.

Media is best sent over UDP but call participants on the internet may find themselves behind firewalls which could block UDP traffic, so an optional TCP fallback is offered. For media traffic, the firewall should accept incoming connections to the Edge Server on TURN listening port UDP 3478. When enabling TURN using TCP, TURN server listens also listens on TCP 3478 and the designated port. If enabling TURN using TCP 443, a second DMZ IP interface is needed on the server with TURN and Web Bridge 3 each listening on different interfaces.

Note: If the DMZ is NAT'd and you are using multiple Edge Servers, separate IPs are needed in the NAT configuration for each Edge server because each must be directly addressable from the Intranet for UDP traffic.

3.3.2.2 DMZ to Internet traffic rules

Description	Direction	Source IP	Source Protocol: Port	Target IP	Target Protocol: Port
Client Browser HTTPS	INCOMING	Any	TCP {unreserved}	{WB3}	TCP 443
Client Browser (Optional)	INCOMING	Any	TCP {unreserved}	{WB3}	TCP 80
Client STUN/TURN	INCOMING	Any	UDP {unreserved}	{TURN}	UDP 3478
Client STUN/TURN TCP	INCOMING	Any	TCP {unreserved}	{TURN}	TCP 3478
Client STUN/TURN TCP 443 (Optional)	INCOMING	Any	TCP {unreserved}	{TURN}	TCP 443
Symmetric return TURN traffic (usually automatic)	OUTGOING	{TURN}	UDP {3478}	Any	UDP {unreserved}

Note:

- {WB3} = IP list of Web Bridge 3 server listening interfaces
- {TURN} = IP list of TURN server listening interfaces
- TURN TCP 443 is an optional deployment. If you want to enable TURN TCP on 443 and you are already using TCP port 443 for Web Bridge 3, regardless of whether they are on separate interfaces, you must deploy a new Meeting Server Edge server.
- Firewall must allow symmetric or return UDP traffic to the Internet for media from TURN Server relay
- Each TURN Server must be independently addressable from the Internet when using multiple TURN servers

3.3.2.3 Intranet to DMZ boundary

By default, the firewall should not permit any TCP connections to be made from the Meeting Server Edge server instance towards the intranet to protect the intranet. It must also not allow any UDP packets to be sent from the Meeting Server Edge server into the intranet, unless a

UDP packet has already (and recently) been sent from the intranet to the edge box on the same address/port pairing i.e. a UDP packet from <DMZ IP>:50342 to <Intranet IP>:50131 should be blocked unless there was previously a packet from <Intranet IP>:50131 to <DMZ IP>:50342.

The firewall must allow incoming TCP connections to the Meeting Server Edge server on the C2W listening port from the Call Bridges operating in the core. It should also permit inbound UDP packets from the call bridges operating in the core (i.e. source <any core callbridge IP>:< 32,768 to 65,535 > to destination <Edge CMS IP>:< 50,000 to 62,000 >). The firewall must allow the return UDP traffic for these connections.

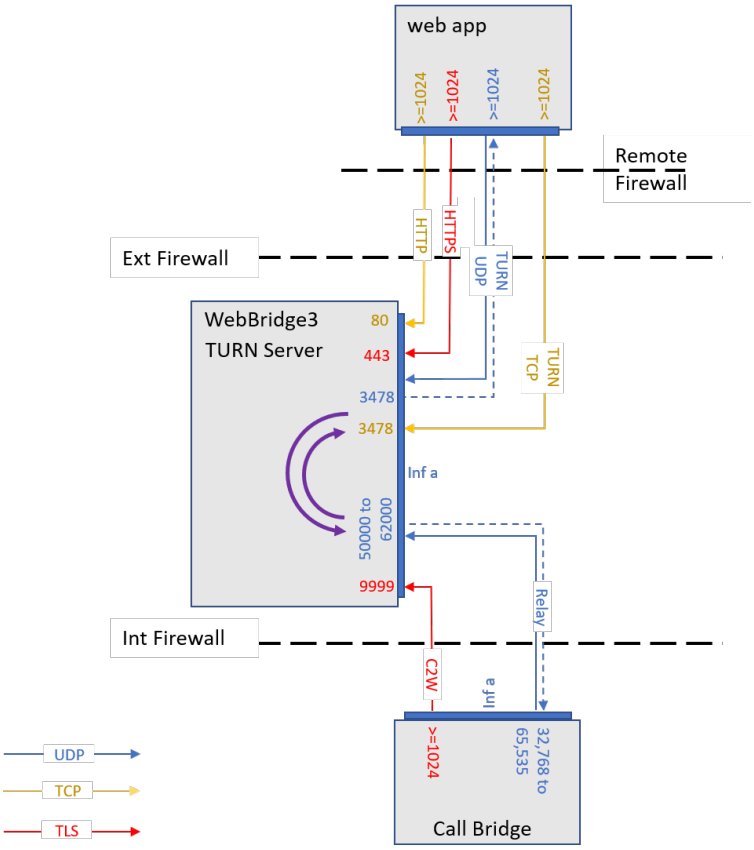
Having the core Call Bridges directly routable with the Meeting Server Edge nodes is preferred, but the core Meeting Servers can behind a NAT relative to the DMZ services and still use the TURN relay allocated by external clients. The core Meeting Servers do not need to connect to the TURN listening ports as the relay setup by the external client is sufficient for both parties. If using NAT, traffic to the call bridge will be seen as a peer-reflexive candidate in the ICE connection testing.

Description	Direction	Source IP	Source Protocol: Port	Target IP	Target Protocol: Port
Meeting Server C2W Interface	OUTGOING	{Call Bridge IPs}	TCP {unreserved}	{WB3}	Web Bridge 3 C2W listening port. Eg., webbridge3 c2w listen a:9999 would use TCP 9999
Call Bridge Media Traffic	OUTGOING	{Call Bridge IPs}	UDP {32,768 to 65,535}	{TURN}	UDP {50,000 to 62,000}
Symmetric return TURN traffic (usually automatic)	INCOMING	{TURN}	UDP {50,000 to 62,000}	{Call Bridge IPs}	UDP {32,768 to 65,535}

Note:

- {WB3} = IP list of Web Bridge 3 servers
- {TURN} = IP list of TURN servers
- Call Bridge = IP list of the Call Bridge servers in the core
- Firewall must allow symmetric/return UDP traffic to the Internet for media from TURN Server relay
- You must configure the listening ports for a TURN Server on a single interface.

Figure 14: web app using TURN 3478 UDP or 3478 TCP



3.3.2.4 Management and Platform Traffic

For clarity purposes, the prior network requirement sections excluded discussing requirements for management services and platform needs. Management and Platform requirements are covered separately in this section. Because infrastructure services and administrative management policies for DMZ networks will vary between organizations, these topics will be described in terms relative to the Edge Meeting Server instance, rather than describing which network boundary they cross. Apply these concepts to the specifics of your environment.

For the Meeting Server to handle TLS and certificates properly the Edge server must be able to access NTP and DNS services. Administrators will also need to use SFTP and SSH to configure and update the Meeting Server software. Syslog for centralized logging is optional, but strongly recommended. These services should be configured to be accessible from the Edge's DMZ network interface while adhering to common security practices such as restricting traffic to known sources.

3.3.2.5 Management Traffic for Meeting Server Edge

Description	Direction	Source IP	Source Proto:Port	Target IP	Target Proto:Port
NTP	OUTGOING	{WB3} or {TURN}	UDP 123	{NTP Server}	UDP 123
DNS	OUTGOING	{WB3} or {TURN}	UDP {unreserved}	{DNS Server}	UDP 53
Syslog	INCOMING	{WB3} or {TURN}	TCP {unreserved}	{Syslog Server}	TCP 514*
App Management (SSH, SFTP)	INCOMING	{Intranet/Admin IPs}	TCP {unreserved}	{WB3} or {TURN}	TCP 22

Note:

- {WB3} = IP list of Web Bridge 3 servers
- {TURN} = IP list of TURN servers
- Syslog destination port is configurable
- Verification of certificates may require outbound connections to OSCP or CRL destinations as defined by the certificates in use
- Not documented here are other server management technologies that maybe used to manage the server hardware or hypervisor used to host the Meeting Server virtual machine (ESXi, Cisco CIMC interface, etc).

3.3.3 Deploying Meeting Server web edge

The following steps give a high level view of how to deploy Meeting Server web edge:

1. Configure the TURN server on the Meeting Server edge via the MMP.
2. Configure Web Bridge 3 on the Meeting Server edge via the MMP.
3. Link the Web Bridge 3 to the Call Bridge, (i.e. add the **callBridge** parameter via the Web Admin user interface under **Configuration > API** to **/api/v1/turnServers** and **/api/v1/webBridges**, and check Web Bridge 3 certificate requirements).
4. Check that connections are functioning correctly – to do this, you can test manually by logging in via the web app address, or check on the Web Admin interface under **Status > General** and look at the **Fault conditions**, and **Recent errors and warnings**. (Note that Web Bridge 3/TURN connection failure messages aren't shown.)
5. Add the firewall settings as follows:
    a. Call Bridge must be able to connect to the TCP connection Web Bridge 3 c2w connection ports (as specified on the " c2w://address:port" in the API; i.e. in the **url**

field on **/api/v1/webBridges**.)

b. TURN relay ports on the Meeting Server edge are 50000 to 62000, so Call Bridge must be able to contact those on UDP to send media.

c. External web app clients must be able to reach the TURN Server on UDP 3478. A fallback to TCP is possible, the port depends on the 'turn tls <port>' configuration so that port would also need to be opened on that occassion.

4 Configuring the MMP

The Meeting Server components are configured using the MMP. Each Meeting Server instance will need configuration.

4.1 Creating and managing MMP and Web Admin interface user accounts

You should have created an MMP administrator user account on each Meeting Server by following the Cisco Meeting Server Installation Guide ; if so, go on to the next section unless you want to set up additional accounts. The same account is used to access the Web Admin Interface.

(If you do not have these MMP administrator user accounts, you will have to use the emergency admin recovery procedure detailed in the Installation Guide appropriate to your deployment.)

You can create additional user accounts for the MMP that have admin level rights using the MMP add user command **`user add <account name> <role>`**.

1. SSH into the MMP.
2. Add an admin level user account, for example:

   ```
   user add adminuser2 admin
   ```
3. Enter the password you want to use for this account twice in order to complete the account creation.

   On login the user will be forced to configure a new password.

Note: See the MMP Command Reference Guide for the full range of MMP commands, including setting up additional administrator user accounts and user accounts with other roles.

4.2 Upgrading software

If you downloaded the software some days ago, we advise you to check on the Cisco website in case a later version is available, and if so, upgrade to the latest version.

CAUTION: If a database cluster was configured without certificates using an earlier version of Meeting Server software which did not require certificates, then on upgrading to version 2.7 the database will stop and remain unreachable until certificates are configured and the database cluster is recreated.

The following instructions apply to all types of deployment:

1. To find out which software version is running on a Meeting Server, SSH into the MMP of the server, log in and type:

   `version`

2. Before upgrading your Meeting Servers:

   a. take a backup of the current configuration on each of the servers. Use the MMP command **`backup snapshot <name>`**. Save the backup safely to a local server. See the MMP Command Reference guide for full details. Do NOT use the automatic backup file that is created during the upgrade process.

   b. save the cms.lic and certificate files to the local server.

   c. using the Web Admin interface, check the database cluster status, and that all calls (SIP and clients) are working and no fault conditions are listed.

   d. If your deployment has clustered database, uncluster the nodes using the **`database cluster remove`** command.

3. To upgrade, first download the appropriate software file from the Cisco website. Click on this link, then click on the appropriate Meeting Server type listed in the right hand column of the web page and follow any instructions displayed with the download link.

4. Use an SFTP client to upload the new software image to the MMP of the Meeting Server. For example:

   ```
   sftp admin@10.1.124.10
   put upgrade.img
   ```

   where 10.1.x.y is an IP address or domain name.

5. Upgrade all Core servers one by one, connect via SSH to the MMP and type:

   `upgrade`

   Start with the non-database servers first, followed by the replica database servers, and the primary database server last. Wait until each server has fully booted, and for database servers to have connected to the database cluster before moving on to the next server.

   For each server: wait approximately 10 to 12 minutes for the Web Admin to be available after upgrading the server. Log in to the Web Admin interface. At this point, the server may report an error for example “Error: remote database has scheme version 7026 (current version is 7045)”. Do not proceed beyond this point until the login to the Web Admin has been successful. Verify the new version using the MMP command **`version`**. If the version is not correct, upload a new image to this server, use the MMP command **`upgrade`** and wait to get back to this point.

6. Once all servers have upgraded, cluster the nodes again by following the steps explained in Chapter 5.

7. Check that the database servers are connected and in sync using the MMP command **`database cluster status`**. Do not go onto the next step until the database servers are in sync.
8. Check that the Web Admin interface on each Call Bridge can display the list of spaces.
9. Upgrade all Edge servers and verify that the upgrades were successful.

This completes the upgrading of the resilient Meeting Server deployment. Now verify that:

- dial plans are intact,
- and no fault conditions are reported on the Web Admin interface and log files.

Check that you are able to connect using SIP and web app (as well as Web Bridge 3 if that is supported).

Note on rollback procedure: If anything unexpected happens after you upgrade the servers and you decide to downgrade, simply upload the software release for the previous version, and type **`upgrade`**. Then use the MMP command **`factory_reset app`** on each server. Once the server has rebooted from a factory reset, use the **`backup rollback <name>`** command to restore the backup file on each server. Providing you restore the backup configuration file that was created from that specific server, the license files and certificate files will match the server.

4.3 Configuring the Call Bridge listening interface

The Call Bridge service should run on the main Meeting Server instance in your internal network. The Call Bridge needs a key and certificate pair that is used to establish TLS connections with SIP Proxies, peers like the Skype Front End (FE) server and for C2W connections for Web Bridge. If your peer SIP Proxy requires TLS (Example: Skype for Business) the certificate must be trusted by the peer.

Note: SIP and Skype calls can traverse local firewalls using the Cisco Expressway, you will need to configure trust between the Call Bridge and the Cisco Expressway. Cisco Expressway must be running X8.9 or later. For more information, see Cisco Expressway Options with Cisco Meeting Server and/or Microsoft Infrastructure (Expressway X8.9.2) or if running X8.10 see Cisco Expressway Web Proxy for Cisco Meeting Server (X8.10) and Cisco Expressway Session Classification Deployment Guide (X8.10).

The command **`callbridge listen <interface>`** allows you to configure a listening interface. The default recommendation is to enable Call Bridge to listen on the first interface 'a'

Configure listening interfaces on each Call Bridge as follows:

1. Create and upload the Call Bridge certificate and keys as described in the Certificate Guidelines.

2. Sign into the MMP and configure the Call Bridge to listen on interface a.

   ```
   callbridge listen a
   ```

   Note: The Call Bridge must not have a NAT between it and SIP participants or SIP Proxies it needs to directly communicate with. Call Bridge can be paired with firewall traversal solutions like Cisco Expressway to address Firewall Traversal or NAT issues, but must not traversal a NAT between it and the SIP Proxy.

3. Configure the certificates Call Bridge will use with the command:

   `callbridge certs <key file> <certificate file> <ca bundle>`

   Example:

   ```
   callbridge certs callbridge.key callbridge.crt ca-bundle.crt
   ```

   More information regarding certificates and using a certificate bundle as provided by your CA, is described in the Certificate Guidelines.

4. Restart the Call Bridge interface to apply the changes.

   ```
   callbridge restart
   ```

4.4 Configuring the Web Admin interface for HTTPS access

The Web Admin interface is needed on your Meeting Server instance where Call Bridge is running but is not required for the Meeting Server instances in the Edge. For reduced attack surface, the recommendation is to not run Web Admin on Edge instances.

The Web Admin Interface is the Call Bridge's user interface. You should have set up the certificate for the Web Admin Interface (by following one of the Installation Guides). If you have not, do so now.

1. The installation automatically set up the Web Admin Interface to use port 443 on interface A. However, the Web Bridge also uses TCP port 443. If both the Web Admin Interface and the Web Bridge use the same interface, then you need to change the port for the Web Admin Interface to a non-standard port such as 445, use the MMP command **`webadmin listen <interface> <port>.`** For example:

   ```
   webadmin listen a 445
   ```

2. To test that you can access the Web Admin Interface, type your equivalent into your web browser: **`https://meetingserver.example.com:445`**

   If it works, proceed to the next section.

3. If you cannot reach the Web Admin Interface:

   a. Sign into the MMP, type the following and look at the output:

      ```
      webadmin
      ```

The last line of the output should say **"webadmin running".**

b. If it does not there is a configuration problem with your Web Admin Interface. Check that you have enabled it by typing:

```
webadmin enable
```

c. The output of the **webadmin** command should also tell you the names of the certificates you have installed, e.g. webadmin.key and webadmin.crt.

Note: They should be the same names of the certificates you uploaded previously.

Assuming these are the names then type:

```
pki match webadmin.key webadmin.crt
```

This will check that the key and certificate match.

d. If you are still experiencing issues, troubleshoot the problem as explained in the Certificates Guidelines.

4.5 Stage Edge Server instances

Complete this section if you will be using Meeting Server as the Edge for external web app participants. If you are not supporting web app clients that cannot directly access the Call Bridge, you do not need Meeting Server Edge and may skip this section.

Meeting Server Edge instances should only be configured with the minimum services necessary to keep its security exposure as minimal as possible. To perform their role, Edge Server instances only need the Web Bridge 3 Service and TURN Services enabled. The server will also need NTP and DNS clients configured so it can do lookups and maintain accurate time required for TLS operations. While optional, configuring syslog to send logs to a central server is recommended. The deployment steps will cover both the standard TURN UDP configuration, and the optional TURN configuration using TCP 443.

Before configuring Web Bridge and TURN, any Meeting Server instances in the Edge should be deployed per the Installation Guide relevant to your platform and have completed:

- Setup access to the server MMP interface (console or SSH)
- Configuring the IP information for the network interface(s)
- Configured the DNS client on the server
- Configured the NTP client on the server
- Configured Syslog if desired

For help with any of these tasks, please refer to the Installation Guides and MMP Command reference.

4.6 Configuring Web Bridge 3

Web Bridge 3 is used to enable the use of the browser based Cisco Meeting Server web app. If you are not enabling use of the web app in your deployment, you do not need the Web Bridge Service and can skip this section.

- If you need to support web app clients from your internal network, you should configure Web Bridge on your main Meeting Server instance in the Core and complete the steps in this section.
- If you are using Cisco Expressway as your proxy and TURN Server for web app, Web Bridge needs to be configured on your main Meeting Server instance in the Core and you should complete the steps in this section.
- If you are using the Edge Meeting Server model, you have the option of running Web Bridge just in the Edge or running it both in the Edge and the main Internal Meeting Server instance. Enabling Web Bridge on the internal server allows clients to use web app without making connections to the Web Bridge in the DMZ. The recommendation for deployments using the Edge Meeting Server model is to run Web Bridge in both the DMZ and internal server instances. Complete the steps in this section and configure Web Bridge on the Edge instances and the main Meeting Server instance in the Core.

Note: Running Web Bridge in both the Core and Edge requires clients resolve the same Web Bridge hostname to either the internal or Edge instance as appropriate for them - this is normally referred to as 'Split-DNS' where the DNS Server resolves names to addresses based on where the client is located.

CAUTION: Important notes for Expressway users
Deployments of Web Bridge 3 and the web app have been validated on Expressway version X14.3 or later, earlier Expressway versions may not be supported by Web Bridge 3.

Note: For more information on the web app, see Cisco Meeting Server web app Important Information.

4.6.1 Useful information to help configure Web Bridge 3

The following is useful information to help you configure Web Bridge 3 so that you can use web app:

- " Call Bridge to Web Bridge" protocol (C2W) is the link between the callbridge and webbridge3. It is an outgoing connection from the Call Bridge to the Web Bridge to establish a control channel between them. Certificates are used to authenticate and secure the C2W connection. C2W is exclusive to Call Bridge - Web Bridge traffic and is not used by users or other services.

- A C2W listening port is defined on the Web Bridge server (using **`webbridge3 c2w listen`**) to allow the Call Bridge to connect to the Web Bridge using an HTTPS connection. There is no set default value for the port number to use, but this guide uses 9999 as the example. This connection must be secured with certificates.
- We recommend you protect the C2W port from external access – it only needs to be reachable from Call Bridges.
- A Call Bridge must be able to uniquely reach the C2W interface of each Web Bridge it is configured to work with (C2W connections must use unique hostname or IP per Web Bridge 3 instance).
- Web app clients will have a single address to reach the Web Bridge so when multiple Web Bridges are used, DNS or Load Balancer solutions should be used to direct a shared name to an available Web Bridge instance. The client to Web Bridge connection is stateless for non-call activity and a session does not need to stay with a single Web Bridge.
- When establishing the TLS connection, both sides must present a certificate to verify. The Call Bridge uses the certificate set using the **`callbridge certs`** command and the Web Bridge uses the certificate set using the **`webbridge3 c2w certs`** command.
- The Web Bridge will trust certificates of Call Bridges and Schedulers that are in the Web Bridge's C2W trust store or have been signed by a certificate in the trust store, set by **`webbridge3 c2w trust`**. It is recommended to use a bundle containing the Call Bridge certificates that will connect to this Web Bridge so that only specific certificate matches will be allowed (certificate-pinning).
- The Call Bridgewill trust certificates of Web Bridges that are in the Call Bridge's C2W trust store or have been signed by a certificate in the trust store, set by **`callbridge trust c2w`**. It's recommended to use a bundle containing the Web Bridge certificates that this Call Bridge will connect so that only specific certificate matches will be allowed (certificate-pinning).
- The Scheduler trusts certificates of Web Bridges that are in the Scheduler's C2W trust store or have been signed by a certificate in the trust store, set by the command **`scheduler c2w certs <key-file> <crt-fullchain-file>`**.
- If the certificates used for C2W or Call Bridge have extended key usages defined, they must have the usages enabled to allow a Mutual TLS authentication exchange between Call Bridge and Web Bridge. If extended key usages are defined in a certificate, the Web Bridge 3 C2W certificate must include the " server authentication" extended key usage, and the Call Bridge certificate must include " client authentication" extended key usage. If no extended key usages are defined in a certificate, all usages are assumed valid.
- As the C2W connection is only between internal services, you do not explicitly need to use a certificate signed by a public authority. You can use self-signed certificates created

within the MMP.

- The SAN/CN in the Web Bridge C2W certificate must match the FQDN or IP address that is used in the c2w:// url used to register the Web Bridge 3 in the Call Bridge API. If this does not match, the Call Bridge will fail the TLS negotiation, rejecting the certificate presented by the Web Bridge, and will fail to connect with the Web Bridge.

  Note: If you want a certificate signed by a Public CA you will need to use the FQDN. (Certificates containing an IP address cannot be signed by a Public CA.) If you want to use an IP address in the C2W address you can create your own certificates as the C2W connection is not a public connection, therefore using Public CAs is not necessary.

- The certificate used for the Web Bridge listening interface should be signed by a certificate authority the clients will trust to avoid certificate warnings when clients connect. The FQDN the clients use to reach Web Bridge should be in the certificate CN or SAN list to avoid certificate warnings when clients connect.
- For general certificate information, see the Certificate Guidelines appropriate for your deployment.

4.6.2 Enabling the Web Bridge 3 Service

The Web Bridge service should be enabled on the Core Meeting Server instance if using the Cisco Expressway proxy or supporting web app clients who can reach the Call Bridge directly. When using the Meeting Server Edge deployment, Web Bridge 3 should run on all Edge instances and can optionally be ran on the Core Meeting Server instance where Call Bridge is running.

Complete these steps on each Meeting Server instance where Web Bridge 3 will run.

1. SSH into the MMP and log in.
2. Configure the interface and port web bridge will use for the web server with the command **`webbridge3 https listen <interface>:<port>`**.

   Using the first interface and port 443 is recommended. Example:

   ```
   webbridge3 https listen a:443
   ```

3. Set the HTTPS certificate and key pair Web Bridge will use for its web server with the command **`webbridge3 https certs <key file> <full certificate chain file>`**.

   This command requires the certificate be defined as the full certificate chain - meaning a certificate bundle that starts with the end entity certificate, includes all the intermediate signing certificate authorities, and ends with the root certificate. Example:

   ```
   webbridge3 https certs wb3-https.key wb3-https-fullchain.crt
   ```

4. Configure the interface and port for the C2W connection with the command

   `webbridge3 c2w listen <interface>:<port>` .

Using the first interface and the default example port 9999 is recommended. Example:

```
webbridge3 c2w listen a:9999
```

5. Configure the C2W connection certificates with the command **`webbridge3 c2w certs <key file> <full certificate chain file>.`**

   Example:

   ```
   webbridge3 c2w certs wb3-c2w.key wb3-c2w-fullchain.crt
   ```

   Note: This certificate must include the FQDN or IP address of the C2W interface in the CN or SAN list of the certificate. Additional information is also available in this FAQ - How do I configure connection certificates for use with Web Bridge 3?

6. The Web Bridge 3 C2W trust store must be configured to control which Call Bridge will be allowed to connect to this Web Bridge. The trust bundle should include the Call Bridge certificate of all Call Bridges that will connect to this Web Bridge, or the certificate of the CA that signed the Call Bridge certificates. For the most control, it is recommended to use the individual Call Bridge certificates in the bundle (certificate-pinning) rather than the certificate of the signing authority. Configure the web bridge's c2w trust bundle with the command **`webbridge3 c2w trust <certificate bundle>`**Example:

   ```
   webbridge3 c2w trust wb3-c2w-trust-bundle.crt
   ```

7. Enable the http redirect. This is optional, but recommended for end-user ease of use

   ```
   webbridge3 http-redirect enable
   ```

8. Enable the web bridge service

   ```
   webbridge3 enable
   ```

Repeat the above steps for each Meeting Server instance where Web Bridge will be running and ensure the certificate or key pairs used are correct for each instance.

4.6.3 Configuring Call bridge C2W connections

C2W is the control interface between the Call Bridge and Web Bridge instances and must be configured in the Call Bridge if Web Bridge is deployed. The Call Bridge's C2W trust bundle should include the Web Bridge C2W certificates of all Web Bridge that this Call Bridge will connect to, or the certificate that signed the Web Bridge C2W certificates. For the most control, it is recommended to use the individual Web Bridge C2W certificates in the bundle (certificate-pinning) rather than the certificate of the signing authority.

1. Connect to the MMP interface of the Internal Meeting Server running Call Bridge.
2. The Call Bridge should already be configured with a certificate from the steps performed in Configuring the Call Bridge listening interface. Confirm by running the command **`callbridge`** and checking that the Key File and Certificate file settings are configured. If

not, repeat the steps in Configuring the Call Bridge listening interface before proceeding. The Call Bridge must be configured with certificates for C2W functionality.

3. Use the command **`callbridge trust c2w <certificate bundle file>`** to configure the Call Bridge's C2W trust store with a certificate bundle that includes the C2W certificates of the Web Bridge instances. Example:

   ```
   callbridge trust c2w c2w-callbrige-trust-store.crt
   ```

   Note: Unless limited by scopes, the Call Bridge will attempt to connect to all Web Bridge that are defined in the Meeting Server API.

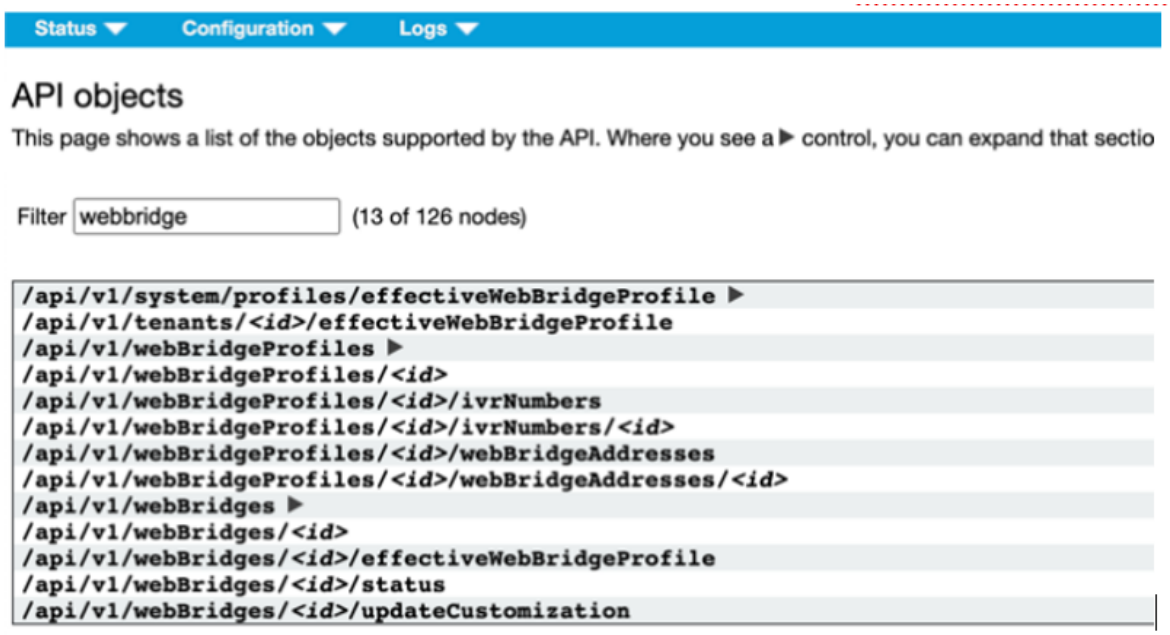
4. Restart the Call Bridge

   ```
   callbridge restart
   ```

4.6.4 Configure Call Bridge with Web Bridge Addresses

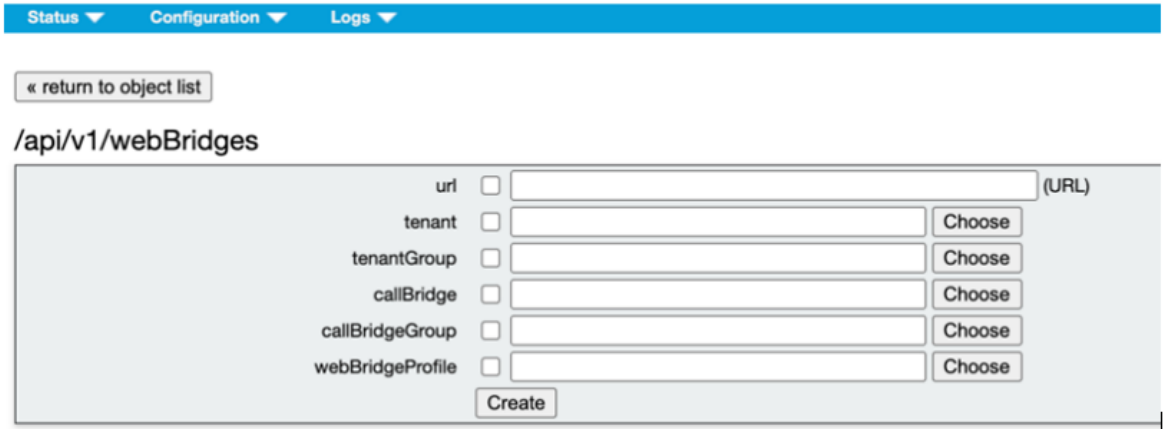
The Call Bridge must be told the C2W address of each Web Bridge it will connect to (including a co-resident Web Bridge) by creating a Web Bridge entry in the Meeting Server API. This guide will use API explorer in the Web Admin interface of Meeting Server to illustrate how to complete this task.

1. Log in to the Meeting Server Web Admin interface and select **Configuration** > **API.**
2. Using the Filter input box, type **`webBridges`** to filter the list view, as shown here:



3. Locate the **`/api/v1/webBridges`** row from the resulting list and click the ▶ icon to expand it.

4. Click **Create new** to create a new Web Bridge object and the following parameter fields display as shown here:



5. Fill in the **url** field using the format *c2w://<Web Bridge FQDN>:<c2w port>* with the FQDN address of the C2W interface for Web Bridge being added. Example:

```
c2w://cmsedge1.company.com:9999
```

Note: The FQDN entered here must be the CN or in the list of SAN names of the certificate assigned to the C2W interface of Web Bridge 3 and must resolve to the IP of the C2W interface for the Web Bridge. IP Addresses can only be used if the C2W certificate has the IP address in the certificate's SAN or CN.

---

6. Click **Create** to save the new Web Bridge entry.

If you have multiple Web Bridges, repeat the above steps creating one Web Bridge object for each Web Bridge instance.

4.7 Configuring the Email server for Scheduler

This section describes the steps to configure the Email server for the Scheduler component. Email notifications are sent to the participants when a meeting is scheduled, canceled, or modified. Scheduler supports sending the email notifications via configuration of an SMTP email server.

The configuration of the server address and port, enabling email protocol, and configuring a username for authentication are specified via the following scheduler MMP commands:

```
scheduler email server <hostname|address> <port>
scheduler email server none
scheduler email username <smtp username>
scheduler email protocol <smtp|smtps>
```

```
scheduler email auth <enable|disable>
scheduler email starttls <enable|disable>
```

Email will not be configured on a scheduler if no server address is configured on it. At least one email server must be configured for the scheduler to send email invites. Emails can be sent from any scheduler and not necessarily from the scheduler which was used to schedule the meeting. If an email server is down, then a different scheduler sends the email.

If the Meeting Server does not receive a response from the Exchange email server within 5000 milliseconds (5 seconds), the Scheduler will automatically reject the request.

Scheduler supports the following types of email configurations:

1. SMTP
2. SMTP with Authenticated Login (Auth Login)
3. SMTP and STARTTLS
4. SMTP with Auth Login and STARTTLS
5. SMTPS (end to end TLS Encryption for the entire SMTP transaction)
6. SMTPS with Auth Login

Note: It is recommended to use Exchange Server 2016 CU22 - 15.1.2375.7 and Exchange Server 2019 CU11 - 15.2.986.5.

From version 3.4 meeting invites can be sent to all the participants from a common email address. The MMP command **`scheduler email common-address <address@mail.domain> "<Display name>"`** configures the common email address and a display name on the Meeting Server. The Scheduler sends the meeting invites from the common email address to the participants.

If the common email address is left blank, the Scheduler sends the email invites from the organizer's email address.

Note: If common email address is not configured, authentication with the SMTP server requires an email address to be configured using the MMP command **`scheduler email username <smtp user-name>`**. This account configured on the MMP must have appropriate permissions to be able to send emails on behalf of web app users.

The organizer's name can also be included to appear as display name besides the email address to identify the sender. When a meeting is scheduled using web app, web app sends the name of the user scheduling the meeting as the organizer display name, to the scheduler. A name of choice can be set as display name by including the optional parameter **`organizerDisplayName`** in the scheduler API.

Note: On the Meeting Server 2000 appliance, after configuring and enabling the Scheduler, it is required to execute the MMP commands 'webbridge3 restart' and 'scheduler restart'. The '**Schedule Meeting**' option will appear in the Meeting Server Web App only after these services have successfully restarted.

To enable the Scheduler to send email notifications via the SMTP, configure the email server to listen on a specified port for the SMTP protocol.

1. Disable the Scheduler component if it is currently running:

   ```
   scheduler disable
   ```

2. Configure the email server and port:

   ```
   scheduler email server <hostname|address> <port>
   ```

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

3. Enable the Scheduler:

   ```
   scheduler enable
   ```

To enable the Scheduler to send email notifications via the SMTP with Auth Login, configure the email server to listen on a specified port for the SMTP protocol, enable the SMTP server to support Auth Login, and configure a user account for authentication. This account configured on the MMP must have appropriate permissions to be able to send emails on behalf of web app users.

1. Disable the Scheduler component if it is currently running:

   ```
   scheduler disable
   ```

2. Configure the email server and port:

   ```
   scheduler email server <hostname|address> <port>
   ```

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

3. Enable the Auth Login option:

   ```
   scheduler email auth enable
   ```

4. Set the username to be used for authentication:

   ```
   scheduler email username <username>
   ```

   Enter the password:

   ```
   scheduler email username test@test.com
   Please enter password:
   ```

```
Please enter password again:
```

5. Enable the Scheduler:

   ```
   scheduler enable
   ```

To enable the Scheduler to send email notifications via the SMTP and STARTTLS, configure the email server to listen on a specified port for the SMTP protocol and enable STARTTLS.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

1. Disable the Scheduler component if it is currently running:

   ```
   scheduler disable
   ```

2. Configure the email server and port:

   ```
   scheduler email server <hostname|address> <port>
   ```

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

3. Enable the STARTTLS option:

   ```
   scheduler email starttls enable
   ```

4. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

   ```
   scheduler email trust <cert or bundle name>
   ```

   The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

5. Enable the Scheduler component:

   ```
   scheduler enable
   ```

To enable the Scheduler to send email notifications via the SMTP using Auth Login and STARTTLS, configure the email server to listen on a specified port for the SMTP protocol. Additionally, enable the SMTP server to support Auth Login, configure a user account that will be used for authentication, and enable STARTTLS.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and

establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

1. Disable the Scheduler component if it is currently running:

   ```
   scheduler disable
   ```

2. Configure the specified email server and port:

   ```
   scheduler email server <hostname|address> <port>
   ```

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

3. Enable the Auth Login option:

   ```
   scheduler email auth enable
   ```

4. Set the username to be used for authentication:

   ```
   scheduler email username <username>
   ```

   Enter the password:

   ```
   scheduler email username test@test.com
   Please enter password:
   Please enter password again:
   ```

5. Enable the STARTTLS option:

   ```
   scheduler email starttls enable
   ```

6. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

   ```
   scheduler email trust <cert or bundle name>
   ```

   The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

7. Enable the Scheduler component:

   ```
   scheduler enable
   ```

To enable the Scheduler to send email notifications via the SMTPS, configure the email server to support end to end SMTP encryption on a specific port.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email

server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

1. Disable the Scheduler component if it is currently running:

   ```
   scheduler disable
   ```

2. Configure the specified email server and port:

   ```
   scheduler email server <hostname|address> <port>
   ```

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

3. Set the email protocol to SMTPS:

   ```
   scheduler email protcol smtps
   ```

4. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

   ```
   scheduler email trust <cert or bundle name>
   ```

   The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

5. Enable the Scheduler component to complete the email configuration using SMTPS:

   ```
   scheduler enable
   ```

To enable the Scheduler to send email notifications via the SMTPS using Auth Login, configure the email server to support end to end SMTP encryption on a specific port. Additionally, enable the SMTPS server to support Auth Login and configure a user account that will be used for authentication.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

1. Disable the Scheduler component if it is currently running:

   ```
   scheduler disable
   ```

2. Configure the specified email server and port:

   ```
   scheduler email server <hostname|address> <port>
   ```

   For example,

```
scheduler email server exchange.example.com 25
scheduler email server 10.27.33.55 25
```

3. Enable the Auth Login option:

   ```
   scheduler email auth enable
   ```

4. Set the username of the user which will be used for authentication:

   ```
   scheduler email username <username>
   ```

   Enter the password:

   ```
   scheduler email username test@test.com
   Please enter password:
   Please enter password again:
   ```

5. Set the email protocol to SMTPS:

   ```
   scheduler email protcol smtps
   ```

6. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

   ```
   scheduler email trust <cert or bundle name>
   ```

   The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

7. Enable the Scheduler component to complete the email configuration using SMTPS with Auth Login:

   ```
   scheduler enable
   ```

4.7.1 Scheduler detailed logging

The Scheduler supports the option to enable detailed logging for Web Bridge connections, email notifications, and API using the scheduler timedLogging MMP command.

When timedLogging is not enabled, Meeting Server displays the following output:

```
cms-vm> scheduler timedLogging
{
"webBridge": "0",
"api": "0",
"email": "0"
}
```

To enable any of the timedLogging options, use the command:

```
scheduler timedLogging (webBridge|api|email) <time>
```

For example,

```
cms-vm> scheduler timedLogging webBridge 600
SUCCESS
```

The time variable is expressed in seconds, and enables timedLogging for the set duration.

```
cms-vm> scheduler timedLogging
{
"webBridge": "594",
"api": "0",
"email": "0"
}
```

After the set duration expires or the specific investigation or troubleshooting step is completed download the log files using SFTP.

4.8 Configuring the TURN Server

The TURN Server is used to provide media traversal services for web app users who cannot directly reach the Call Bridge.

- If you are not using the web app client in your deployment, you can skip this section.
- If you are using Cisco Expressway as your web proxy and TURN provider, please use the Cisco Expressway Web Proxy for Cisco Meeting Server (X14.3) for instructions on configuring the TURN Server and Call Bridge settings instead of this section.
- If you are using a Meeting Server Edge deployment, the TURN Server should be configured on each of your Edge instances. Complete the steps in this section to configure the TURN services.

Complete the following sections to configure a TURN Server and add it to Call Bridge.

4.8.1 Enable the TURN Service

1. SSH into the MMP and log in.
2. Enable the short term credential mode for TURN Server. Introduced in version 3.1, short term credentials offer a significant increase in security over the previously used static TURN Server credentials. The TURN credentials are used to control who can request a relay on the TURN Server and are automatically given to web app clients during call setup to allow use of the TURN Server. It is recommended that all deployments usingMeeting Server Edge

enable the short term credential mode. Enter the following command to enable short term credential mode:

```
turn short_term_credentials_mode enable
```

3. Set the shared secret and realm for the TURN Server's short term credential feature using the command:

   `turn short_term_credentials <shared secret> <realm>`

   These two values can be any string and should be treated like passwords. These values will also be needed when defining the TURN Server in the Call Bridge Settings. Example:

   ```
   turn short_term_credentials mysharedsecret example.com
   ```

   CAUTION: Your TURN Server password and credentials must be unique. Do not reuse your admin username or password.

4. If the TURN Server's listening interface is located behind a NAT relative to the Internet/external network, tell the TURN Server the public IP Address that maps to the TURN Server using the command:

   `turn public-ip <ip address>`.

   If your TURN Server uses a public routable IP address, skip this step. Example:

   ```
   turn public-ip 5.10.20.99
   ```.

5. Configure the TURN Server to listen on a specific interface using the command **`turn listen <interface allowed list>`**. You should configure TURN to listen on the first interface 'a' along with Web Bridge. Example:

   ```
   turn listen a
   ```

6. If enabling TURN TCP on 3478, configure the TCP port the TURN Server should use with the **`turn tls <port|none>`** command.

   Example:

   ```
   turn tls  3478
   ```

   The example assumes you are using the TCP 3478 port. If you are not enabling TURN TCP, skip this step.

7. If enabling TURN TCP, the TURN Server must be configured with a certificate and key pair to use. The certificate should be signed by the same CA that signed the Web Bridge's certificate. If you are not enabling TURN TCP, you may skip this step. Configure the TURN Server's certificate with the command **`turn certs <key file> <certificate file> <ca cert>`**. Example:

   ```
   turn certs turnCert.key turnCert.crt CAbundle.crt
   ```

Note: The certificate used for TURN Server can be an existing certificate such as Web Bridge 3 certificate.

8. Enable the TURN Server

```
turn enable
```

If multiple Edge Server instances are to be used, repeat the above TURN configuration steps for each Edge Meeting Server instance ensuring the certificate/ key pairs used are correct for each instance.

4.8.2 Configure Call Bridge with TURN Addresses

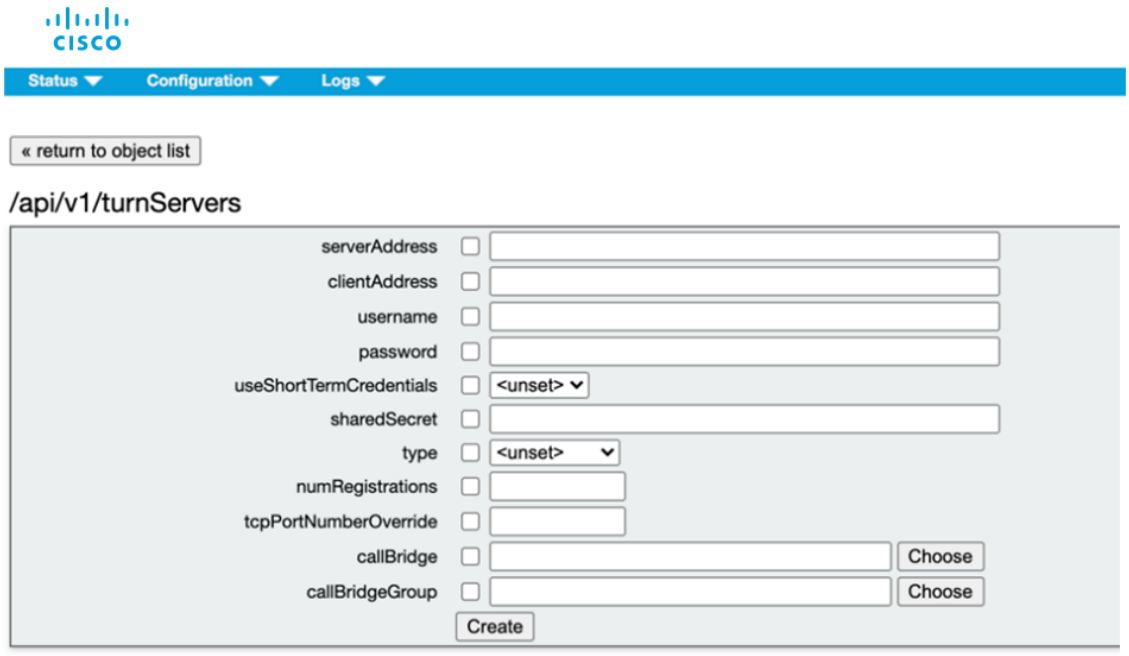
The Call Bridge must be configured with the details of the available TURN Servers to use. These TURN configurations are only used for web app participants and Skype for Business call flows. See section Dial plan configuration – integrating Lync/Skype for Business for details on configuring Skype for Business support.

The Call Bridge must be told the TURN Server it can use by creating a **turnServers** entry for each TURN Server in the Meeting Server API. This guide will use API explorer in the Web Admin interface of Meeting Server to illustrate how to complete this task.

1. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
2. Using the Filter input box, type **turn** to filter the list view, as shown here:



3. Locate the **/api/v1/turnServers** row from the resulting list and click the ▶ icon to expand it
4. Click **Create new** to create a new **turnServer** object and the following parameter fields will be shown:



5. Fill in the following fields for the TURN Server being added:

   serverAddress - Fill in the IP address or the DNS name of the TURN Server only if the Call Bridge has to connect TURN Server's listening port else provide a dummy address so the Call Bridge will not attempt to contact the TURN Server - Example: nothing.local

   clientAddress - Fill in with the IP address or DNS name that external clients will use to reach the TURN Server

   Note: If TURN is NAT'd, enter the public NAT address). For example: 128.8.5.2

   useShortTermCredentials - Set to **true** if you configured the TURN Server to use short term credentials in the prior section (recommended).

   sharedSecret - Enter the sharedSecret string used when the TURN Server was configured in Step 3 of the previous section.

   type - If this parameter is unset, it defaults to " standard" , and tells the clients to use UDP 3478 and fallback on TCP 443 to connect to the TURN server. When deploying Meeting Server web edge, this parameter should be set to " cms" .

   tcpPortNumberOverride - If you configured TURN TCP on a port other than 443, enter the port number configured with the turn tls command.

   Note: Using this configuration can generate a status that the Call Bridge cannot connect to the TURN Server due to the dummy address in the serverAddress field. This is a known issue but does not impact the deployment.

   ---

6. Click **Create** to save the new TURN Server entry.

If you have multiple TURN Servers, repeat the above steps creating a TURN Server object for each TURN Server instance.

4.9 LDAP authentication for MMP users

The new **`ldap`** option is added to the **`user add`** MMP command enables you to configure details of an LDAP server, directory search parameters, TLS settings, and enable or disable LDAP authentication. During Meeting Server deployment, administrators and web app users with LDAP user accounts can log in to Web Admin Interface, SSH, SFTP, and serial console using LDAP authentication. User login will be rejected in case of failure in LDAP authentication.

Note: For Common access card (CAC) deployments, CAC authentication takes precedence over both, LDAP authentication and local authentication.

This feature does not support importing MMP users via LDAP, or converting existing local users to LDAP authenticated users. Administrator has to manually pre-configure the LDAP users by adding each user with the MMP command **`user add`**. Ensure that the login names are unique for local and LDAP users. To add LDAP users, a new option [**`ldap`**] is added to the command:

```
user add <username> (admin|crypto|audit|appadmin|api) [ldap]
```

Note: Meeting Server API does not support access to users with LDAP authentication.

The authentication of the users added using the **`ldap`** option is done by the LDAP server. No look up for local password is done in this case. In case of local users, authentication is done using a local password lookup only. LDAP authentication does not support password changes.

Note: In case the LDAP server becomes unavailable or Meeting Server is unable to reach the LDAP server, then LDAP users will be unable to log in. As a backup, it is a good practice to always keep at least one local admin user configured on the MMP.

Meeting Server supports configuration of a Microsoft AD LDAP server or an Open LDAP server, with either one of hostname/IPv4/IPV6, along with port, using the new **`ldap`** option. This LDAP server can be the same as the one used for web app user authentication. Make sure the LDAP server being used is a supported server type and it has to be configured separately for Meeting Server.

See the MMP Command reference guide for details.

5 Configuring the Databases

You do not need to create or enable databases as happens for other components: an empty database is created on every host server when you install the Meeting Server software image on the server.

Database clustering works differently to Call Bridge clusters. A database cluster creates what is essentially an 'online' backup of the running database which is maintained as the system runs. It also provides the ability to move to using the backup in an automated fashion in the event of a failure being detected.

During the clustering process you select the node with the primary database, and then add 'replica' database nodes to the cluster. Make sure that you have atleast three database nodes in the cluster. Do not have a cluster of 2 database nodes, if you do and there is a failure on the primary node, the remaining replica node will not be able to check whether it's safe to promote itself to primary and continue servicing the database requests. By adding a third database node, the database cluster has a way to help determine where the failure is, and if it's safe to elect a different database as primary, enabling the Meeting Server operations to continue uninterrupted.

Only 3 Cisco Meeting Servers can be joined to a database cluster and any further servers should be added to the cluster using **`database cluster connect`** command.

Database clustering does not do any kind of load balancing, or any caching. Nor does it perform any sharding of data for more efficient local access within any kind of geographically distributed arrangement. All queries are directed at the current primary database, where ever it is. The replicas are not available as read-only instances.

Note: There is a network latency limitation (or Round Trip Time) of 200 ms or less between the database servers, and between the Call Bridge and the primary database.

Follow the instructions in this section to create the cluster(s). Unless otherwise noted, these instructions apply equally to combined or split deployments.

Note: If a WAN optimizer is deployed between clustered database nodes, it may prevent keep-alive checks from completing, causing errors to appear in logs. In cases where a WAN optimizer is being used between cluster nodes, it is important to ensure that all keep alive traffic is sent in a timely manner.
Please consult your WAN optimizer documentation about how to either disable this functionality between specific IP addresses, or for options that control which optimizations are applied.

5.1 Database on a Separate Server

5.1.1 Requirements for a database on a separate server

Note: This section is applicable only if you choose to use one or more external databases.

The host server for a database has modest CPU requirements, but requires large storage and memory. We do not mandate a qualified VM host but recommend the specification in this link. In addition:

- The data store should reside on either a high IO per second SAN or local SSD storage
- The data must reside on the same vdisk as the OS

The Cisco UCS C220 M4 which is currently used as the host for the Cisco Meeting Server 1000 could be used, but the VM database would only use a small percentage of the server's resources. Using this server, other VMs could be also hosted on the same server as the VM database, if desired.

It should be possible to run other VMs on the same host server, if desired.

5.1.2 Deploying a database on a separate server

1. Install the Meeting Server image on to each of the external database host servers. An empty database is set up automatically.
2. The host servers will require certificates – see the next section.

5.2 Deploying and validating Certificates on the Database and Call Bridge Servers

Database clustering uses public/private key encryption for both confidentiality and authentication, with a single, shared Certificate Authority (CA). If no certificates are used then there is no confidentiality nor authentication. Because the database clustering is not user-accessible, the certificates can be signed by a local CA. Refer to the Certificate guidelines for information on creating, uploading and assigning the certificates and certificate bundles to the database cluster.

Note: In any production environment, you must use encryption on database traffic. This is achieved by using certificates. However, for testing (and only for testing) you can skip using certificates. If you do not use certificates then there is no security nor access control for the database.

CAUTION: Certificates can only be assigned to a disabled database cluster. If you have already set up a database cluster you must run the **`database cluster remove`** command on every server in the cluster, then run the commands to upload and assign the certificates to the host servers (see the Certificate guidelines), before re-creating the cluster using the steps in the following sections.

Meeting Server validates the nodes by checking the certificate chain up to the root certificate. From version 3.5, the identity(hostname/IP address) of the Meeting Server that is added to the cluster can also be validated. The Meeting Server along with other validations, verifies if the server identity matches with the name stored in the server certificates.

The **`database cluster verifymode <full/ca>`** command allows you to set the preference for the validations accordingly. If the command is set to **`full`**, the Meeting Server along with other validations, verifies if the server identity matches with the name stored in the server certificates. While, if the command is set to **`ca`**, the Meeting Server will validate only the Certificate Authority.

If you choose to validate the server identity by setting the verify mode to full, ensure that:

1. The server's name must match an exact FQDN on its corresponding server certificate.
2. If your deployment does not have a DNS record, a DNS RR record must be created to resolve the server identity locally.
3. While adding the replica database nodes to the cluster:
    a. If the replica database nodes are added using the hostname, then the hostname must be added in the CN or SAN list of the Meeting Server database server certificate.
    b. If the replica database nodes are added using the IP address, then the IP address must be added in the SAN list of the Meeting Server database server certificate.
    c. All the replica databases in a cluster must be added using the same joining method: hostname or the IP address.
4. If IP address is used in the server certificate, then, while generating the SAN for the certificate, the IP address must be added as DNS name in SAN along with the IP address field.

    For example:

    `pki csr dbserver CN:server.db.example.com subjectAltName:10.1.1.1`

    The above command will generate the following SAN entries in the certificate signing request:

    `IP Address: 10.1.1.1, DNS:10.1.1.1, DNS:server.db.example.com`

    See Certificate Guidelines for more information on creating and generating the certificates.

5. The server identity of all the nodes in the cluster must match the IP address in the server certificate.

Note:
- The verify mode must be set to the same value(full/ca) on all the nodes in a cluster.
- The verify mode can be modified only on unclustered database. If you have already set up a database cluster, you must run the **`database cluster remove`** command on every server in the cluster and then set the preference for the verify mode.

5.3 Selecting the Primary Database for a Cluster

To deploy a database cluster, decide which will be the primary database (that is, the database instance that will be used by all Call Bridges initially). If you have been deploying without scalability, initially the primary database must be the current database so that no data is lost. Therefore this database will be co-located with a Call Bridge.

Note: A single database can be a "cluster" in that it can have one or more Call Bridges using it ("attached" to it). However, there is no resilience.

3. On the server with the database that will start as primary, sign in to the MMP.
4. If you have not already configured the database cluster certificates, then set the certificates using the command:

   `database cluster certs <server.key> <server.crt> <client.key> <client.crt> <ca.crt>`

   `database cluster certs db01server.key db01server.crt db01client.key db01client.crt db01cert-bundle.crt`
5. Enter the following command to select the interface for this database cluster:

   `database cluster localnode <interface>`

   `database cluster localnode a`

   The <interface> can be in the following formats:
   - [a|b|c|d|e] – the name of the interface (the first IPv6 address is preferred, otherwise the first IPv4 address is chosen) e.g. **`database cluster localnode a`**
   - ipv4:[a|b|c|d|e] – the name of the interface restricted to IPv4 (the first IPv4 address is chosen) e.g. **`database cluster localnode ipv4:a`**
   - ipv6:[a|b|c|d|e] – the name of the interface restricted to IPv6 (the first IPv6 address is chosen) e.g. **`database cluster localnode ipv6:a`**
   - <ipaddress> – a specific IP address, can be IPv4 or IPv6 e.g.

```
database cluster localnode 10.1.3.9
```

- Do NOT use the Admin interface for database clustering.

6. Optionally, you can validate the server identity(hostname/IP address) with the name stored in the server certificates. Set the preference for validation using the command:

   `database cluster verifymode <full/ca>`

   `ca` - Meeting Server validates the node from the certificate chain upto the root certificate without validating the server identity.

   `full` - Meeting Server along with other validations, verifies if the server identity matches with the name stored in the server certificates.

   Note: If the command is not used, the Meeting Server will validate only the Certificate Authority.

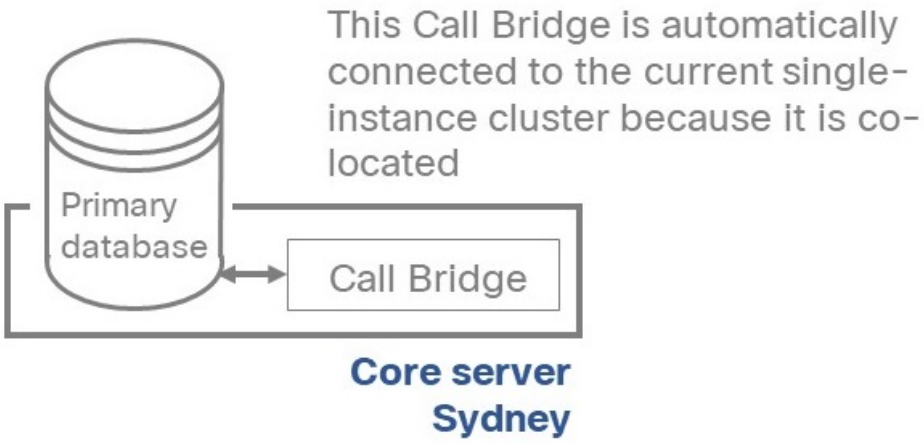
   ---

7. Enter the MMP command: **`database cluster initialize`** and press **`Y`** in response to the prompt to initialize this as the primary databasefor this database cluster.

```
database cluster initialize
WARNING!!!
Are you sure you wish to initialize this node as a new database cluster?
(Y/n)
The contents of this node's database will become the primary version of the
database in the new cluster.
Initialization started...
```

   This triggers a restart of the local Call Bridge and takes approximately 30 seconds.

Figure 15: Co-located Call Bridges are automatically connected



8. Check that the initialization completed correctly by entering the following command until the Status is reported as Enabled:

   ```
   database cluster status
   ```

   You should see messages similar to:

   ```
   Status: Initializing
   Nodes: 10.1.2.3 (me)       : Connected Primary
   Interface                  : a
   ```

   And later if you re-run the status command:

   ```
   Status: Enabled
   Nodes: 10.1.2.3 (me)       : Connected Primary
   Interface                  : a
   ```

5.4 Attaching other Database Instances to the Database Cluster

Note: These server(s) can have an empty database and do not need to have a co-located Call Bridge e.g. virtualized servers set up to be external databases only. These host servers require the database cluster certificates and keys.

CAUTION: The contents of the database currently on this server (if any) will be destroyed.

If you have not already configured the database cluster certificates, then set the certificates using the command:

```
database cluster certs <server.key> <server.crt> <client.key>
<client.crt> <ca.crt>
```

```
database cluster certs db01server.key db01server.crt db01client.key
db01client.crt db01cert-bundle.crt
```

1. Attach other servers hosting a database that you want to be part of this database cluster.
   a. On each such server set the listening interface using the following command:

      ```
      database cluster localnode <interface>
      ```

      Note: <interface> can be in any of the formats listed previously in this section

      For example:

      ```
      cms> database cluster localnode a
      Interface updated
      ```

   b. Optionally, you can validate the server identity (hostname/IP address) with the name stored in the server certificates. Set the preference for validation using the command:

      ```
      database cluster verifymode <full/ca>
      ```

ca - Meeting Server validates the node from the certificate chain upto the root certificate without validating the server identity.

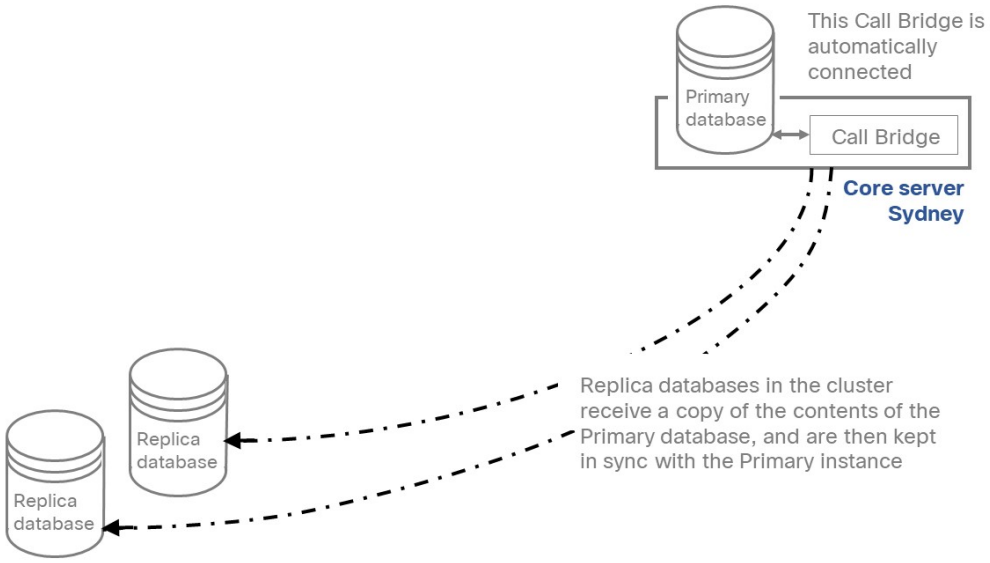
full - Meeting Server along with other validations, verifies if the server identity matches with the name stored in the server certificates.

Note: If the command is not used, the Meeting Server will validate only the Certificate Authority.

---

c. "Attach" this database to the primary database using the command **database cluster join <primary hostname/IP>** and then use the **database cluster status** command to monitor the progress, as shown below.

Figure 16: Attaching databases



Note: A hostname can be used in the **database cluster join <primary hostname/IP>** command below but it will be replaced by the IP address of the interface specified for the primary database.

The attach command pulls a full copy of the primary database onto this server, and therefore may take some time depending on the connection speed. For an empty database, this operation is expected to take approximately 30 seconds.

```
cms> database cluster join 10.1.2.3
WARNING!!!
Are you sure you wish to attach this node to an existing database
cluster? (Y/n)
```

```
The contents of this node's database will be destroyed!
Attachment started...
cms> database cluster status>
Status    ;           : Attaching
Nodes:
    10.1.2.3          : Connected Primary
    10.1.2.8 (me)     : Connected Replica
Interface             : a
cms> database cluster status
Status                : Enabled
Nodes:
    10.1.2.3          : Connected Primary
    10.1.2.8 (me)     : Connected Replica
Interface             : a
```

This triggers a restart of the local Call Bridge (if there is one).

d. Verify that the primary database is aware of the attached database by entering the **database cluster status** command in the MMP of the primary database host server. (This information should have propagated automatically within 10 seconds of the join command completing.)

```
cms> database cluster status
Status                : Enabled
Nodes:
    10.1.2.3 (me)     : Connected Primary
    10.1.2.8          : Connected Replica
Interface             : a
```

5.5 Connecting Remote Call Bridges to the Database Cluster

Call Bridges that are co-located with a database (primary or replica) are automatically connected to the database cluster that the co-located database is part of.

Note: The **database cluster connect** command below does not have to be run if **database cluster initialize** or **database cluster join** has already been run on this host server. You can check by running **database cluster status**, the server will be in the list of nodes.

"Connection" means that the Call Bridge knows how to access all the databases in the cluster; therefore it does not matter which database's address is used to connect. (The actual database that is read from/written to is the current primary database).

1. Sign in to the MMP of the Core server with an unconnected Call Bridge and issue the command **database cluster connect <hostname/IP>**. The hostname or IP address can be for any database in the cluster.

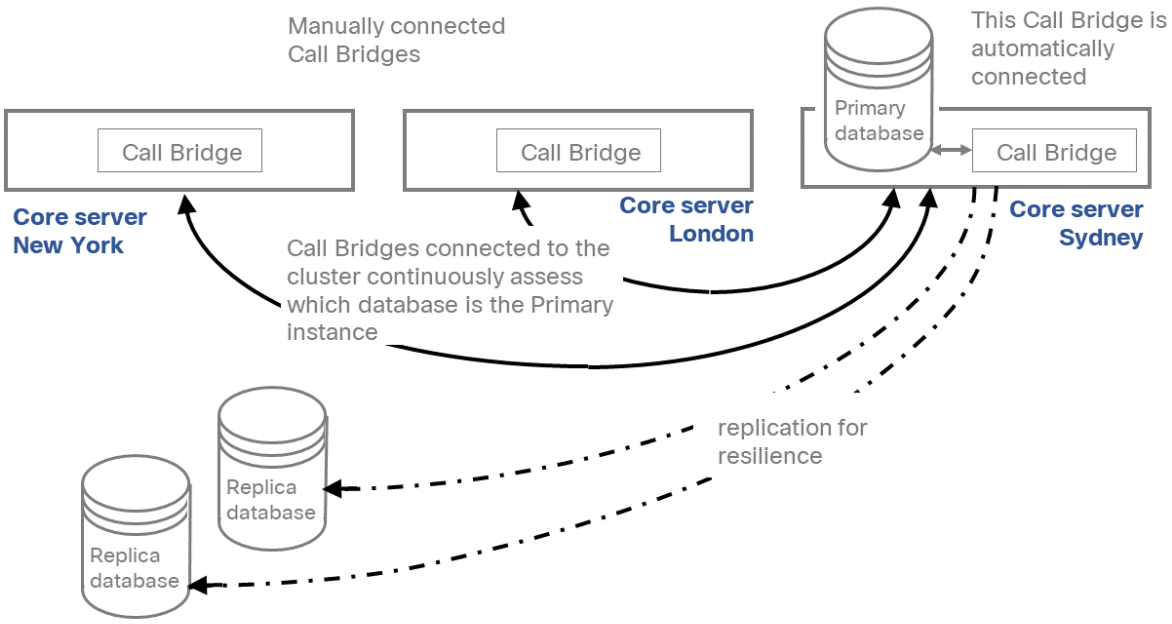
```
cms> database cluster connect 10.1.2.3
WARNING!!!
Are you sure you wish to connect this node to an existing database cluster?
```

```
(Y/n)
Connecting started...
```

This triggers the Call Bridge on this Core server to restart.

Note: Unlike the attach command used in the previous section, the connect command does not delete any existing database on the server hosting this Call Bridge. Therefore if you use the command on a Core server with an existing local database then the contents of the database are not destroyed, but the local database is invisible until this server's Call Bridge is disconnected from the cluster.

Figure 17: Example of database clustering and Call Bridge connections



5.6 Upgrading the database schema

Note: This section is **not** required when initially configuring database clustering or if the database cluster is disbanded during an upgrade (which is the recommended practice). These steps should **only** be performed if the database cluster is **not** disbanded during the upgrade process or if a database schema version mismatch is reported in the event logs.

---

1. Check the **database cluster status**. A healthy database cluster status does not guarantee full system functionality. Even if all nodes report as healthy and synchronized, Call Bridges may fail to operate correctly if there is a database schema version mismatch. Verify the

system state by checking for the following indicators:

- **Event/Syslogs**: Look for errors related to database schema version mismatches.
- **Web Admin Interface**: Confirm that coSpaces are visible on each Call Bridge. If the interface shows no coSpaces, the Call Bridges are not communicating correctly with the database.

2. If you encounter the error, remote database has schema version **<version_number> (current version is version_number>**), follow these steps to resolve the mismatch:
   a. **Identify the Primary Database**: Determine which server is hosting the primary database (refer step 1).
   b. **Access the MMP**: Log in to the MMP of the server hosting the primary database.
   c. **Execute the Command**: Synchronize the schema Issue using **database cluster upgrade_schema** command.

5.6.1 Verify the database schema upgrade

a. **Verify Cluster Status**: Use the **database cluster status** command to verify the operation. All nodes should report as **healthy** and **in sync**.
b. **Review Logs**: Examine the Event/Syslogs to confirm there are no errors related to a database schema version mismatch.
c. **Validate Call Bridge Connectivity**: Access the Web Admin interface on each Call Bridge and verify that the list of coSpaces is correctly displayed.

5.7 Further information on database clusters

For further information relating to database clusters refer to the Frequently Asked Questions for the Cisco Meeting Server.

6 Deploying the Call Bridges

This chapter covers setting up and deploying multiple Call Bridges in a deployment. It covers:

- an overview of the certificates required for a Call Bridge,
- setting up the listening interface on a Call Bridge,
- how to cluster multiple Call Bridges,
- dial plan information,
- using Call Bridge grouping to load balance inbound and outbound calls on the Meeting Server in deployments with Cisco Unified Communications Manager or Cisco Expressway,
- information on Call Bridge integration with Lync Edge,
- selecting the appropriate Call Bridge mode to connect participants in dual homed conferences,
- a preview of the additional video streams over distribution links feature.

Unless otherwise noted, this instruction applies equally to combined or split multi-server deployments.

6.1 Setting up the Call Bridges' certificates

The Call Bridge needs a key and certificate pair that is used to establish TLS connections with SIP call control devices and with the Lync Front End server.

If you are using Lync, this certificate will need to be trusted by the Lync FE Server; the best way to achieve this is to sign the certificate on the CA server that has issued the certificates for the Lync FE Server.

Follow the instructions in the Certificate guidelinesfor information on creating, uploading and assigning certificates to Call Bridges.

6.2 Setting up the Call Bridges

If you have not already done so, on each Call Bridge:

1. Configure the Call Bridge's listening interface

   The command **`callbridge listen <interface>`** allows you to configure a listening interface (chosen from A, B, C or D). By default, the Call Bridge listens on no interfaces. A full list of commands is in the MMP Command Reference Guide.

   Configure listening interfaces as follows:

a. Configure the Call Bridge to listen on interface A.

```
callbridge listen a
```

b. Configure the Call Bridge to use the security certificates (created previously) by typing the following (so that a TLS connection can be established between the Lync FE server and the Call Bridge):

```
callbridge certs callbridge.key callbridge.crt
```

The full command and using a certificate bundle as provided by your CA, is described in the Certificate guidelines document.

c. Restart the Call Bridge to apply the changes.

```
callbridge restart
```

Note: You will need to add the Call Bridge certificate to every Web Bridge's trust store after you've configured the Web Bridges, as described in the Certificate guidelines.

6.3 Clustering Call Bridges

Within your Meeting Server deployment, you can enable Call Bridge clustering which will allow multiple Call Bridges to operate as a single entity and scale beyond the capacity of any single Call Bridge.

You have a choice whether to setup the Call Bridges in the cluster to link peer-to-peer, or for calls to route via call control devices between the clustered Call Bridges.

Linking Call Bridges peer-to-peer:

- reduces call complexity as the call will go from Call Bridge A to Call Bridge B directly, with nothing in the middle to interfere with the routing of the call.
- reduces load on the call control device, and frees up resources to handle calls that need to route through the call control device. This may be important if the call control device is licensed on a per call basis.

Routing via call control device(s):

- creates a consistent call flow for your Meeting Server and Local SIP devices. This can make network configuration a little simpler, particularly if there are firewalls between networks with fixed "allow rules" which only allow calls routed through call control devices.

How calls are routed in deployments with clustered Call Bridges is determined by the Peer Link SIP domain field (see below) and the dial plan (see Section 6).

A CallBridge cluster should have at least three nodes. In a CallBridge group, if the primary CallBridge becomes unavailable, calls will immediately fall back to the next available CallBridge. However, with fewer nodes, fallback may not be possible, and recovery could take approximately 15 minutes.

Follow these instructions to cluster Call Bridges:

Note: The instructions in this section assume that:
- all the databases are running as a cluster ,
- all the Call Bridges that will form part of the cluster are configured as standalone Call Bridges,
- all the Call Bridges have been connected to the database cluster.

On every Call Bridge that will be part of the Call Bridge cluster:

1. Sign in to the Web Admin Interface and go to **Configuration > Cluster**

   Note: Cluster will not appear in the **Configuration** drop-down list if you have not already created the database cluster.

   ---

2. In the Call Bridge Identity section, enter a unique name for that Call Bridge (e.g. " London-Core1") and click **`Submit`**.

   Note: The unique name must not contain any spaces. If there are spaces in the unique name then Call Bridge clustering will fail.

   ---

3. Enter a Peer link bit rate, the per call rate in which servers will connect at in a distributed call (optional)

On one Call Bridge that will be part of the cluster:

4. Sign in to the Web Admin Interface and go to **Configuration > Cluster**. In the table headed **Clustered Call Bridges**:

   a. Add an entry for this Call Bridge, using the Unique Name of this Call Bridge entered in step 2.

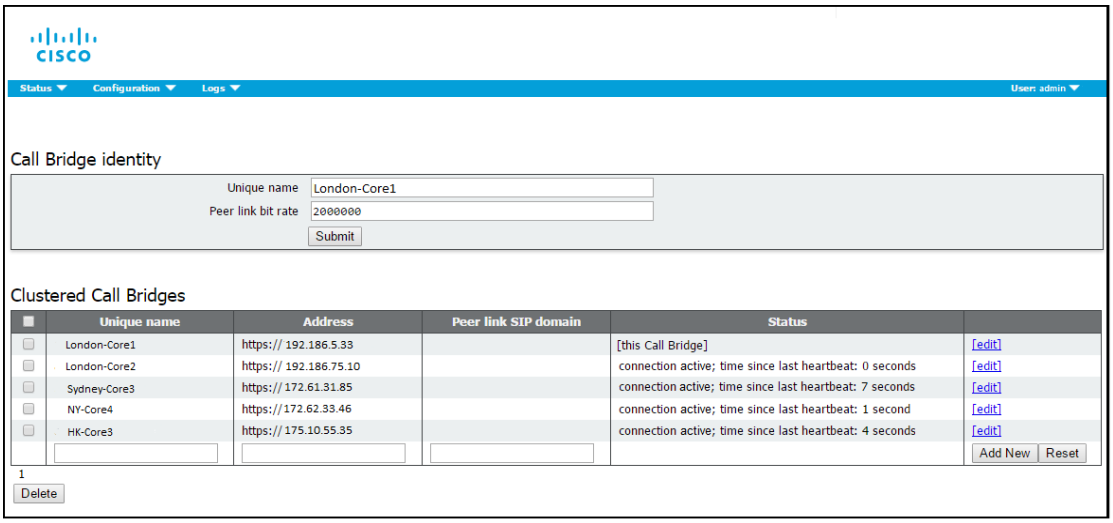
   b. Add the Address by which the Web Admin Interface of the Call Bridge can be reached from other servers in the cluster via HTTPS. This address will be used for management messaging e.g. Participant lists. Note: Web Admin can be set to listen on the Admin interface as well as interfaces A through D.

   c. If you specify the Peer Link SIP Domain then, that is what will be used when calling to a remote server for a Peer call. You can have it route to your call control device if you have set up an Outbound rule that has your call control defined to use as SIP Proxy. Or you can still call direct by putting your Meeting Server FQDN in your outbound calls as SIP Proxy to use to route direct without a call control. If routing through an existing call control, we recommend that you use a unique domain or each servers' FQDN for the Peer link SIP domain. This ensures no accidental call loops which can occur if using the same domain on all.

d. Click **Add**

e. Repeat steps 4a to 4d for each Call Bridge that will be part of the cluster, entering the unique name for each Call Bridge that you set up in step 2.

The table headed Clustered Call Bridges should now have one entry for every Call Bridge that will be part of the cluster, and the Unique Name in the Call Bridge Identity section identifies which Call Bridge this is.



The information in the Clustered Call Bridges table is replicated to every Call Bridge in the cluster. Therefore, you can now go to any Call Bridge server, sign in to the Web Admin Interface and go to **Configuration > Cluster** to see the status of all the clustered Call Bridges.

The Call Bridge cluster is now setup, and the clustered Call Bridges will share the same dial plan (Inbound, Outbound and Call Forwarding dial plan rules). For the next step, you need to configure dial plans for inter-peer calls between the clustered Call Bridges. See *Setting up dial plan rules for Inter-peer calls*.

Note: When adding Call Bridges in the **Configuration > Cluster** page, you need to use the Web Admin's IP address. However, when configuring the dial plan to route calls between Call Bridges (when 'Peer link SIP domain' field is left empty), you need to use the Call Bridge's IP address in both 'domain' and 'sip proxy address' fields of the outbound dial plan. For more information, see the section *Setting up dial plan rules for Inter-peer calls*.

6.3.1 Call Bridge cluster validation

You can improve the security of a Call Bridge cluster by using the Call Bridge trust store to validate Call Bridges within the cluster. As Call Bridges connect to each other over HTTPS, which is fronted by the Web Admin, you need to create a certificate bundle holding the Web Admin certificates of the clustered Call Bridges, and upload the certificate bundle to the trust store of each Call Bridge in the cluster.

Note: From version 3.4, certificate name validation has been implemented when '**`callbridge trust cluster`**' is enabled and thus the peers configured on the clustering must match the exact FQDN on its corresponding Web Admin certificate and failure to make this configuration will result in Call Bridgecluster failure.

For more information, see the Certificate Guidelines for Scalable and Resilient Server Deployments.

6.3.2 Using DTMF sequences in clustered Call Bridge deployments

Note: This feature does not support sending DTMF sequences to Lync participants in dual homed conferences.

DTMF sequences can be configured for participants. This enables DTMF sequences to be sent to any participant in the conference regardless of which Call Bridge they are connected to. Similarly, DTMF can now be sent when calling out from a cluster of Call Bridges using the participants API to call out. This applies to cases where the Call Bridge for the outbound call is either implicitly or explicitly chosen via load balancing, dial plan rules, or selection of Call Bridge Group or Call Bridge.

- To send DTMF key sequences to the far end in clustered and load balanced deployments:

  POST to **`/calls/<call id>/participants`** with the **`dtmfSequence`** parameter containing a string composed of: digits 0 to 9, # and a “,” which adds a pause between digits.

  This will send the DTMF sequence to the far end when a participant is initially created or during the call.

- To set a DTMF sequence to get played to a specific participant already in a call:

  PUT to **`/participants/<participant id>`** the parameter **`dtmfSequence`** .

6.4 Dial Plan Information

Chapter 1 discusses setting up a dial plan for scalable and resilient deployments: inbound dial plan rules, outbound dial plan rules and call forwarding rules. The specific dial plan rules you require and their priority depend on your deployment, not only the topology of the Meeting Server deployment but also of your call control platform(s) and whether you prefer to use local resources or want to load balance calls.

This section discusses some decisions you will need to make when setting up dial plans. The dial plan is stored on the database server, you can amend the dial plan from any Call Bridge in the Call Bridge cluster. You can also see the full dial plan in the Web Admin Interface of any Call Bridge, but we recommend setting up the dial plan using the API because this provides more flexibility. Outbound dial plan rules must be configured via the API for clustered Call Bridges. Appendix 1 provides examples of using the API.

For example, you may want every Call Bridge in a cluster to use the same outbound dial plan, but this will not allow for geographic differences. To account for location and topology you can mix rules that apply to all Call Bridges with those that are specific to one Call Bridge – which you specify in the API. For example, you may want calls placed to +01 numbers to always be made from a Call Bridge in the US.

In the **Configuration > Outbound calls** page of the Web Admin Interface, there is a column called **Call Bridge Scope**. You cannot edit this column, it simply shows you what was set in the API. Specifically it does not show which Call Bridge a rule applies to if the rule is Call Bridge-specific.

Note: If the **Call Bridge Scope** is set to **All** this is equivalent to the API scope setting of " global" , while the **Call Bridge Scope** of **One** is equivalent to a scope of " callbridge" in the API.

When setting the dial plan rules, ensure that:

- Outbound dial plan entries for calls are either valid for all Call Bridges in the cluster i.e. have a **Call Bridge Scope** of **All** in the Web Admin Interface, or that rules with a setting of **One** are completely defined using the API to specify the Call Bridge that the rule applies to.
- The configured Incoming dial plan for SIP calls covers the set of domains that will be routed to any Call Bridge in the cluster from outside the cluster.

6.4.1 Setting up dial plan rules for Inter-peer calls

Inter-peer calls between Call Bridges are placed using the outbound dial plan rules on the initiating Call Bridge.

If you want to make calls directly between Call Bridges, you should leave 'Peer link SIP domain' field blank (on **Configuration > Cluster** page). If you want to make inter-peer calls between Call Bridges via a call control device, you should configure the 'Peer link SIP domain' field with each Call Bridge's FQDN. The outbound dial plan rules will need to be configured depending on which method you have used in your deployment.

- If you have left 'Peer link SIP domain' field blank in the **Configuration > Cluster** page, inter-peer calls will be placed in the format: randomURI@callbridge_ip_address. For example, if Call Bridge A (ip address 10.10.10.10) is placing an inter-peer call to Call Bridge B (ip address 10.10.10.20), the outgoing call from Call Bridge A will have a SIP address in the format: randomURI@10.10.10.20.

  In this case, each Call Bridge needs an outbound dial plan rule to each of its peers, based on the peer's IP address. So the outbound dial plan rule should be configured with the peer Call Bridge's IP address in both 'Domain' and 'Sip proxy to use' fields.
- If you have configured 'Peer link SIP domain' with the peer Call Bridge's FQDN, inter-peer calls will be placed in the format: randomURI@callbridge_FQDN. For example, if Call Bridge A (FQDN: callbridgeA.example.com) is placing an inter-peer call to Call Bridge B (FQDN:

callbridgeB.example.com), the outgoing call from Call Bridge A will have a SIP address in the format: randomURI@callbridgeB.example.com

In this case, each Call Bridge needs an outbound dial plan rule to each of its peers, based on the peer's FQDN. So the outbound dial plan rule should be configured with the peer Callbridge's FQDN in the 'Domain' field, and the call control device's address in the 'SIP proxy to use' field."

Note: Inter-peer calls between Call Bridges will be placed using the outbound dial plan rules on the initiating Call Bridge. Therefore you may need to add outbound dial plan rules if the calls are to IP addresses in order to place the call direct to the other Call Bridge, rather than via a SIP proxy. However, your existing outbound dial plan rules may already cover inter-peer calling if you are using domain dialing.

Note: Inter-peer signaling between Call Bridges uses HTTPS.

Note: Outgoing calls with the same dial plan rule priority will favor a local call control device over one in another geographical location.

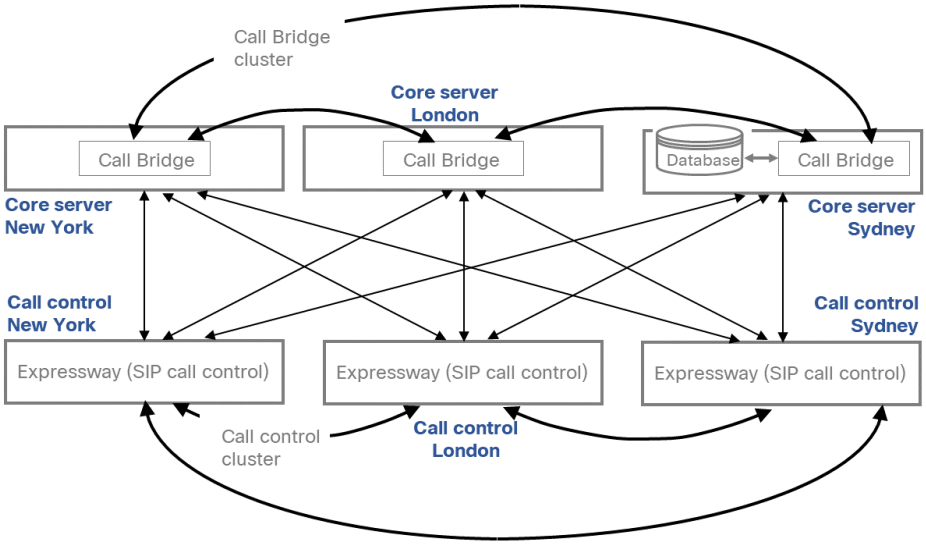
6.4.2 Examples

These examples use a Cisco VCS but the concepts are independent of the Call Control device.

In the simplest case, a single VCS is trunked to a Meeting Server (and therefore a single Call Bridge) to provide integration with SIP endpoints. In a larger deployment of multiple Call Bridges across multiple data centres more thought must be given to the dial plan.

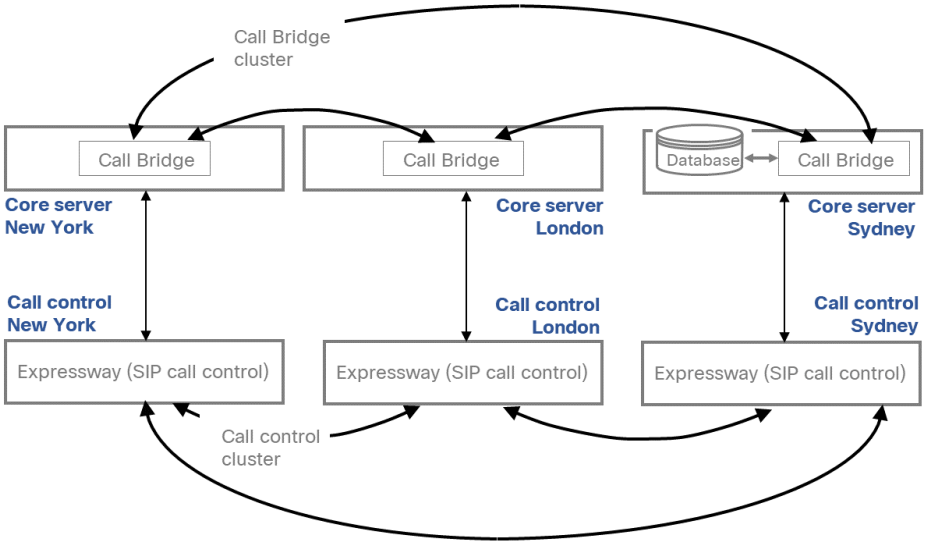
Example #1: One option is to configure a cluster of VCSs to forward calls to the cluster of Call Bridges. This provides resiliency because any VCS can route a call to any Call Bridge, and calls from any Call Bridge can be routed to any VCS (see the figure below).

Figure 18: Mesh configuration for SIP Call Control



Example #2: If you have a geographically distributed call control system then you can choose to selectively trunk these VCSs to Call Bridges that are local to them. For example, VCSs in London can be trunked to London-based Call Bridges and New York-based VCSs can be trunked to New York-based Call Bridges. This ensures that media does not travel between data centers unnecessarily, and if you have a conference that spans locations there is a single peer-link between Call Bridges to minimize bandwidth usage between data centers.

Figure 19: Geographical configuration example for SIP Call Control



6.5 Load Balancing calls across Meeting Servers

Use the Call Bridge Groups feature to load balance incoming and outgoing calls on clustered Meeting Servers, and avoid overloading individual Meeting Servers in the cluster.

With Call Bridge Groups configured, a Meeting Server cluster can intelligently load balance calls across the Call Bridges within the same location or across nodes in different locations. The intelligent decision making behind where calls end up, is handled by the Meeting Servers. The call control system needs to be able to handle SIP messages from the Meeting Servers, in order to move calls to the correct location. This functionality has been tested using Cisco Unified Communications Manager and Cisco Expressway as call control systems. These are the only Cisco supported call control systems for this functionality. For load balancing with Cisco Expressway, use Cisco Expressway release X8.11 or later with Cisco Meeting Server release 2.4 or later.

The Cisco Meeting Server white paper on Load Balancing Calls provides example deployments with Cisco Unified Communications Manager and Cisco Expressway as the call control device.

Note: There are different call capacities for Meeting Servers in a Call Bridge Group compared to a single or cluster of Meeting Servers. provides an overview of the difference in call capacities.

Note: If you are not using load balancing with Call Bridge Groups, then calls will not be rejected, but the quality of all calls will be reduced when the load limit (see Section 6.5.2) is reached. If this happens often, we recommend that you buy additional hardware.

To determine the media loading on a Meeting Server, perform a GET on the the API object **`/system/load`**. A numeric value for parameter **`mediaProcessingLoad`** will be returned, this represents the load on the Meeting Server.

6.5.1 Call Bridge Groups

The load balancing of calls occurs between a group of Call Bridges that exist in the same location. To configure which Call Bridges are in each location, the concept of Call Bridge Groups is used. A Call Bridge Group defines a subset of cluster nodes that are more closely linked and should be treated as equivalent. This could refer to those in a single data center, or those in the same continent. The decision of how to group Call Bridges will depend on the specifics of network configuration and the desired behavior.

For the load balancing feature to work correctly, a Round Trip Time (RTT) of less than 100 ms is required for the servers in a Call Bridge Group. The maximum RTT between any two nodes in the same cluster remains as 300 ms.

When using Cisco Unified Communications Manager, call routing relies on the use of route patterns, route groups and route lists across the Cisco Unified Communications Manager

deployment. It is assumed that these concepts are understood. For information on configuring these features, please consult the Cisco Unified Communications Manager documentation.

When using Cisco Expressway, call routing relies on the use of dial plans and zones, and it is assumed that these concepts are understood. Ensure the Cisco Expressway can route to the domain of the callee in the event of a replaced call. For information on configuring dial plans and zones, please consult the Cisco Expressway documentation.

For examples on load balancing calls, see the Cisco white paper "Load Balancing Calls Across Cisco Meeting Servers".

6.5.2 Configuring Call Bridges for load balancing incoming calls

There are three aspects to setting up the load balancing of calls across a Meeting Server cluster:

- creating the Call Bridge Groups,
- enabling load balancing,
- and optionally, fine-tuning the load balancing on each Call Bridge. In most deployments this will not be necessary.

In addition, load balancing incoming calls involves outbound calls from Call Bridges to Cisco Unified Communications Manager or Cisco Expressway. For these outbound calls to work, outbound dial plan rules must be configured, see Load balancing outbound SIP calls .

Note: If load balancing incoming calls involves outbound calls from Call Bridges to Cisco VCS, instead of Cisco Expressway, then a traversal license is required on the VCS. There is no requirement for a Rich Media Session license on Cisco Expressway for any load balanced Meeting Server deployments.

Note: If you are not using load balancing with Call Bridge Groups, then calls will not be rejected, but the quality of all calls will be reduced when the load limit is reached. If this happens often, we recommend that you buy additional hardware.

Creating Call Bridge groups

1. For each Meeting Server cluster, decide how to group the Call Bridges, for instance by data center or by country or region.
2. Using the Web Admin interface of one of the servers in the cluster, select **Configuration>API**

3. Create a new Call Bridge Group
   a. From the list of API objects, tap the ► after **/api/v1/callBridgeGroups**
   b. Select the **Create new** button, enter the name of the new callBridgeGroup and set the parameters for the Call Bridge Group. Select **Create**.
   c. The new group will appear in the list of callBridgeGroups.
4. Identify the Call Bridges to be grouped
   a. From the list of API objects, tap the ► after **/api/v1/callBridges**
   b. Select each Call Bridge to be added to the group by clicking on the callBridge id
      i. Click on the **Choose** button beside the **callBridgeGroup** field, and select the callBridgeGroup created in step3b
      ii. Click **Modify**
   c. Repeat step 4b for each Call Bridge that needs to be added to the Call Bridge Group.
5. Repeat for all other Call Bridge Groups.

Specifying the load limit on a cluster and enabling load balancing

1. On each Call Bridge in a cluster, specify the load limit for that server
   a. From the list of API objects, tap the ► after **/system/configuration/cluster**
   b. Select the **View or edit** button, and enter a value for **loadLimit**. Click the **Modify** button. This sets a load limit for the maximum load on the server, see Table 6 for load limits.

   Table 6: Load limits for server platforms

System	Load Limit
Meeting Server 2000 M6	1,296,000
Meeting Server 1000 M6	160,000
VM	1250 per vCPU
Cisco Meeting Server 1000 M7	240,000

   Setting a load limit on any Call Bridge means it will reject calls based on the current load. By default, the rejection of calls from new participants occurs at 80% of the load limit to allow for the distribution of calls. This value can be fine tuned, see below.
2. Enable load balancing on each server in the cluster.

   For Cisco Unified Communications Manager deployments:
   a. From the list of API objects, tap the ► after **/callBridgeGroups**
   b. Click on the **object id** of the Call Bridge Group trunked to Cisco Unified Communications

Manager

c. Set **loadBalancingEnabled**=**true**. Click **Modify**

For Cisco Expressway deployments:

a. From the list of API objects, tap the ► after **/callBridgeGroups**

b. Click on the **object id** of the Call Bridge Group trunked to Cisco Expressway

c. Set **loadBalancingEnabled**=**true** and set **loadBalanceIndirectCalls**=**true**. Click **Modify**

Tip: If you have only one Call Bridge, and you want to reject calls rather than reducing quality, then create a Call Bridge Group with a single Call Bridge and enable load balancing.

Fine-tuning the load balancing

It is possible to fine tune the load balancing parameters, but take care as it could impact the availability of the solution. Changing the default values may lead to overloading of servers and a degradation of video quality. This could occur due to either conferences becoming fragmented over multiple Call Bridges, or conferences using too many resources on a single Call Bridge.

Load balancing calls on a Call Bridge is controlled by 3 parameters:

- **loadLimit** – a numeric value for the maximum load on the Call Bridge, as set above.
- **newConferenceLoadLimitBasisPoints** – a numeric value for the basis points (1 in 10,000) of the load limit at which incoming calls to non-active conferences will be disfavored, ranges from 0 to 10000, defaults to 5000 (50% load). Value is scaled relative to **LoadLimit**.
- **existingConferenceLoadLimitBasisPoints** – a numeric value for the basis points of the load limit at which incoming calls to this Call Bridge will be rejected, ranges from 0 to 10000, defaults to 8000 (80% load). Value is scaled relative to **LoadLimit**.

To change the default threshold values for a Call Bridge:

1. From the list of API objects, tap the ► after **/system/configuration/cluster**
2. Select the **View or edit** button, and set values for **newConferenceLoadLimitBasisPoints** and **existingConferenceLoadLimitBasisPoints**. Click **Modify**.

Note: distribution calls are always accepted, and will consume additional resources. If modifying the load balancing parameters, ensure that any necessary overhead for these calls has been included in the calculations.

How load balancing uses the settings

Within each Call Bridge Group there is a particular preference order in which Call Bridges will be chosen for each space. Any call for a space landing anywhere in the Call Bridge Group will be preferentially redirected to Call Bridges based on this order. The redirection is based on two thresholds: the existing conference threshold and the new conference threshold.

The thresholds are defined as:

existing conference threshold = existingConferenceLoadLimitBasisPoints/10000×loadLimit

new conference threshold = newConferenceLoadLimitBasisPoints/10000×loadLimit

When a call lands on a Call Bridge the load limit is checked, if the load limit is above the existing conference threshold, then the call is rejected. Note calls can also be rejected for other reasons. Rejected calls should be redirected by the call control device.

If the load limit is below the existing conference threshold, then the call will be answered and any IVRs traversed. Once the conference is known then the Call Bridge preference order within the group can be determined. This order is used to decide between Call Bridges in cases where there are multiple Call Bridges that could be chosen.

If any Call Bridges within the group are already running the conference, then the load limits of these Call Bridges are checked. If any of these are below the existing conference threshold, then one of these will be used.

If no Call Bridge has yet been chosen, then one of the Call Bridges with a load limit less than the existing conference threshold is chosen.

6.5.3 Load balancing outbound SIP calls

Call Bridge Groups supports the load balancing of outbound SIP calls, in addition to inbound SIP calls.

To load balance outbound SIP calls, do the following:

- enable load balancing of outbound SIP calls from spaces,
- set up outbound dial plan rules for load balancing outbound SIP calls,
- supply the Call Bridge Group or a specific Call Bridge for the outbound SIP calls.

Once load balancing is enabled, outbound SIP calls follow the logic:

- Find the highest priority outbound dial plan rule that matches the domain,
  - if this applies to a local Call Bridge, then balance the call within the local Call Bridge Group.
  - if this only applies to remote Call Bridges, then load balance the call within the Call Bridge Group to which the Call Bridge is a member.

Note: Load balancing of calls from/to Lync clients, is not currently supported by Call Bridge Groups.

How to enable load balancing of outbound SIP calls

To configure the Call Bridges in a specific Call Bridge Group, to attempt to load balance outgoing SIP calls from spaces:

1. From the list of API objects, tap the ▶ after **/callBridgeGroups**
2. Click on the **object id** of the selected Call Bridge Group or **Click new** to create a new Call Bridge Group.
3. Set **loadBalanceOutgoingCalls** = **true**. Click **Modify**.

For load balancing of outbound calls, each Call Bridge in the group must have the same dial plan rules.

How to set up outbound dial plan rules for load balancing outbound SIP calls

There are 3 ways to set up outbound dial plan rules for load balancing outbound SIP calls:

1. Setting the **scope** parameter to **global** in all of the outbound dial plan rules. This ensures that all Call Bridges are able to use all of the outbound dial plan rules to reach a matching domain.
2. Creating identical outbound dial plan rules for each Call Bridge in the Call Bridge Group. Set the **scope** parameter to **callBridge**. Use the **callBridge** parameter to set the **ID** of the Call Bridge.
3. Creating outbound dial plan rules for the specific Call Bridge Group. Set the **scope** parameter to **callBridgeGroup**, and set the **callBridgeGroup** parameter to the **ID** of the Call Bridge Group.

Before using load balancing of outbound calls, review the existing outbound dial plan rules for each Call Bridge in the Call Bridge group:

1. From the list of API objects, tap the ▶ after **/outboundDialPlanRules**
2. Either create a new outbound dial plan rule or click on the **object id** of an existing outbound dial plan that you plan to use for load balancing outbound SIP calls
3. Select the settings for **scope**, **callBridge** and **callBridgeGroup** depending on how you plan to use the dial plan (see the 3 alternative ways above)

How to supply the Call Bridge Group or specific Call Bridge to use for outbound SIP calls to participants

To make a call from a specific Call Bridge Group,

1. From the list of API objects, tap the ▶ after **/calls**
2. Click on the **object id** of the individual call
3. Select **api/v1/calls/<call id>/participants** from the **Related objects** at the top of the page
4. Scroll down the parameters to **callBridgeGroup**, tick the box and click **Choose**. Select the **object id** of the Call Bridge Group to use for this call. Click **Create**.

Handling load balancing of active empty conferences

The load balancing algorithm preferentially places new calls onto a Call Bridge where the conference is already active. An empty conference can be started on a Call Bridge by selecting **/calls** from the API object list and then clicking on **Create new**. By default these empty conferences are treated as active. This means that the first call to the empty conference is preferentially load balanced to this Call Bridge. You can prevent the load balancing preferentially using the empty conferences, by setting the parameter **activeWhenEmpty** to **false** when creating the new call.

6.5.4 Load balancing web app calls

In addition to inbound and outbound SIP calls through Cisco Unified Communications Manager or Cisco Expressway, load balancing using Call Bridge Groups can also be applied to web app participants:

- a web app user joining as a member of the space,
- a web app user joining as a non-member of the space, with and without a passcode
- a guest user joining the space,

Every Call Bridge in a Call Bridge Group must have a connection to the same Web Bridges to ensure load balancing works.

Note: A call control device is not required for load balancing calls in deployments where only web app is used to make calls (no SIP calls).

Enabling load balancing web app participants

To enable load balancing web app participants:

1. From the list of API objects, tap the ► after **/api/v1/callBridgeGroups**
2. Select the object id of the Call Bridge Group
3. Set **loadBalancingEnabled** = **true**.
4. Set **loadBalanceUserCalls** = **true**.
5. Click **Modify**.

Disabling load balancing web app participants

To disable load balancing web app participants while continuing to load balance SIP calls:

1. From the list of API objects, tap the ► after **/api/v1/callBridgeGroups**
2. Select the object id of the Call Bridge Group
3. Set **loadBalanceUserCalls** = **false**. Click **Modify**.

6.6 Lync Account Information

For Call Bridge integration with Lync Edge, we recommend that you configure each Call Bridge with its own login account so that there are no conflicts. For each Lync call to or from that Call Bridge, the Meeting Server requests TURN resources from the Lync Edge using that account. Until that call is disconnected, that resource is considered " Used" from a Lync point of view. Lync will only allow up to 12 TURN allocations per user account; therefore, with 1 registration, only 12 calls are possible.

Note: If you share that one account across multiple Call Bridges, you will only be allowed 12 Lync calls in total across all Call Bridges.

6.7 Choosing Call Bridge mode to connect participants to Lync conferences

You can choose the behavior of the Call Bridge when connecting participants to Lync conferences. A request parameter **`lyncConferenceMode`** has been added when POSTing to **`/callProfiles`** or PUTing to **`/callProfile/<call profile id>`**.

Set **`lyncConferenceMode`** to **`dualHomeCluster`** if you want the calls to be distributed between clustered Call Bridges, with one of the Call Bridges calling out to the AVMCU meeting. This is the same behavior as version 2.2 and earlier.

Set to **`dualHomeCallBridge`**if you do not want the calls to be distributed between clustered Call Bridges, but calls on the same Call Bridge need to be combined into one conference. This will result in a single conference on each Call Bridge, each Call Bridge will call out to the AVMCU meeting.

Set to **`gateway`** if you do not want the calls to be distributed between Call Bridges or calls on the same Call Bridge combined into one conference. Each SIP participant will be in their own conference with an associated call out to the AVMCU meeting.

Note: Set **`lyncConferenceMode`** to **`gateway`** to disable dual home conferencing.

For example, in a deployment with three SIP participants connecting to an AVMCU conference via two Meeting Servers, with two of the SIP participants on the same Meeting Server, the following behaviors will be seen by selecting the different modes:

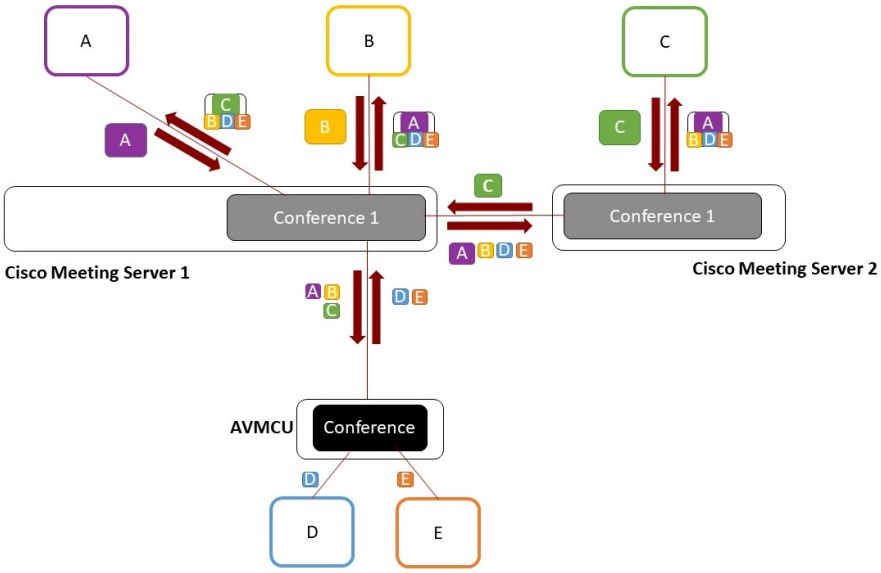
- **`dualHomeCluster`**: media streams are sent between the clustered Meeting Servers, see Figure 20. All calls from the SIP participants will be combined into one conference spanning both Call Bridges; one Call Bridge will call out to the AVMCU. **`dualHomeCluster`** uses one Multiparty Plus license for the single conference.

Note: In the **dualHomeCluster** mode, video streams for participants directly connected to the AVMCU, come from the AVMCU. If using Lync2013 or Skype for Business and four or more participants join the meeting, then the resolution of these streams may limited to a maximum of 360p.

This mode typically allows more video streams to be available, often at high resolution. This comes from two factors: firstly, if fewer media streams are requested from Lync, these streams may be at higher resolution, secondly the streams sourced from SIP devices are typically available at a higher resolution. However, since all audio streams need to be sent, then even without video, this can be a substantial overhead leading to increase bandwidth requirements. Since video streams traverse multiple hops, then even more bandwidth is required. And the multiple hops can add latency.

Note: This mode leads to less predictability, since the order that people join the conference changes the connections made, and hence the available streams. In addition, the first Call Bridge to connect to Lync may not be the best choice, and in some cases can mean that fewer participants are seen.

Figure 20: Lync AVMCU/Meeting Server deployment using dualHomeCluster mode



- **dualHomeCallBridge**: will result in the two SIP participants on the same Call Bridge being combined into one conference, see Figure 21. Streams seen by endpoint C come via the AVMCU, the stream of endpoint A seen by endpoint B does not come via the AVMCU. **dualHomeCallBridge** mode involves multiple conferences on the Meeting Servers and will

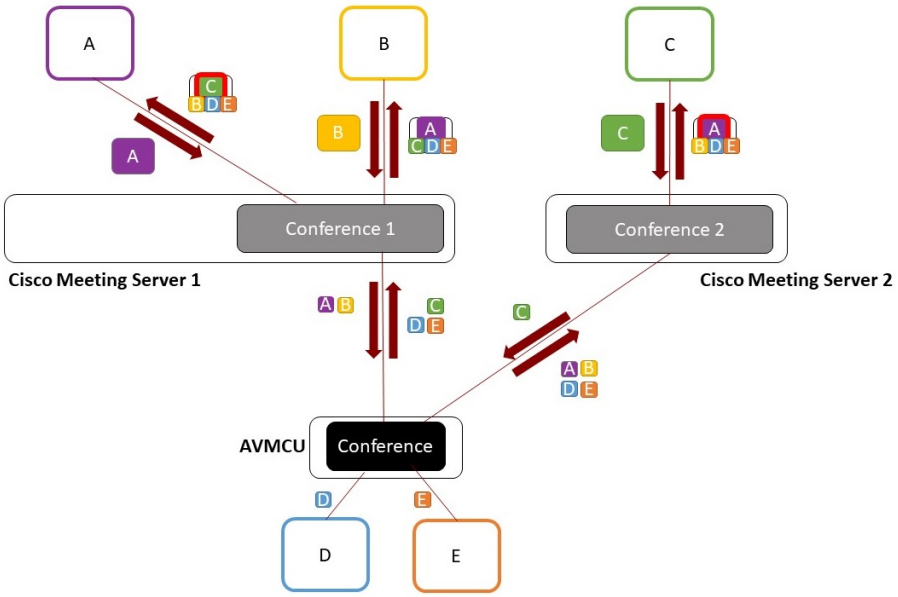
consume multiple Multiparty Plus licenses; two Multiparty Plus licenses are consumed in the example given in Figure 21.

Note: In the dualhomeCallBridge mode, video streams for participants on another Call Bridge and directly connected to the AVMCU, come from the AVMCU. If using Lync2013 or Skype for Business and four or more participants join the meeting, then the resolution of these streams may limited to a maximum of 360p.

This mode cuts down on the bandwidth usage, as media streams going towards the AVMCU do not need to be sent to a single Meeting Server node. However, video coming from the AVMCU can potentially be at lower resolution (indicated in Figure 21 by a red outline around the main panes potentially affected).

Note: This mode is more predictable since the order of people joining the meeting is not relevant.

Figure 21: Lync AVMCU/Meeting Server deployment using dualHomeCallBridge mode



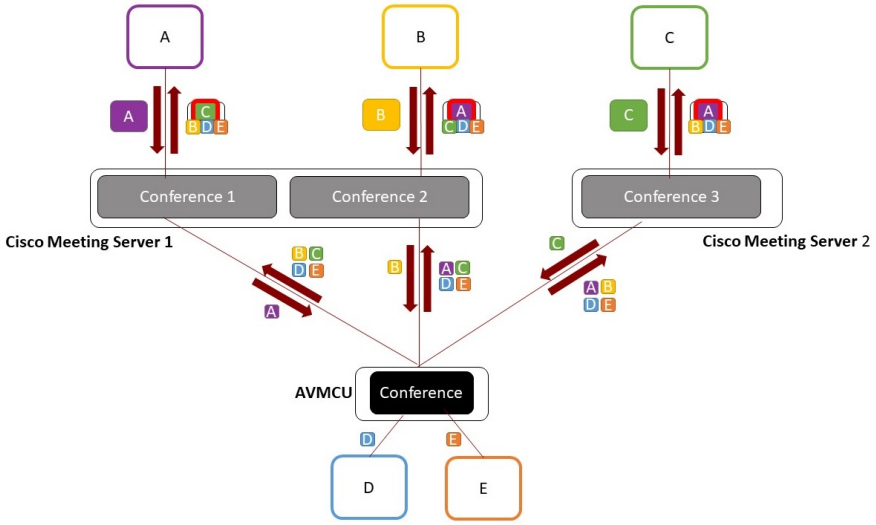
- **`gateway`** this will result in all three Meeting Server conferences calling out to the AVMCU meeting. Video streams seen by endpoints A, B and C all come via the AVMCU, see Figure 22, and can potentially be at lower resolution, indicated by a red outline around the main panes potentially affected.

  Since each call leg is handled separately, then a single Call Bridge may be requesting multiple copies of the same video stream, consuming more bandwidth.

A Shared Multiparty Plus license entitles you to six **gateway** calls. Each participant dialing through CMS to another user, or to a Microsoft Lync AVMCU meeting using the gateway mode consumes one sixth (1/6) of an SMP plus license. In the example given in Figure 22, one half (3/6) of a Shared Multiparty Plus license is consumed. Note that reporting license usage via the API does not reflect this yet—every gateway call will currently report 1 full license consumed rather than the one sixth (1/6) that is actually consumed.

Note: In **gateway** mode, all video streams come from the AVMCU. If using Lync 2013 or Skype for Business and four or more participants join the meeting, then the resolution of each stream may be limited to a maximum of 360p.

Figure 22: Lync AVMCU/Meeting Server deployment using gateway mode



Note: This remains a beta feature.

Prior to version 2.3, video from a maximum of four remote participants could be sent over each distribution link between clustered Call Bridges. From version 2.3, the Meeting Server supports up to nine video streams over the distribution links. Participants using single, dual and three screen endpoint systems can now have a more consistent conference experience whether conferences are hosted on clustered Call Bridges or on a single Call Bridge.

To configure the maximum number of video streams sent over each distribution link between clustered Call Bridges, set the **maxPeerVideoStreams** parameter on API object **/system/configuration/cluster** to a value of 1, 4 or 9; the parameter defaults to 4 if not set.

Note: The API parameter **maxPeerVideoStreams** parameter can take any value between 1 and 9. However, the screen resolution sent is optimized for 1, 4 or 9, so if you set the variable to 2, 3, 5, 6, 7 or 8 then not all of the screen will be used. For example, if set to " 5" then each of the

5 participants will be 1/9th of the screen, similarly if set to " 2" then the two participants will be 1/4 of the screen.

To support more than four video streams across a distribution link, it is recommended that the bandwidth of the link be set to greater than 2Mbps. Use the API or the Web Admin Interface to set the bandwidth. If using the API, PUT a value for the **`peerLinkBitRate`** parameter to the API object **`/system/configuration/cluster`**; the value will be the maximum media bit rate to use on distribution links between Call Bridges in the cluster. Alternatively, using the Web Admin Interface, go to **Configuration > Cluster > Call Bridge identity** and enter the **Peer link bit rate**.

7 Deploying Web Bridge 3

Note: If you are not using the web app, skip this chapter.

Unless otherwise noted, these instructions apply equally to combined or split scalable deployments.

7.1 Deploying multiple Web Bridge 3s

It is possible to deploy multiple Web Bridges in a Meeting Server cluster. You need to ensure all Web Bridges pointed at by a particular A record are associated with the same Call Bridge group

7.2 Setting up the Web Bridge 3 certificates

Each Web Bridge needs a key and certificate pair that is used when web app clients establish TLS connections. If you have not already done so, create, upload and assign certificates to the Web Bridges. More information can be found in Section 4.6, and also the Certificate Guidelines.

Note: If your deployment requires the Cisco Expressway Web Proxy to connect to the Web Bridge, then ensure the SAN field of the Web Bridge certificate includes the A record used by the Expressway-C that will connect to the Web Bridge, otherwise the connection will fail. For example, if the Expressway is configured to connect to the Web Bridge on join.example.com, an A record must exist for this FQDN, and the SAN field of the Web Bridge certificate must include join.example.com.

7.3 Setting up the Web Bridge 3

Use the API to configure the following:

- the FQDN of the URL for the Call Bridge to use to reach the Web Bridge, this should be for a single Web Bridge 3 only. Do not specify the URL of multiple Web Bridges, the Call Bridge needs to be able to open C2W connection to a specific Web Bridge 3 at any time.
- (optional) either the Call Bridge id or the CallBridgeGroup id if they are to be associated with a Web Bridge,
- controls over accessing spaces, including whether the Web Bridge 3 will allow guests to access a spaceby following a web link included in a meeting invite. By default, the secure mode is set, requiring both the call ID and passcode to be entered by the guest before joining the space . Access via a web link is allowed by default without having to supply additional details.

- whether the Web Bridge 3 should accept space and space access method call IDs for the purpose of allowing visitors to join a space. If this parameter is not supplied, it defaults to true allowing visitors to join.
- whether the Web Bridge 3 will accept registered user IDs to resolve to Lync scheduled conference IDs. If this parameter is not supplied, it defaults to false, not resolving IDs to Lync scheduled conference IDs. If the parameter is set to true, web app users can join scheduled Lync conferences by entering the Lync meeting id on the web app sign-in page.
- tenants and tenant groups associated with the Web Bridge 3, if required,
- customization of the background image and logo on the landing page for the web app user, if required

Version 3.0 onwards allows you to configure Web Bridge configuration options in a common place rather than solely on a per Web Bridge basis — you can apply the same settings for all, or a specified group of Web Bridges.

The /webBridgeProfiles API object contains the various Web Bridge configuration options. A newly defined Web Bridge profile can be assigned to the individual webBridge objects, or to the top level (global) profile or tenants.

See the section on Web Bridge and Web Bridge Profile Methods in the API Reference Guide for further details on configuring the Web Bridge 3.

7.3.1 How to create and apply a web bridge profile example

1. To create a webBridgeProfile using the Meeting Server Web Admin interface:
    a. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
    b. From the list of API objects, tap the ▶ after **/api/v1/webBridgeProfiles**
    c. Click **Create new**.
    d. Set the **name** field to the name you wish to call this web bridge profile.
    e. Set the **resourceArchive** field to the address of any customization archive file that the Meeting Server should use for web bridges using this web bridge profile.
    f. Set the **allowPasscodes** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. If this parameter is not supplied, it defaults to **true**.
    g. Set the **allowSecrets** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric ID and secret. If this parameter is not supplied, it defaults to **true**.

h. Set the **userPortalEnabled** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should display the sign-in tab on the index page. If this parameter is not supplied, it defaults to **true**.

i. Set the **allowUnauthenticatedGuests** field to either **true** or **false**. If set to **true**, guest access is allowed from the landing screen on web bridges using this web bridge profile. If set to false, visitor access is only allowed once users have logged into the User Portal. If this parameter is not supplied, it defaults to **true**.

j. Set the **resolveCoSpaceCallIds** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings. If this parameter is not supplied, it defaults to **true**.

k. Set the **resolveCoSpaceUris** field to either **off**, **domainSuggestionDisabled** or **domainSuggestionEnabled**. This field determines whether or not this web bridge should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. When set to **off**, join by URI is disabled; when set to **domainSuggestionDisabled**, join by URI is enabled but the domain of the URI won't be auto-completed or verified on this web bridge; when set to **domainSuggestionEnabled** join by URI is enabled and the domain of the URI can be auto-completed and verified on this web bridge. If this parameter is not supplied, it defaults to **off**.

l. Click **Create**.

2. Once you've created the profile, you can then add addresses — this is the Web Bridge URI used to generate meeting invites and the cross launch URL for the web app.

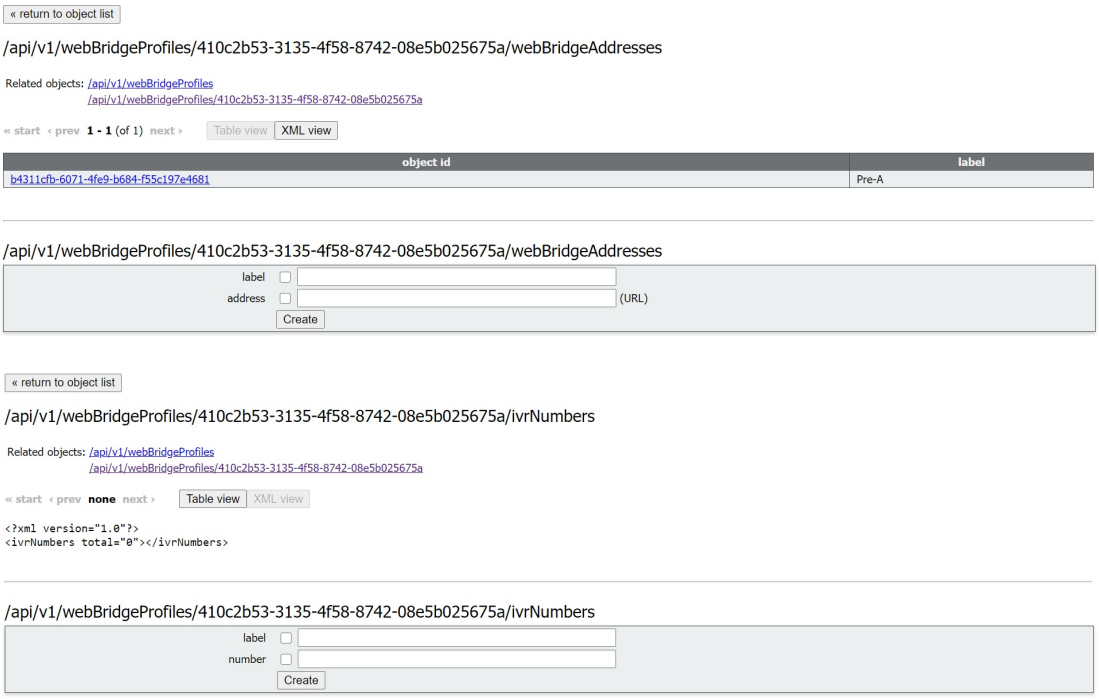
Note: From version 3.1 you can also now specify multiple IVR numbers and Web Bridge addresses — up to 32 IVR numbers and up to 32 Web Bridge addresses per Web Bridge profile. These are used when displaying join information, and for generating email invitations.

In this example a Web Bridge URI and IVR telephone number are applied to a webBridgeProfile as follows:

a. From the list of API objects tap the ► after **/api/v1/webBridgeProfiles**

b. Click **View or edit**

c. From the resulting " webBridgeProfile object selector window" , click **Select** for the **object id** of the **webBridgeProfile** that you have created in Step 1 that you wish to assign a Web Bridge URI and IVR number to. Enter the **label** and URL **address** for the Web Bridge, and enter the **label** and **number** for the IVR as required.



d. Click **Create**.

3. Assign the ID of the newly created webBridgeProfile to any or all of the following, as required:

    - Top level (global) profile (**/api/v1/system/profiles**)
    - Tenants (**/api/v1/tenants/<*id*>**)
    - WebBridges (**/api/v1/webBridges/<*id*>**)

    In this example an updated webBridgeProfile is assigned to the top level (global) profile as follows:

    a. From the list of API objects tap the ▶ after **/api/v1/system/profiles**

    b. Click **View or edit**

    c. Scroll down the parameters to **webBridgeProfile** and click **Choose**.

    d. From the resulting " webBridgeProfile object selector window" , click **Select** for the **object id** of the **webBridgeProfile** that you have created in Step 1 that you wish to assign to the top level global profile.

    e. Click **Modify**.

    f. The newly assigned webBridgeProfile object id should now be listed under **Object configuration**.

7.4 Web Bridge 3 call flow

This section describes the call flow between the web app and components in the Meeting Server.

1. The web browser opens an HTTPS connection.
2. User is prompted to **Join meeting** (see step 3) or **Sign in** (see step 4)
3. If **Join meeting** is selected, user is prompted to enter the Call ID/URI and passcode and set their name.
    a. Call details are sent over HTTPS to Web bridge 3; Web Bridge 3 queries the Call Bridge over the C2W connection to validate call details.
    b. If successful, the user is asked to pick media settings.
    c. After choosing media settings, the call details and desired name are sent over HTTPS to the Web Bridge 3; forwarded over C2W to the Call Bridge. Call Bridge will respond with a call access token which is returned to the browser and details theTURN servers to be used by the browser.
    d. Call Bridge requests allocations from its configured TURN server.
    e. Web app requests allocations from the provided TURN server.
    f. The browser opens a websocket connection to the Web Bridge 3, which is forwarded over C2W connection to the Call Bridge. The call access token is sent over this websocket.
    g. The browser and Call Bridge exchange SDP over websocket containing local media IP address/ports as well as media relay address/ports.
    h. ICE negotiation sends UDP packets between all browser media IP address/port combinations and all Call Bridge address/port combinations; attempts TCP connections to TCP media relay address/ports.
    i. Shortest successful media path is used for transmitting media between the browser and Call Bridge, either directly, through a TURN UDP relay, or through a TURN TCP relay (with the TURN server translating media packets between TCP stream and UDP)
4. If **Sign in** is selected, user is prompted to enter Username and Password.
    a. Sent over HTTPS to web bridge, which is forwarded to call bridge to obtain an portal access token if successful.
    b. Enters user portal, all requests are made over HTTPS sending portal access token as header.
    c. If a join call request is made, the flow is the same as above from step 3c onwards, except instead of sending call details and desired name to obtain a call access token, the browser instead sends call details and portal access token.

Useful information: call access tokens and portal access tokens are different, although similar. The portal access token is valid for 24 hours and allows a user to access the user portal. The call access token is only valid for the duration of a user's participation in the call, and is used only to join a call. The only way to obtain a portal access token is by signing in with a user name and password. A call access token can be obtained either by doing a guest join, or by using the portal access token along with the details of the meeting the user wants to join

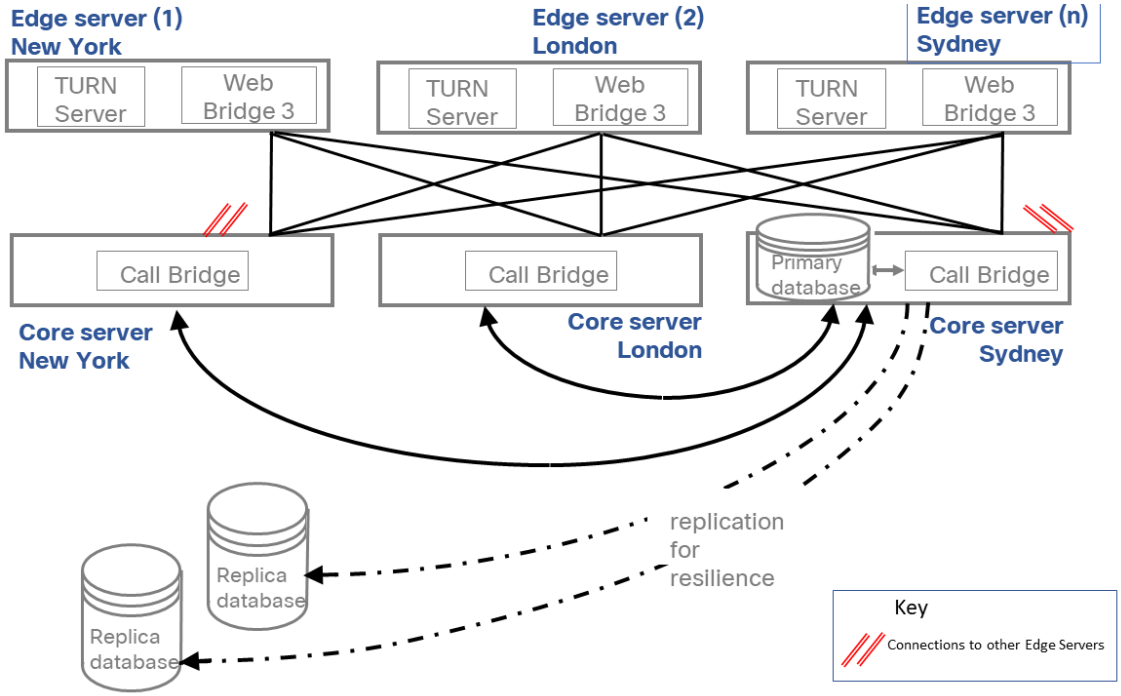
7.5 Web app information

When a web app user calls in:

1. The user's web browser performs a simple DNS A record lookup to choose a Web Bridge 3.
2. The Web Bridge 3 supplies the address of the TURN server for the web app to use.

The Web Bridge 3 and TURN server used for a call do not have to be co-located.

Figure 23: Web Bridges configured in a split deployment



7.6 Enabling HTTP redirect and Web Bridge 3

Web Bridge 3 supports HTTPS. It will forward HTTP to HTTPS if configured to use "httpredirect". This is explained in Section 4.6.

8 Deploying the TURN Servers

The TURN server listens on port 3478 for UDP connections from the Call Bridge, and is also available for remote connections.

The TURN server can also listen on a second port for TCP from client connections, typically 443 (requires 'turn tls <port>' configuration).

Although the Meeting Server configuration option for enabling TURN TCP is named " tls" , TURN TLS is not used by Meeting Server or web app. Web app uses TCP or UDP and Call Bridge always uses UDP (Media is encrypted using SRTP).

Ensure your firewall rules permit UDP port 3478 from the Call Bridge to the TURN server.

8.1 Configuring TURN servers

Unless otherwise noted, these instructions apply equally to combined or split deployments.

Note: While you can still configure a single TURN server via the Web Admin Interface, we strongly suggest that if you have multiple TURN servers you use only the API to configure them, as described in *Setting up Turn Servers on a Call Bridge*.

---

1. Configure and enable each TURN server using the MMP, see Section 4.
2. Set up either a /turnServer/<turn server id> node for each TURN server on the Call Bridge, or have one node with the DNS record pointing to multiple instances.

   For example, using the Meeting Server Web Admin interface:

   a. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:

   b. From the list of API objects, tap the ▶ after **/api/v1/turnServers**

   c. To configure or modify a Call Bridge's TURN server, select **Create new** or the object id of the required existing TURN server and fill in the following parameter values replaced by your values:

   `serverAddress` = **`edge1.example.com`**

   `clientAddress` = **`edge1.example.com`**

   `username` = **`fred`**

   `password` = **`password`**

   `type` = cms
3. If during the MMP configuration you set a non-standard port for TCP on the TURN Server, use the API parameter **`tcpPortNumberOverride`** on object /turnServers/<turn Server id> to configure this value on the Call Bridge.

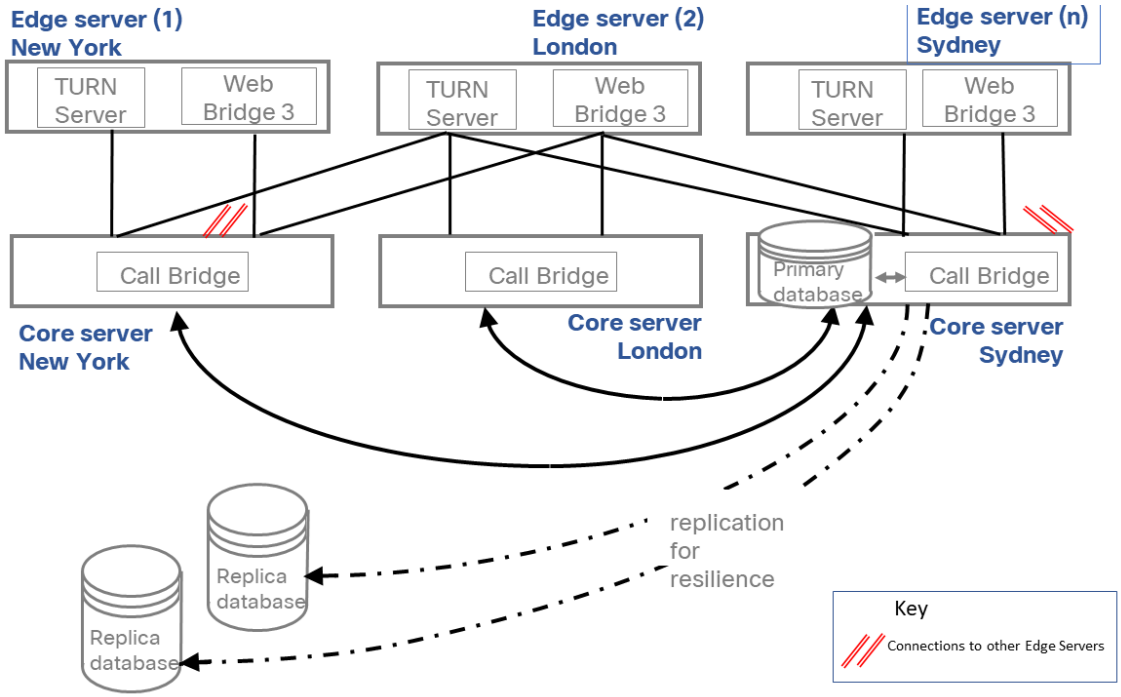
For example, for the TURN server which will interwork the media, POST to the Call Bridge's /turnServers node the following parameter values replaced by your values:

tcpPortNumberOverride = **447**

Note: This parameter is not required for configured Lync Edge servers, where the TCP port number can always be determined automatically.

See Figure 24 for the completed deployment.

Figure 24: TURN servers configured in a split deployment



9 web app in distributed deployments

Spaces can be created via API calls to any Call Bridge in the cluster, and these spaces are visible to all Call Bridges connected to this cluster. An example using the Meeting Server API through the Web Admin interface is provided in J.5.

Figure 25 and Figure 26 each show two web app users'calls to different spaces in a combined and a split deployment. The figures demonstrates that:

- All Call Bridges read from the same database instance irrespective of location. The database holds information on the spaces, for example the members of each space.
- The control and media routes (and therefore the component instances used) vary depending on the client location. This can involve more than one point of presence – as in the call from the web app user in London using a Call Bridge and Web Bridge in a different location.

Figure 25: Two web app users making calls to different spaces in a combined deployment

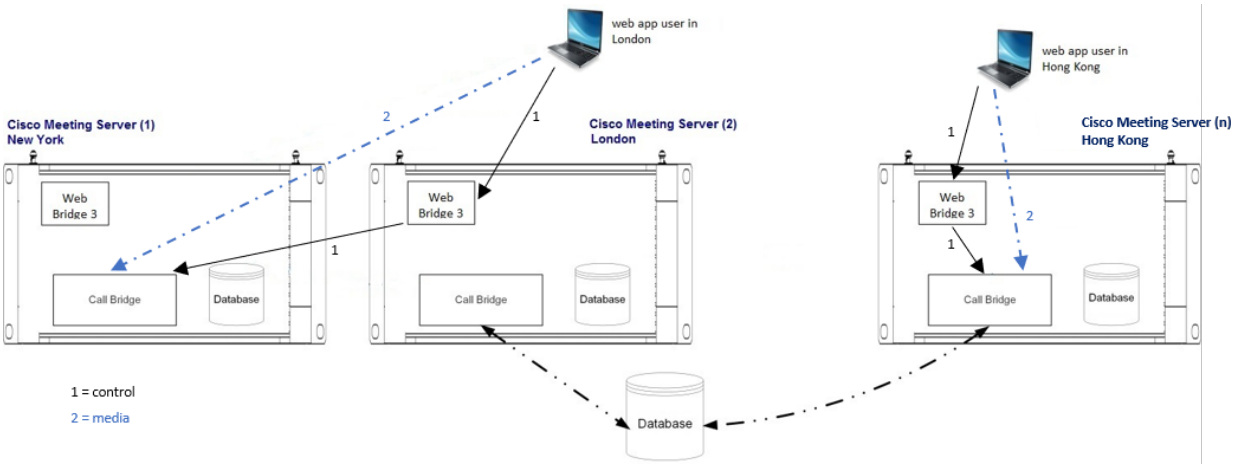
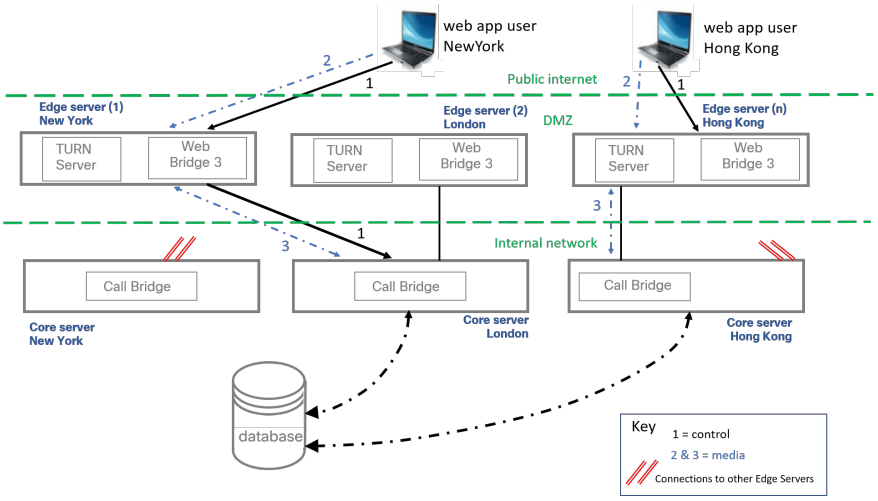


Figure 26: Two web app users making calls to different spaces in a split deployment



However, when a meeting is hosted on more than one Call Bridge (a distributed meeting), these Call Bridges exchange the necessary audio and video streams so that each is aware of every participant, see Figure 27 and Figure 28. Every participant joining a meeting in the space has the same joining experience regardless of where they are located or which components are used, just as if the space was hosted on a single Call Bridge. The in-call experience, including the layout that each participant sees, will depend on the settings for their call leg (set via the API) and who is speaking or the previous speaker, this is the same in-call experience as that for a meeting hosted on one Call Bridge.

Participants see continuous presence for all participants in the same meeting who are connected to the same Call Bridge as themselves, and up to four participants on each of the distributed links to the other Call Bridges.

Example 1: Participant 1 is on Call Bridge A along with participants 2, 3 and 4. Participants 5, 6, 7 and 8 are on Call Bridge B and participants 9, 10, 11 and 12 are on Call Bridge C. If participant 1 selects the layout to be “all equal” they will see the other 11 participants (numbers 2 through 12).

Example 2: Participant 1 is on Call Bridge A along with participants 2 and 3. Participants 4, 5, 6 and 7 are on Call Bridge B and participants 8, 9, 10, 11 and 12 are on Call Bridge C. If participant 1 selects the layout to be “all equal” they will see participants 2 to 7 from Call Bridges A and B along with the four most recent speakers from participants 8, 9, 10, 11 and 12 from Call Bridge C.

Whenever a Call Bridge receives a call from an endpoint, it queries the other Call Bridges to see whether there is already an instantiation of the space. If there is an instantiation, a link is established between the Call Bridge receiving the call and the Call Bridge with the instantiation, this is called a " distribution link" . (However, the Call Bridge may also redirect the endpoint to the other Call Bridge if it's within the same Call Bridge group.) In the case of a distribution link, if more endpoints join the call on either Call Bridge, the link starts to send active participant video streams between the two Call Bridges. When the last endpoint leaves on either side of the

distributed call, the link from that Call Bridge to the other one is torn down, and the call is no longer distributed.

Figure 27: Two Call Bridges calling space A in a combined deployment

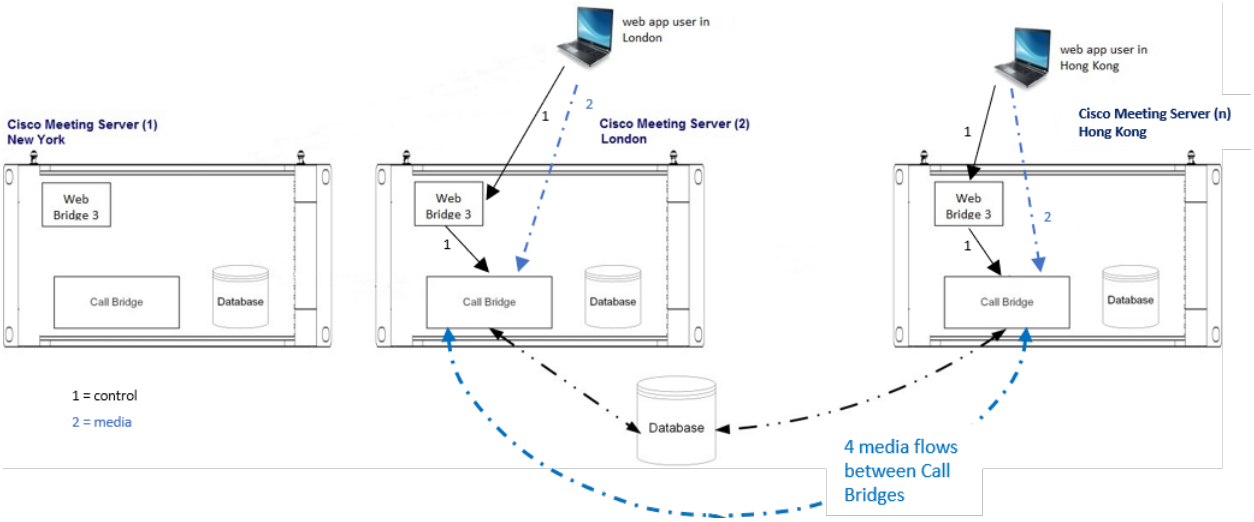
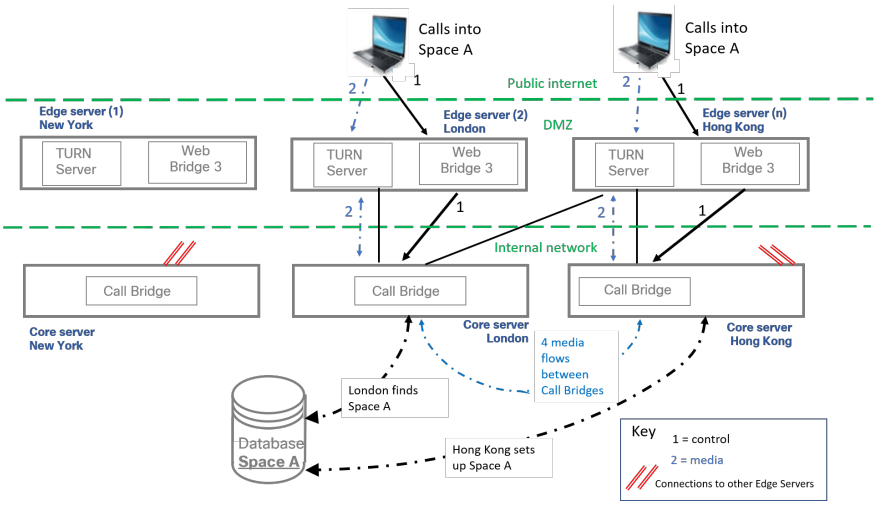


Figure 28: Two Call Bridges calling space A in a split deployment



Example views and media flows are shown in the two figures below with four participants in a space.

Figure 29: Example media flows and views when PC Clients dial into space A in a combined deployment

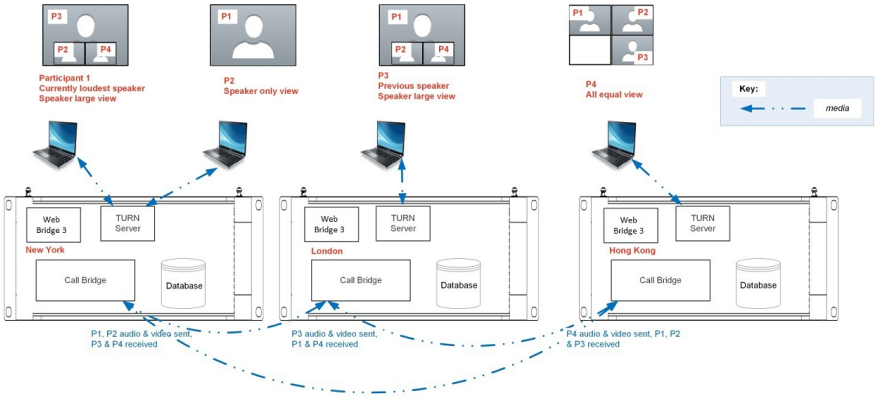
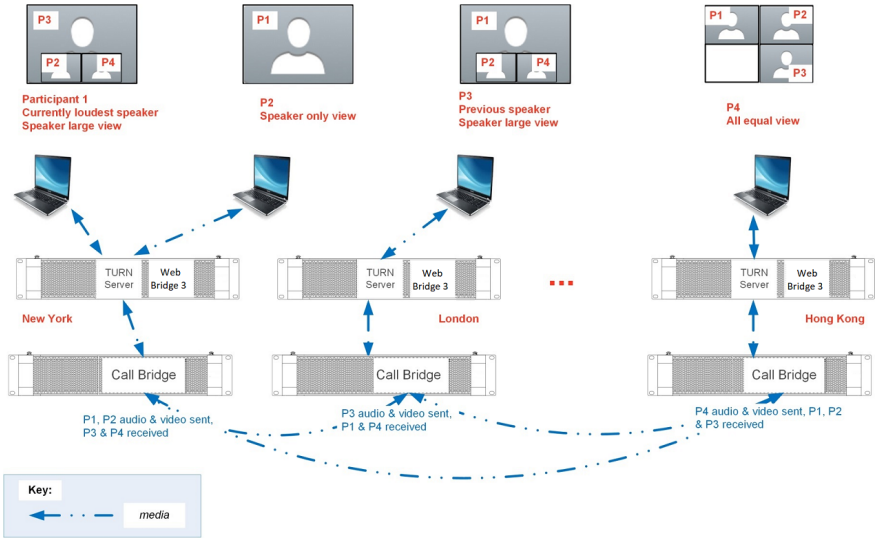


Figure 30: Example media flows and views when PC Clients dial into space A in a split deployment



10 Dial plan configuration – overview

10.1 Introduction

For the Meeting Server to be integrated in a SIP, Lync and voice environment, connections need to be set up from the SIP Call Control, Lync FE Server and Voice Call Control to the Meeting Server. Changes to the call routing configuration on these devices is required in order to correctly route the calls that require the Meeting Server.

Figure 31 assumes a company deployment which has a mix of SIP video endpoints, Lync clients and IP phones: the Meeting Server enables connectivity between Lync clients and SIP video endpoints, and between Lync clients and IP phones.

The SIP video endpoints are configured on a domain called vc.example.com and the Lync clients on example.com. You will need to adapt the example, as appropriate.

Figure 31: Example Meeting Server deployment for dial plan configuration (with Cisco Expressway at the edge)

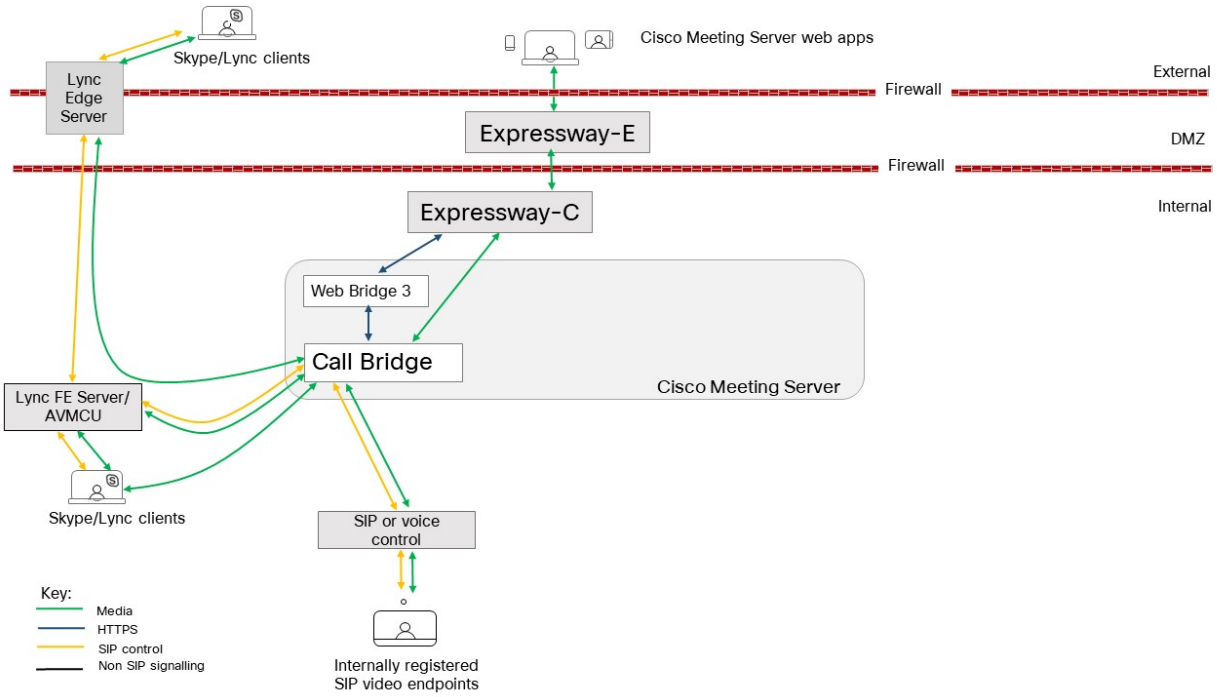
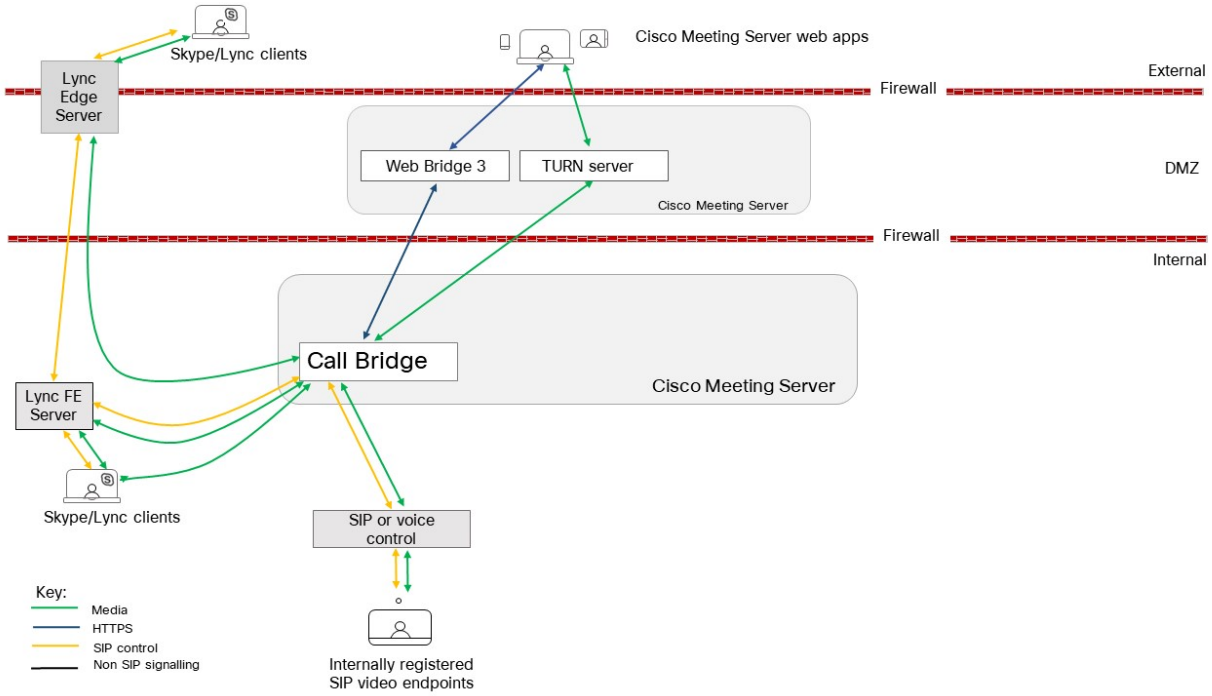


Figure 32: Example Meeting Server deployment for dial plan configuration (using the TURN server component in a split server deployment)



As shown in the figure above, the Lync FE server needs a trusted SIP Trunk to the Meeting Server, configured to route calls originating from Lync clients, through to Meeting Server space, Cisco Meeting Server web app users and also SIP video endpoints. The subdomains vc.example.com (for SIP video endpoints) and meetingserver.example.com (for spaces and Cisco Meeting Server web apps) should be routed through this trunk from the Lync FE server to the Meeting Server.

Note: Connections to Office 365 or on-premise Lync deployments in another organization, should route via a Cisco Expressway. See the Expressway deployment guides for more information.

The SIP Call Control platform needs a SIP trunk set up to route calls to the example.com domain (for Lync Clients) and meetingserver.example.com (for spaces and web apps) to the Meeting Server.

The Meeting Server requires a dial plan to route calls with domain example.com to the Lync FE server and subdomain vc.example.com to the SIP Call Control platform.

The next section discusses the Meeting Server API objects that determine how the Meeting Server handles incoming calls and outbound calls.

Following this chapter, Chapter 11 and Chapter 12 provide step-by-step instructions on configuring the total solution.

10.2 Dial plan rules for incoming calls and outbound calls.

This section explains how to use the Meeting ServerAPI to configure the rules that determine how to handle incoming and outbound calls.

Note: The Single Combined and Single Split Deployment Guides explain how to use the Web Admin interface to configure the dial plan rules. But scalable and resilient deployments generally need more functionality which is only provided through the API.

The following API objects affect dial plan rules on the Meeting Server:

- **`/outboundDialPlanRules`** controls how outbound calls are handled.
- **`/dialTransforms`** can be applied to Outbound calls to control the routing of outbound calls, see Dial Transforms.
- **`/inboundDialPlanRules`** determines how inbound calls are handled. If the "domain" part of the destination URI of an incoming SIP call matches an inbound dial plan rule, then the call is handled by that rule. If it doesn't match an existing inbound dial plan rule, then the call is handled by the call forwarding dial plan rules. Any call routed to the Meeting Server on any domain can be tested for a match for IVRs, web app users or for preconfigured spaces on that server.
- **`/forwardingDialPlanRules`** contains information on how to forward calls that do not match any inbound dial plan rule, or for resolving to a Lync conference.

Note: An incoming call is terminated if does not match any of the inbound dial plan rules or any of the domain matching patterns in the call forwarding dial plan rules

---

10.2.1 /outboundDialPlanRules

To create a new outbound dial plan rule, POST to API object **`/outboundDialPlanRules`** with the following parameters set:

Parameters	Type/Value	Description/Notes
domain *	String	The domain to match in order to apply the dial plan rule; either a complete value (e.g. " example.com") or a "wildcarded" one (e.g. " *.com")
priority	Number	A numeric value which determines the order in which dial plan rules (including rules with wildcarded domains) will be applied. Rules with higher priority values are applied first. If a rule is matched, but the call cannot be made, then other lower priority rules may be tried depending on the failureAction parameter for the rule.

Parameters	Type/Value	Description/Notes
localContactDomain	String	Used when forming an explicit contact domain to be used: if you leave this field blank then the localContactDomain is derived from the local IP address. If you are using Lync, we suggest that you set localContactDomain. If you are not using Lync, we recommend that localContactDomain is not set to avoid unexpected issues with the SIP call flow.
localFromDomain	String	Used when forming the calling party for outgoing calls using this dial plan rule
sipProxy	String	The address (IP address or hostname) of the proxy device through which to make the call. If not set, it is a direct call.
trunkType	**sip**| lync| avaya	Used to set up rules to route calls out to third party SIP control devices such as CiscoExpressway, Avaya Manager or Lync servers. If set to lync or avaya then outgoing calls that use this rule will be made as Lync or Avaya calls with some specialized behavior. sip means that calls using this rule will be standard SIP calls. A common use of the Meeting Server is with an Avaya PBX; these calls will be audio-only. However, the Meeting Server does not impose this restriction on interoperability with Avaya products (some of which support video also): therefore a call of type of 'avaya' does not imply that the call is audio-only.
failureAction	**stop**| continue	Whether or not to try the next outbound dial plan rule if the current one did not result in a connected call. If a rule has a failureAction of stop, then no further rules are used.
sipControlEncryption	**auto**| encrypted| unencrypted	Whether to enforce use of encrypted control traffic on calls made via this rule: • encrypted: allow only encrypted SIP control traffic (TLS connections) • unencrypted: use only unencrypted traffic (TCP or UDP) • auto: attempt to use encrypted control connections first, but allow fall back to unencrypted control traffic in the event of failure. Note: Ensure all " Lync" outbound dialing rules are explicitly set to **Encrypted** mode to prevent the Call Bridge attempting to use unencrypted TCP for these connections in the event of the TLS connection attempt failing.

Parameters	Type/Value	Description/Notes
scope	**global**| callBridge| callBridgeGroup	The entities for which this outbound dial plan rule is valid: • global – all Call Bridges are able to use this outbound dial plan rule to reach a matching domain. • callBridge – this outbound dial plan rule is only valid for a single nominated Call Bridge – whose ID is given in callBridge parameter. • callBridgeGroup – this outbound dial plan rule is only valid for a single nominated Call Bridge Group – whose ID is given in the callBridgeGroup parameter. (From version 2.2). If this parameter is not supplied in a create (POST) operation it defaults to "global".
callBridge	ID	If the rule has a scope of callBridge (see above), this is the id of the Call Bridge for which the rule is valid
callBridgeGroup	ID	If the rule has a scope of callBridgeGroup (see above), this is the id of the Call Bridge Group for which the rule is valid (from version 2.2).
tenant	ID	If a tenant is specified, this rule will only be used to make outbound call legs from calls associated with that tenant; otherwise, this rule may be used from any call.
callRouting (beta feature)	**default**| traversal	This is the media routing that should be used for SIP calls originating from this rule: • default – calls using this rule will use normal, direct, media routing • traversal – media for calls using this rule will flow via a TURN server if this parameter is not supplied in a create (POST) operation, it defaults to "default" .

Note: In a deployment with a Call Bridge cluster, selection of a Call Bridge is done based on the highest priority rule that matches. In the event that multiple matching rules have the same priority, then a rule with a **`scope`** that covers the local Call Bridge will be used preferentially. Once a Call Bridge has been selected, only rules that apply for that Call Bridge will be used. If there are other matches with a scope not including the selected Call Bridge, they will not be used.

To modify an existing outbound dial plan rule, PUT to API object **`/outboundDialPlanRules/<outbound dial plan rule ID>`** with the parameters you want to change.

To retrieve the settings of the outbound dial plan rules already set, use GET on API object **/outboundDialPlanRules**. For more information on using the API, refer to the API Reference Guide.

10.2.2 /inboundDialPlanRules

We recommend that you create rules for every domain expected for incoming calls. With some call control solutions, the domain may be the IP address or hostname of the server. In these cases the highest priority domain is expected to be the main domain, with IP address and hostname rules having lower priority.

You can choose to route calls to users or spaces on a per domain basis. For example, if the incoming call was to **name.space@meetingserver.example.com** and there was a configured space called **name.space** the call would be routed to the space with that name. If the incoming call was to **firstname.lastname@meetingserver.example.com** the call would be routed to that user with that first and last name.

Alternatively, you can choose not to route calls to users or spaces on a per domain basis, and instead use one incoming domain for spaces and another for users.

Points to note:

- Matching for a space and/or users is only done on the part of the URI before the @.
- Rules with a higher priority value are matched first, the highest priority rule that matches a space is used to form the URI in the invitation text. It is expected that the highest priority rules are for the deployment as a whole rather than for individual IP addesses or hostnames. In cases where multiple rules have the same priority then matching occurs based on alphabetical order of the domain.
- Do not leave the **domain** parameter blank, otherwise the Call Bridge will refuse the call.
- if there are no inbound dial plan rules configured then all domains will be matched.
- after a rule is executed, rules further down the list are ignored for the call.

To create a new inbound dial plan rule, POST to API object **/inboundDialPlanRules** with the following parameters set:

Parameters	Type/Value	Description/Notes
domain *	String	The domain to match in order to apply the dial plan rule. Must be a complete value (e.g. " example.com")
priority	numeric	inbound dial plan rules' configured domain values are always exactly matched against incoming calls. For the purposes of generating full URIs to advertise for incoming calls (especially cases where multiple rules are applicable) you can also set a numeric priority value - higher values will be preferred

Parameters	Type/Value	Description/Notes
resolveTocoSpaces	**true**|false	If set to true, calls to this domain will be matched against coSpace URIs (if a match is then found, the incoming call leg becomes a participant in the coSpace).
resolveTolvrs	**true**|false	If set to true, calls to this domain will be matched against configured IVR URIs (if a match is then found, the incoming call leg connects to that IVR).
resolveToLyncConferences	true|**false**	If set to true, calls to this domain will be resolved to a Lync conference URL; if the resolution is successful, the incoming call leg becomes a participant in the Lync conference. If this parameter is not supplied in a create (POST) operation, it defaults to "false".
resolveToLyncSimplejoin	true|**false**	If set to true, calls to this domain will be resolved by an HTTPS lookup to the given URL. If the resolution is successful, the incoming call leg becomes a participant in the Lync conference. If this parameter is not supplied in a create (POST) operation, it defaults to "false". (From version 2.2).
tenant	ID	If specified, calls to this inbound domain will only be matched against coSpace URIs for the specified tenant

To modify an existing outbound dial plan rule, PUT to API object **`/inboundDialPlanRules/<inbound dial plan rule ID>`** with the parameters you want to change.

To retrieve the settings of the outbound dial plan rules already set, use GET on API object **`/inboundDialPlanRules`**. For more information on using the API, refer to the API Reference Guide.

10.2.3 /forwardingDialPlanRules

If an incoming call fails to match any of the inbound dial plan rules, the call will be handled by the forwarding dial plan rules. Forwarding dial plan rules can overlap, and can include wildcards. Order the rules using the **`priority`** parameter value; higher numbered rules are tried first.

Use the **`action`** parameter to decide whether to forward the call or not. It might be appropriate to "catch" certain calls and reject them. You can have rules to decide whether to reject the call outright or to forward the call in "bridge" mode (point-to-point call).

For calls that will be forwarded, you can rewrite the destination domain using the **`destinationDomain`** parameter. A new call is created to the specified domain.

To create a new forwarding dial plan rule, POST to API object **`/forwardingDialPlanRules`** with the following parameters set:

Parameters	Type/Value	Description/Notes
matchPattern	String	The domain to match in order to apply the dial plan rule. Must be a complete domain name (e.g. " example.com") or a “wildcarded” one (e.g. exa*.com). Wildcards are permitted in any part of a domain matching pattern, but do not use “matchPattern=*” as a match all, otherwise you will create call loops.
destinationDomain	String	Calls that are forwarded with this rule will have their destination domain rewritten to be this value
action	**forward** |reject	If set to " forward" causes matching call legs to become point-to-point calls with a new destination. " reject" causes the incoming call leg to be rejected
callerIdMode	**regenerate**| preserve	When forwarding an incoming call to a new destination address, whether to preserve the original calling party's ID or to generate a new one. If this parameter is not supplied in a create (POST) operation, it defaults to " regenerate"
priority	Number	Numeric value used when determining the order in which to apply forwarding dial plan rules; higher values will be applied first
tenant	ID	If a tenant is specified, calls using this rule will be associated with the specified tenant.
uriParameters	**discard**| forward	When forwarding an incoming call to a new destination address, this parameter determines whether to discard any additional parameters that are present in the destination URI of the incoming call, or to forward them on to the destination URI of the outbound call. If this parameter is not supplied in a create (POST) operation, it defaults to " discard”. This parameter is present from version 2.0 onwards

To modify an existing forwarding dial plan rule, PUT to API object **`/forwardingDialPlanRules/<outbound dial plan rule ID>`** with the parameters you want to change.

To retrieve the settings of the forwarding dial plan rules already set, use GET on API object **`forwardingDialPlanRules`**. For more information on using the API, refer to the API Reference Guide.

Note: We do not recommend adding a forwarding dial plan rule that matches all, for example **`matchPattern=*`** as this will result in call loops.

10.3 Dial Transforms

Dial Transforms are applied to outgoing calls prior to the Outbound rules taking effect. When dial transforms are applied, the outbound dial plan rules are applied to the transformed number. Dial Transforms only affect Outbound calls, they do NOT affect gateway calls.

There are three stages to the transform:

- A “type” is applied, which defines the type of preprocessing to apply to the transform.
  - Raw: produces one component - $1
  - Strip: removes dots, dashes, spaces and produces one component - $1
  - Phone: use to transform to an international phone number - produces two components $1county code and $2number

  Note: A phone URI is recognized as a purely numeric string (optionally prefixed by a ‘+’) when it begins with a valid international dial code (e.g. 44 for UK or 1 for US) followed by the correct number of digits for a phone number for that region.

- The components are matched using regular expressions to see if the rule is valid
- An output string is created from the components according to the defined transform

Examples

Example	Type	Match	Transform
For US numbers, use 'vcs1' directly	Phone	($1/01/)	$2@vcs1
For UK numbers, add a prefix and use 'vcs2'	Phone	($1/44/)	90044$2@vcs2
For UK numbers starting with a 7, add '90044' as a prefix, add '123@mobilevcs' as a suffix	Phone	($1/44/)($2/^7/)	90044$2{}123@mobilevcs
For unrecognized all-digit strings, use '@vcs3' as a suffix	Strip	($1/(\d){6,}/)	$1@vcs3
Replace + with 00	Strip	($1/\+(\d)+/)	$1{/\+/00/}
Replace an alphanumeric regex e.g. (.*)@example.com and replace with \1.endpoint@vc.example.com	Raw	($1/(.*) @example.com/)	$1{/@example.com$/ .endpoint@vc.example.com/}

For a single Meeting Server, use the Configuration > Outbound Calls page in the Web Admin Interface to control how dialed numbers are transformed. If a match expression is provided, the regular expression determines whether the specified transform expression is applied

For example, the dial plan in the screen shot below ensures that outbound " +1" (US) calls use one Call Bridge and +44 (UK) calls use another.

However, if you are deploying Call Bridge clustering you need to use the API object **`/dialTransforms`**, because the shared coSpace database is a single configuration location for all Call Bridges in the cluster. In a cluster you do not need to configure the dial transforms

separately on each Call Bridge. The dial transforms for the cluster are those defined on the Call Bridge host server that is co-located with the first coSpace database in the database cluster.

Note: Although the same dial transforms are applied to all Call Bridge in the cluster, the outbound dial plan rules can be configured per-Call Bridge.

---

11 Dial plan configuration – SIP endpoints

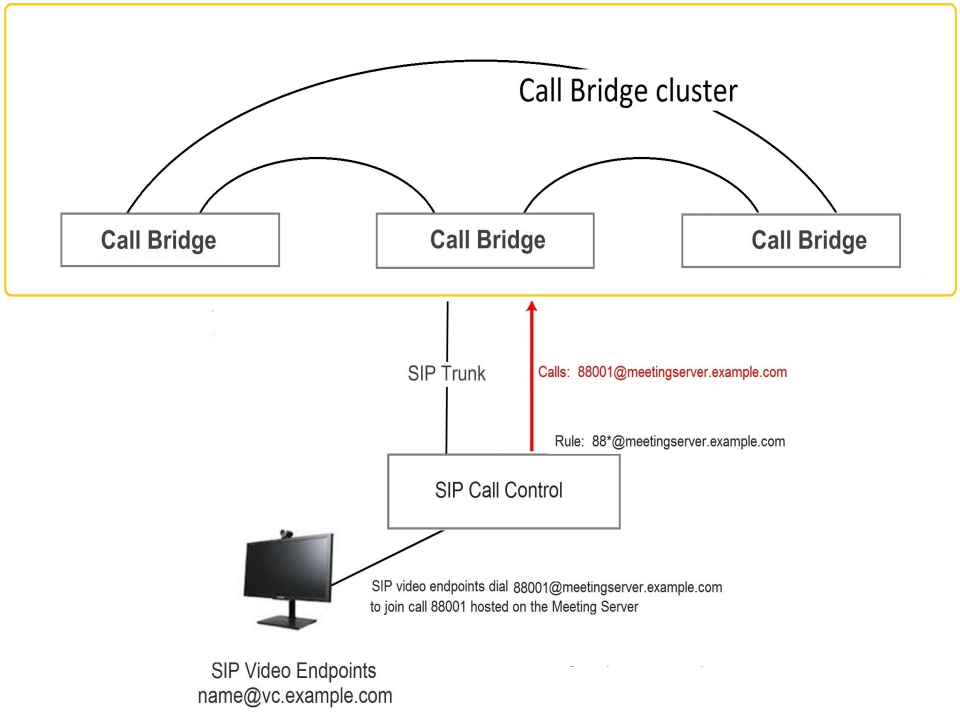
11.1 Introduction

This chapter describes the configuration to enable SIP video endpoints to dial into a meeting hosted on clustered Meeting Servers. Work through the steps in the order provided, adapting the example as appropriate.

11.2 SIP video endpoints dialing a meeting hosted on clustered Meeting Servers

This first step considers the configuration required on the call control device and on the Meeting Servers to direct SIP video endpoints to meetings hosted on clustered Meeting Servers.

Figure 33: Example of SIP video endpoints calling into clustered Meeting Server hosted calls



11.2.1 SIP call control configuration

This example assumes the SIP Call Control is a Cisco VCS, but similar steps are required on other call control devices, for example using the Cisco Unified Communications Manager, see the “Cisco Meeting Server with Cisco Unified Communications Manager Deployment Guide”.

1. Sign in to the VCS as an administrator.
2. Set up a zone to route calls to one of the clustered Meeting Servers
   a. Go to **VCS Configuration > Zones > New**.
   b. Create the zone with the following:
      - H.323 Mode = Off.
      - SIP Mode = On
      - SIP Port = 5060 (5061 if using TLS)
      - SIP Transport = TCP or TLS, as appropriate
      - SIP Accept Proxied Registrations = Allow
      - Authentication Policy = Treat as authenticated
      - SIP Authentication Trust Mode = Off
      - Peer 1 Address = the IP address of the Call Bridge
      - Add other Call Bridges in the cluster as Peer 2, Peer 3 etc.
3. Add a search rule to route calls to the Meeting Server cluster. For example to route any calls on SIP endpoints to a meeting on the Meeting Server cluster using the domain `meetingserver.example.com`.
   a. Go to **VCS Configuration > Dial Plan > Search rules**
   b. Give the rule a suitable name, e.g. `Route EPs to Meeting Server cluster`.
   c. Set the following:
      - Source = Any
      - Request Must Be Authenticated = No
      - Mode = Alias pattern match
      - Pattern Type = Regex
      - Pattern String = `.*@meetingserver.example.com`
      - Pattern Behavior = Leave
      - On Successful Match = Stop
      - Target = the zone you created for the Meeting Server cluster.

11.2.2 Meeting Server configuration

1. Sign in to the API of one of the Meeting Servers in the cluster.
2. Either create a space on the Meeting Server for endpoints to dial into:

a. POST to **/coSpaces** these parameters with values
   - **name=<string>** for example **name="Call 001"**
   - **uri=<user part of URI>** for example. **uri=88001**

   or use an already existing space.

3. Add an inbound dial plan rule for incoming calls to the Meeting Server
   a. POST to **/inboundDialPlanRules** these parameters with values:
      - **domain=<string>** for example **domain="meetingserver.example.com"**
      - **resolveTocoSpaces=true**
      - **resolveToIvrs=true**
      - optional **resolveToUsers=true**
      - **resolveToLyncConferences=true** this is required later, in Section 1.1.2

Note: More information on /inboundDialPlanRulesis given in Section 1.2.2.

---

4. Add an outbound dial plan rule for outbound calls to SIP endpoints via the VCS.
   a. POST to **/outboundDialPlanRules** these parameters with values:
      - **domain=<string>** use the domain to match for the rule, for example **domain="example.com"**
      - **sipProxy=<string>** where <string> is the IP address or FQDN of your VCS
      - **localFromDomain=<string>** where <string> is the FQDN of the Meeting Server cluster
      - **trunkType=sip**

Note: Leave **localContactDomain** blank unless setting up a trunk to Lync (as in Section 1.1.2).

Note: More information on /outboundDialPlanRulesis given in Section 1.2.1.

SIP video endpoints can now dial into a call 88001 hosted on the Meeting Server by dialing **88001@meetingserver.example.com**, and the Meeting Server can call out to SIP endpoints.

Before moving onto creating dial plans for Lync in Chapter 1, consider whether to:

- configure the media encryption setting, see Section 11.3,
- enable TIP support for Cisco CTS endpoints, see Section 11.4,
- configure an Interactive Voice Response (IVR), see Section 11.5

11.3 Media encryption for SIP calls

The Meeting Server supports media encryption for SIP connections, including Lync calls, made to or from the Meeting Server.

SIP media encryption can be set to optional, required or prohibited for SIP calls already in progress (active calls) or to future SIP calls.

1. Sign into the API of one of the Meeting Servers in the cluster.
2. Create a call leg profile with the **sipMediaEncryption** parameter set

   For example POST to **/callLegProfiles** these parameters with values

   - **name=<string>** for example **name="Encrypt media"**
   - **sipMediaEncryption=required**
3. To set media encryption for future SIP calls :
   a. Find the ID of the call leg profile created in step 2.

      GET **/callLegProfiles** will return a list, identify the ID of the call leg profile from the list.

   b. Associate the call leg profile with a coSpace, coSpaceUser, accessMethod or tenant

      For example, POST **/coSpaces/<coSpace id>/accessMethods** these parameters with values

      - **uri=<user part of URI>** for example **uri=exec.reviews**
      - **callLegProfile=<ID from setp 3a>**
4. To set media encryption for an active SIP call:
   a. Identify the active call

      GET **/calls** will return a list of active calls, use the name of the call to identify the ID of the **coSpace** associated with this active call.

   b. Associate the call leg profile created, in step 1, with the coSpace ID from step 3a.

      PUT to **/coSpaces/<coSpace ID>** this parameter with a value

      - **callLegProfile=<ID from setp 3a>**

Note: The **/outboundDialPlanRule** object has a **sipControlEncryption** parameter which allows you to set the control encryption behavior on outbound SIP calls. This ability to separate the control and media encryption allows for a TLS control connection to be used in the absence of media encryption; you can also set the behavior via the Web Admin interface.

11.4 Enabling TIP support

If you use endpoints such as the Cisco CTS range, you need to select TIP protocol support. Use the API as follows:

1. Sign into the API of one of the Meeting Servers in the cluster.
2. Create a call leg profile with the **telepresenceCallsAllowed** parameter set to "true".

   POST to **/callLegProfiles** these parameters with values:

   - **name=<string>** for example, **name="TIP.endpoints"**
   - **telepresenceCallsAllowed=true**
3. Find the ID of the call leg profile created in step 2.

   GET **/callLegProfiles** will return a list, identify the ID of the call leg profile from the list
4. If you want TIP calls to be allowed for any meeting, then associate the call leg profile created in step 2 with **/system/profiles.**

   For example, PUT to **/system/profiles** the parameter **callLegProfile=<ID from setp 3>**
5. If you want TIP calls to be allowed only for a particular coSpace, coSpaceUser, accessMethod or tenant then associate the call leg profile with the appropriate object.

   For example, POST to **/coSpaces/<coSpace id>/accessMethods** with **uri=TIP.meetings** and **callLegProfile=<ID from setp 3>**

Note: TIP calls need Rx and Tx bandwidth settings of at least 4000000. Set bandwidth settings by POSTing to **/calls/<call id>/callLegs** or **/calls/<call id>/participants** with **bandwidth=4000000**.

11.5 IVR configuration

You can configure an Interactive Voice Response (IVR) to manually route to pre-configured calls. Incoming calls can be routed to the IVR where callers are greeted by a prerecorded voice message inviting them to enter the ID number of the call or space that they want to join. Video participants will see a welcome splash screen. After entering the ID, users are routed to the appropriate call or space, or prompted to enter a PIN if the call or space has one. (Callers are disconnected after the third incorrect call ID.)

If you intend to use an IVR follow these instructions:

1. Sign into the API of one of the Meeting Servers in the cluster.
2. Create an IVR:

   POST to **/ivrs** with **uri=<number>** where <number> is the numeric call ID that users call to reach the IVR.
3. Create an **/accessQuery** with the external phone number that users have to call to reach the IVR

POST to **`/accessQuery`** the parameter **`ivr=<string>`** where <string> is the external telephone number to reach this ivr.

4. Match incoming calls to this IVR:

   POST to **`/inboundDialPlanRules`** with **`resolveToIvrs=true`**

5. Configure the appropriate routing on your SIP Call Control to ensure that calls to the numbers set in the previous steps are routed to the Meeting Server.

11.6 Next steps

Now follow the steps in Chapter 1 to configure dial plans to integrate a Meeting Server cluster with Lync deployments.

12 Dial plan configuration – integrating Lync/Skype for Business

Throughout this chapter, references to Microsoft Lync also mean Microsoft Skype for Business.

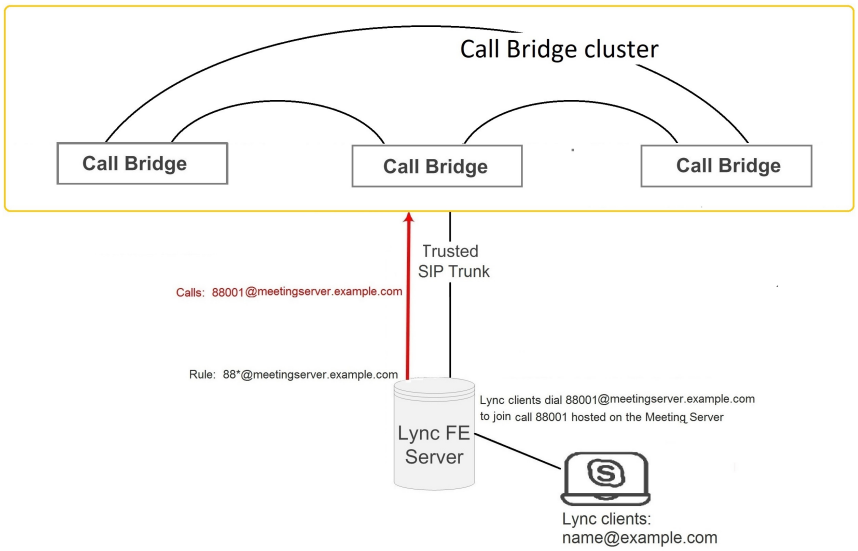
Note: For Call Bridge integration with Lync Edge, the Call Bridge needs its own login account; for a Call Bridge cluster integration with Lync, each Call Bridge in the cluster needs its own unique login account to Lync Edge.

For each Lync call to or from the Call Bridge, the Meeting Server requests TURN resources from the Lync Edge using the Call Bridge account. Until the call is disconnected, that TURN resource is considered " Used" from a Lync point of view. Lync will only allow up to 12 TURN allocations per user account; therefore, only 12 calls are possible for each Call Bridge registration.

12.1 Lync clients dialing into a call on clustered Meeting Servers

This section provides the equivalent of the previous section but for Lync endpoints joining a meeting hosted on clustered Meeting Server. It uses the same call number/URI; adapt the example as appropriate.

Figure 34: Example of Lync clients calling into meetings hosted on clustered Meeting Servers



12.1.1 Lync Front End server configuration

CAUTION: This section provides an example for configuring a static route between a Lync FE server and a Meeting Server cluster, it is only a guideline and is not meant to be an explicit set of instructions for you to follow. Cisco strongly advises you to seek the advice of your local Lync server administrator on the best way to implement the equivalent on your server's configuration.

Note: Before configuring a static route from the Lync FE server, ensure that you have installed certificates on the clustered Meeting Servers which will be trusted by the Lync FE server. Refer to the Certificate Guidelines for Scalable and Resilient deployments.

To route calls originating from Lync clients to the Meeting Server cluster, add a Lync static route pointing to the cluster. This involves setting the Meeting Server cluster as a trusted application for the Lync FE server and adding the static route; for an overview of the configuration, see this link.

1. Open the Lync Server Management Shell.
2. Create a new application pool that will contain the Meeting Server cluster as a trusted application.

   ```
   New-CsTrustedApplicationPool -Identity
   callbridgeclusterfqdn.meetingserver.com -ComputerFqdn
   callbridge1fqdn.meetingserver.com -Registrar fqdn.lyncserver.com -site 1 -
   RequiresReplication $false -ThrottleAsServer $true -TreatAsAuthenticated
   $true
   ```

   Replacing:

   callbridgeclusterfqdn.meetingserver.com with the FQDN of the Meeting Server cluster. This should be an A record that resolves to the IPs of all Call Bridge cluster peers.

   callbridge1fqdn.meetingserver.com with the FQDN of your first Meeting Server, the identity MUST be the CN specified in the Call Bridge's certificate.

   fqdn.lyncserver.com with your Lync FE Server or Pool FQDN.
3. Add the other Meeting Servers to the application pool. Repeat this step for all the other Call Bridges in the cluster (not the first one that you specified in step 2):

   ```
   New-CsTrustedApplicationComputer -Identity callbridge2fqdn.meetingserver.com -
   pool callbridgeclusterfqdn.meetingserver.com
   ```

   Replacing:

   callbridgeclusterfqdn.meetingserver.com with the FQDN of the Meeting Servercluster. This should be an A record that resolves to the IPs of all Call Bridge cluster peers.

   callbridge2fqdn.meetingserver.com with the FQDN of your second/third/fourth/etc... Meeting Server.

4. Add the Meeting Server cluster as a trusted application to the application pool.

   ```
   New-CsTrustedApplication -ApplicationId meetingserver-application -
   TrustedApplicationPoolFqdn callbridgeclusterfqdn.meetingserver.com -Port
   5061
   ```

   Replacing:

   callbridgeclusterfqdn.meetingserver.com with the FQDN of the Meeting Server cluster. This should be an A record that resolves to the IPs of all Call Bridge cluster peers.

5. Create the static route between the Meeting Server cluster and the Lync FE server.

   ```
   $x1=New-CsStaticRoute -TLSRoute -Destination
   "callbridgeclusterfqdn.meetingserver.com" -MatchUri
   "meetingserver.example.com" -Port 5061 -UseDefaultCertificate $true
   ```

   Replacing:

   callbridgeclusterfqdn.meetingserver.com with the FQDN of the cluster. This should be an A record that resolves to the IPs of all Call Bridge cluster peers.

   meetingserver.example.com with the URI matching the domain used for all of your Meeting Server calls.

6. Add the new static route to the existing collection of static routes

   ```
   Set-CsStaticRoutingConfiguration -Identity global -Route @{Add=$x1}
   ```

7. Create a static route between each Call Bridge in the Meeting Server cluster and the Lync FE server. This ensures that content or chat can be shared from Lync via the correct Call Bridge.

   ```
   $x<n>=New-CsStaticRoute -TLSRoute -Destination
   "callbridge1fqdn.meetingserver.com" -MatchUri
   "callbridge1fqdn.meetingserver.com" -Port 5061 -UseDefaultCertificate $true
   ```

8. Add the new static route to the existing collection of static routes

   ```
   Set-CsStaticRoutingConfiguration -Identity global -Route @{Add=$x<n>}
   ```

9. Repeat steps 7 and 8 for each Call Bridge in the cluster.

10. Optional. Before enabling the static route, consider changing the default screen resolution for Lync calls from the default of VGA to HD720p. To enable HD720p on Lync:

    ```
    Set-CsMediaConfiguration -MaxVideoRateAllowed Hd720p15M
    ```

11. Enable the new static route.

    ```
    Enable-CsTopology
    ```

Note: Users may have to logout and login again to update to the new HD720p setting, all other settings are automatic and should work within a few minutes.

12.1.2 Adding a dial plan rule to clustered Meeting Servers

Note: Using the Web Admin to create an Outbound Dial Plan Rule will result in the dial plan rule applying to all Call Bridges in a cluster. To apply the Outbound Dial Plan Rule to a specific Call Bridge or Call Bridge group you need to use the API and POST on **`/outboundDialPlanRules`** with the **`scope`** parameter set appropriately.

CAUTION: Currently, deploying clustered Meeting Servers with Lync will result in the Lync client seeing an incoming call from a SIP endpoint, as coming from the Call Bridge FQDN not the actual SIP domain. This may cause issues with Lync desktop sharing landing on a different Call Bridge in the cluster. To work around this issue, create an outbound dial plan rule for each Meeting Server in the cluster with **`scope = callbridge`** and **`callBridge=<callbridge id>`**. This will allow each Call Bridge in the cluster to use its own server FQDN as the **`localFromDomain`**.

1. Sign in to the API of one of the Meeting Servers in the cluster.
2. Add an outbound dial plan rule to the Meeting Server. This needs to be done for every Call Bridge in the Meeting Server cluster.
   a. POST to **`/outboundDialPlanRules`** these parameters with values:
      - **`domain=<string>`** where <string> is the Lync domain that will be matched for calls that need to be sent to Lync, for example **`domain="example.com"`**
      - **`sipProxy=<string>`** where <string> is the IP address or FQDN of your Lync FE pool or server, or else leave blank, see note below.

        Note: about **`sipProxy`** parameter:
        - if <string> is left blank, the Meeting Server will perform a DNS SRV lookup for the domain using **`_sipinternaltls._tcp.<yourlyncdomain>.com`**
        - if <string> holds the Front End Pool (or Lync SIP domain), the Meeting Server will first perform a DNS SRV lookup for that defined domain using **`_sipinternaltls._tcp.<Server address>.com`**and then perform a DNS A record lookup for the Host if the SRV lookup fails to resolve

      - **`localContactDomain=callbridge1fqdn.meetingserver.example.com`**

        Note: The local contact domain field should contain the Fully Qualified Domain Name (FQDN) for the Meeting Server. It should only be set if setting up a trunk to Lync.

      - **`localFromDomain=<string>`** this is the domain that you want the call to be seen as coming from (the Caller ID), for example **`localFromDomain="callbridge1fqdn.meetingserver.example.com"`**

Note: If you leave `localFromDomain` blank, the domain used for the Caller ID defaults to that entered as the `localContactDomain`.

- `trunkType=Lync`
- `scope=callbridge` the outbound dial plan rule is only valid for the specified Call Bridge
- `callBridge=<callbridge id>` this is the ID of the Call Bridge for which the outbound dial plan rule is valid. For example, `callBridge=callbridge1`.

Note: More information on **/outboundDialPlanRules** is given in Section 1.2.1.

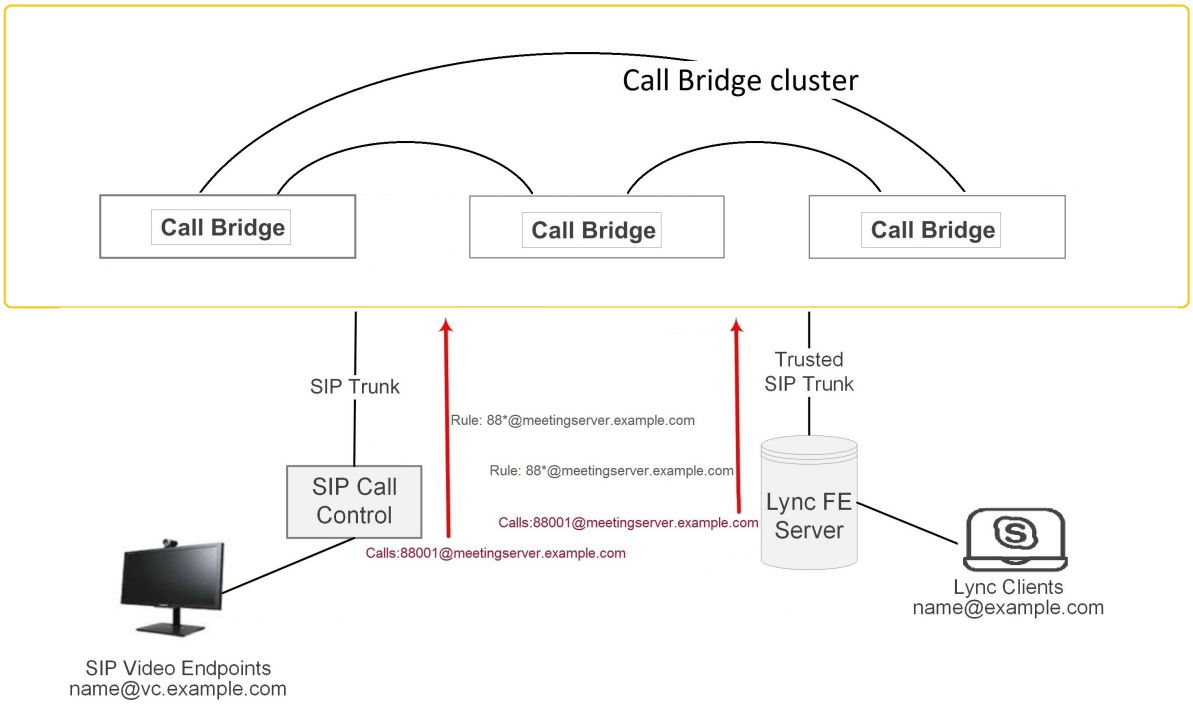
Lync clients can now dial into a call 88001 hosted on the clustered Meeting Server by dialing 88001@meetingserver.example.com.

12.2 Integrating SIP endpoints and Lync clients

To allow SIP endpoints to dial a Meeting Server space, implement the steps in Section 11.2; to allow Lync clients to dial a Meeting Server space, implement Section 12.1.

Then both SIP video endpoint users and Lync client users will be able to enter the same call by dialing <call_id>@meetingserver.example.com

Figure 35: Example of SIP video endpoints and Lync clients calling into Meeting Server hosted meetings

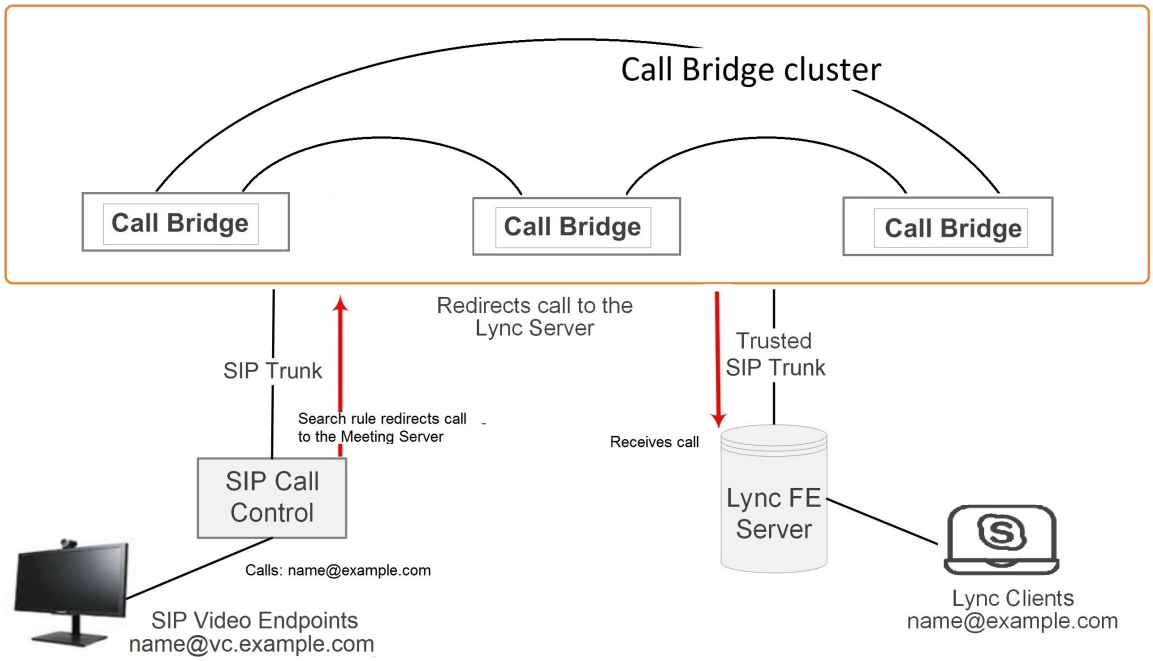


12.3 Adding calls between Lync clients and SIP video endpoints

This section assumes the configuration described for outbound calls and incoming callshas been completed. It expands the example to allow Lync and SIP video endpoints to call each other in a call using the Meeting Server as a gateway to transcode the video and audio (see the figure below).

Note: The `/outboundDialPlanRule` object was used previously to set up a SIP trunk from the Meeting Server to the Cisco VCS. In order to configure the Meeting Server to act as a "point-to-point bridge" between Lync and SIP environments, you need to configure call forwarding as described in this section and also set up a SIP trunk from the Meeting Server to other SIP call control devices you are using such as the Lync FE server, Cisco VCS, CUCM, Avaya CM or Polycom DMA.

Figure 36: Example of SIP video endpoints and Lync clients in calls



In this example:

- A Lync user can dial `<name>@vc.example.com` to set up a call with a SIP video endpoint who is `<name>@vc.example.com`.
- A SIP video endpoint can dial `<name>@example.com` to set up a call with a Lync endpoint who is `<name>@example.com`.

Adapt the example as appropriate.

12.3.1 Lync Front End server configuration

To allow Lync clients to call SIP video endpoints:

- Add a Lync static route pointing to the Meeting Servercluster for **vc.example.com**. For more information, see Section 12.1.1.

this will route Lync client calls to SIP video endpoints.

12.3.2 VCS configuration

To allow SIP video endpoint to call Lync clients:

- Add a search rule on the VCS (SIP call control device) to route calls with the suffix **@example.com** to the Meeting Server.

this will route SIP video endpoint calls to Lync clients.

12.3.3 Meeting Server configuration

Create two forwarding rules on the Meeting Server, one to forward calls to SIP endpoints, and the other to forward calls to Lync clients. Then create two outbound dial plan rules one to route outbound calls to SIP endpoints, and the other to route outbound calls to Lync clients.

1. Sign into the API of one of the Meeting Servers in the cluster.
2. Using the API object **/forwardingDialPlanRules**, create two new rules:
   a. POST to **/forwardingDialPlanRules** a forwarding dial plan rule for calls to vc.example.com for SIP endpoints.
      - **matchPattern=<string>**, where <string> is the domain to match in order to apply the forward dial plan rule, for example, **matchPattern=vc.example.com**. Wildcards are permitted in any part of a domain matching pattern, but do not use "matchPattern=*" as a match all, otherwise you will create call loops.
      - **priority=number**any value is acceptable, including 0 if there are no other forwarding rules configured. To ensure this rule is always used, set its priority as the highest of any rules configured, .

        Note: **forwardingDialPlanRules** are applied in order of priority; highest priority first. If two **matchingPatterns** match a destination domain, then the rule with the higher priority is used.

      - **action=forward**, this causes matching call legs to become point-to-point calls with a new destination. Note: setting **action=reject** causes the incoming call leg to terminate.
      - **callerIdMode=regenerate**this will use the domain from the outbound dial plan.

- **destinationDomain=<string>** only specify this parameter if calls that are forwarded with this rule are to have their destination domain rewritten with the value of this string.

b. For each Call Bridge in the Meeting Server cluster, POST to **/forwardingDialPlanRules** a forwarding dial plan rule for calls to example.com for Lync clients .

For callBridge1:

- **matchPattern=<string>**, where <string> is the domain to match in order to apply the forward dial plan rule, for example, **matchPattern=example.com**.
- **priority=number**any value is acceptable, including 0 if there are no other forwarding rules configured. To ensure this rule is always used, set its priority as the highest of any rules configured, .

Note: **forwardingDialPlanRules** are applied in order of priority; highest priority first. If two **matchingPatterns** match a destination domain, then the rule with the higher priority is used.

- **action=forward**, this causes matching call legs to become point-to-point calls with a new destination. Note: setting **action=reject** causes the incoming call leg to terminate.
- **callerIdMode=regenerate**this will use the domain from the outbound dial plan. This will enable the Lync client to call back a missed call.
- **destinationDomain=<string>** only specify this parameter if calls that are forwarded with this rule are to have their destination domain rewritten with the value of this string.

3. Using the API object **/outboundDialPlanRules**, create two new rules:

a. On one of the clustered Meeting Servers, create a dial plan for calls to SIP endpoints. POST to **/outboundDialPlanRules** these parameters with values:

- **domain=<string>** for example **domain="meetingserver.example.com"**
- **sipProxy=<string>** where <string> is the IP address or FQDN of your VCS
- **localFromDomain=<string>** where <string> is the FQDN of the Meeting Servercluster
- **trunkType=sip**
- **scope=global** all Call Bridges in the cluster will use this dial plan

b. For each Call Bridge in the Meeting Server cluster, POST to **/outboundDialPlanRules** a dial plan for calls to example.com for Lync clients . This is a repeat of step 2 in Section 12.1.2.

For callbridge1:

- **domain=<string>** where <string> is the Lync domain that will be matched for calls that need to be sent to Lync, for example **domain="lync.example.com"**
- **sipProxy=<string>** where <string> is the IP address or FQDN of your Lync FE pool or server, or else leave blank, see note below.

  Note: about **sipProxy** parameter:

  - if <string> is left blank, the Meeting Server will perform a DNS SRV lookup for the domain using **_sipinternaltls._tcp.<yourlyncdomain>.com**
  - if <string> holds the Front End Pool (or Lync SIP domain), the Meeting Server will first perform a DNS SRV lookup for that defined domain using **_sipinternaltls._tcp.<yourlyncdomain>.com**and then perform a DNS A record lookup for the Host if the SRV lookup fails to resolve

- **localContactDomain=callbridge1fqdn.meetingserver.example.com**

  Note: The local contact domain field should contain the Fully Qualified Domain Name (FQDN) for the Meeting Server. It should only be set if setting up a trunk to Lync.

- **localFromDomain=<string>** this is the domain that you want the call to be seen as coming from (the Caller ID), for example **localFromDomain="callbridge1fqdn.meetingserver.example.com"**

  Note: If you leave **localFromDomain** blank, the domain used for the Caller ID defaults to that entered as the **localContactDomain**.

- **trunkType=Lync**
- **scope=callbridge**the outbound dial plan rule is only valid for the specified Call Bridge
- **callBridge=<callbridge id>** this is the ID of the Call Bridge for which the outbound dial plan rule is valid.

Repeat for each Call Bridge in the cluster

SIP video endpoints can now call Lync clients by dialing **<name>@example.com**, and Lync clients can call SIP video endpoints by dialing **<endpoint>@vc.example.com**.

12.4 Integrating Lync using Lync Edge service

For NAT traversal using the Lync Edge server, follow the configuration steps in this section to configure Lync Edge settings on the Meeting Server. This is required to support Dual Homed

Conferencing or if the Lync Edge performs the TURN/ICE role for Lync calls, rather than the Meeting Server.

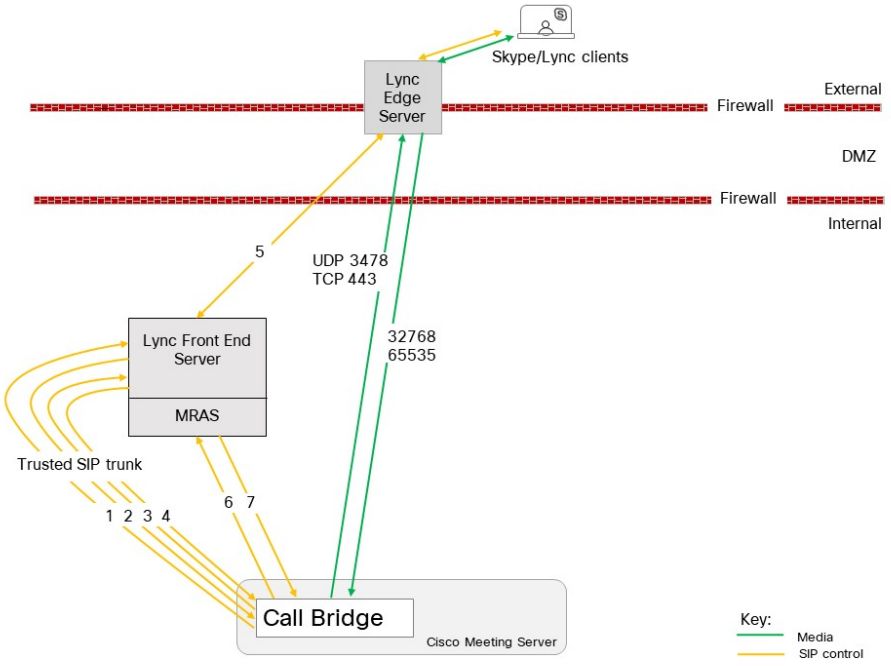
12.4.1 Lync Edge call flow

To establish a call from the Meeting Server to the Lync Edge server (see Figure 37 below):

1. The Call Bridge makes a “register” SIP call to the Lync Front End server.
2. The “register” is acknowledged.
3. The Call Bridge sends a “service” to the Lync Front End server.
4. The Front End server returns the URI of the media relay authentication server (MRAS). (The Lync Edge Server acts as a MRAS.)
5. The Lync client initiates an incoming call.
6. The Call Bridge sends “service” messages to the Lync Front End server to request MRAS credentials to use the Lync Edge MRAS service
7. The Lync Front End server returns the credentials for the Call Bridge to use, as well as the UDP and TCP ports, and the MRAS URI once again
8. The Call Bridge resolves this MRAS URI using DNS and starts sending STUN messages directly to the Lync Edge server
9. The call media then flows directly between the Call Bridge and Lync Edge’s TURN server on UDP port 3478 and returns from the Lync Edge server to the Call Bridge on a port in the ephemeral range above.

Therefore the following ports need to be opened in the firewall for the media between Call Bridge and the Lync Edge server: UDP 3478 outgoing and 32768-65535 incoming.

Figure 37: Call Bridge to Lync Edge server call flow



12.4.2 Configuration on Meeting Server to use Lync Edge

To use a Lync Edge server, you need to configure the Lync Edge settings through the Web Admin interface of each Call Bridge in the Meeting Server cluster, not the API.

When a Lync Edge server is configured, it takes the TURN / ICE role for Lync calls, and so at some level is an alternative to the TURN server settings of the Meeting Servers.

You also need to create a Lync user client account to set up the Meeting Server- Lync Edge server configuration. A different Lync user client account is required for each Call Bridge in the cluster. If the Meeting ServerAPI is used in step 4 below, rather than the Web Admin interface, then the Lync user accounts will be shared by the Call Bridges in the cluster, rather than each Call Bridge having its own.

For each Meeting Server in the cluster, follow these steps to set up the Meeting Serverto use the Lync Edge server:

1. Ensure that you have the appropriate DNS records in place; see Appendix A for a list of DNS records needed for the scalable and resilient server type deployment.
2. Create a new user in your LDAP directory, just as you would any other user in your directory, for example, firstname=”edge”, second name = “user”.
3. Log into the user manager on your Lync FE Server and create a Lync Client user from the user you created in the previous step. Do this in the same way as you would any other user to enable them to use Lync. Using the example name above creates a Lync client user called **edge.user@lync.example.com**

4. Sign in to the Web Admin interface of the Meeting Server, go to **Configuration>General** and configure the Lync Edge settings:
   - **Server address** this field is the address of the Lync Front End server (or a host name that resolves to it)
   - **Username** this field is the username of the Lync client created in step 3
   - **Number of registrations** this field overcomes a feature of the Lync Edge server that limits the number of simultaneous calls that it will run for one registered device. By entering a number greater than 1, the Call Bridge will make that number of registrations, thereby increasing the number of simultaneous calls that the Meeting Server can make out through the Lync Edge server.

     Entering a number greater than 1 adds a number to the end of your Lync Edge username and registers with the resulting username. For example, if you configured Username as edge.user@lync.example.com and set **Number of registrations**=3, you will need to create the following users in your Lync environment so that they can be used with the Edge server:
     edge.user1@lync.example.com
     edge.user2@lync.example.com
     edge.user3@lync.example.com

     Leave **Number of registrations** blank to only make a single registration for example **edge.user@lync.example.com**.

     Note: We recognize that using **Number of registrations** to increase the number of simultaneous calls requires some administrative overhead; however it is due to a limitation of the Lync Edge server as explained above.

     Note: There is no need to enter the password for the Lync users because the Lync Front End server trusts the Call Bridge.

Points to note about configuring the Lync Edge:

- The Meeting Server supports Lync content (presentations contributed over RDP) from external Lync clients whose media arrives via the Lync Edge server. In addition, space (URIs) now report back as busy or available based on how many participants are currently in the space so that Lync clients that have spaces in their favorites can see the space status.
- If you are using a Lync AVMCU, you need to configure the Lync edge settings in order to register with the Lync Front End server.
- web apps continue to use the Meeting Server TURN server even if a Lync Edge server is configured. Note: web app users can only be added to a Lync meeting by a Lync client, they can not dial directly into a Lync meeting.

- If you have a Lync Edge server configured, all Lync calls will use that server for ICE candidate gathering and external media connectivity. If you do not have a Lync Edge server configured, but have configured a Cisco Expressway in your deployment, then the Lync calls will be handled by the configured TURN server in the Expressway.
- In a typical Lync Edge deployment, the internal interface of the Lync Edge server will not have a default gateway defined; only the external interface has a default gateway defined. If the Call Bridge interface is not on the same local subnet as the internal interface of the Lync Edge server, then you must define a static and persistent network route to the Lync Edge server so it can route packets to the Meeting Server correctly, using the internal interface. To add a static and persistent network route to the Lync Edge Server, open CMD and issue the command below , replacing the example data with your own IP information.

  Example Command:

  ```
  route add –p 10.255.200.0 mask 255.255.255.0 10.255.106.1
  ```

  In this example a network route is added that allows the entire subnet of 10.255.200.0 to route through the gateway of 10.255.106.1; 10.255.106.1 is the gateway of the subnet for the internal interface on the Lync Edge server.

  Failure to add this route will result in all STUN packets sent by the Meeting Server to the Lync Edge server to go unanswered, which can result in call failures.

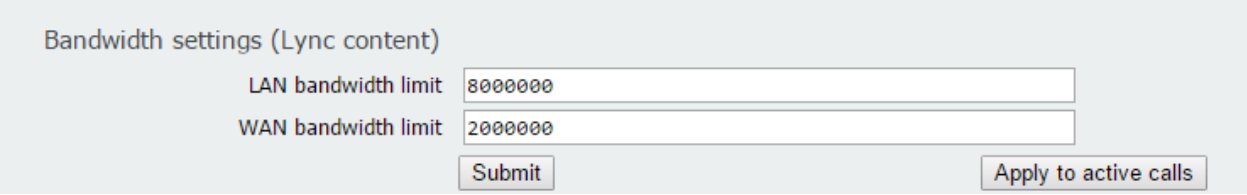
12.5 Controlling the bandwidth for sharing content on Microsoft Lync and Skype for Business calls

Note: Controlling the bandwidth for sharing content on Microsoft Lync and Skype for Business calls needs to be done for each Call Bridge in the Meeting Server cluster, and can only be done via the Web Admin interface. It cannot be done via the API.

The Call Bridge imposes a limit on the amount of bandwidth used for outgoing Lync presentation media. For calls where the connection is directly to the host computer, the LAN bandwidth limit will be applied; for all other cases, for example when the connection involves traversal across the DMZ, as for remote Lync clients, the WAN bandwidth limit will be applied. The default limits are: 8 Mbytes for the LAN bandwidth and 2 Mbytes for the WAN bandwidth.

You can change the bandwidth used to share content with Lync calls through the Web Admin interface. Navigate to **Configuration>Call settings** and set the LAN and WAN bandwidth limits for Lync content. Click on the **Submit** button and click on **Apply to active calls** button if appropriate.

Figure 38: Setting bandwidth for Lync content sharing



12.6 Direct Lync federation

The Meeting Server supports direct federation with Microsoft Lync, by putting the Call Bridge on a public IP address with no involvement from NAT. This allows calls to be made from the Meeting Server direct to any Lync domain and vice versa.

To allow inbound calls you must:

1. Create the DNS SRV record **`_sipfederationtls._tcp.domain.com`**that points to the FQDN of the Meeting Server. This step is required as Call Bridge will need to have a public IP, and NAT is not supported in this scenario.

2. Add a DNS A record that resolves the FQDN of the Meeting Server to a public IP address.

3. Upload a certificate and certificate bundle to the Meeting Server that complies with the following:

   a. The certificate must have the FQDN as the CN, or if using a certificate with a SAN list then ensure that the FQDN is also in the SAN list. Note: if the certificate contains a SAN list, then Lync will ignore the CN field and only use the SAN list.

   b. The certificate must be signed by a public CA.

   Note: you are advised to use the same Certificate Authority (CA) that is trusted by Lync Front End servers. Contact your Lync adviser for details of the CA and for support on the Meeting Server-Lync integration.

   ---

   c. The certificate bundle must contain the Root CA's certificate and all intermediate certificates in the chain in sequence, so that a chain of trust can be established.

   Note: for more information on certificates refer to the Introduction in the Cisco Meeting Server Certificate Guidelines.

   ---

   d. Open the appropriate Firewall ports as stated in Appendix 1 for example: TCP 5061, UDP 3478, UDP 32768-65535, TCP 32768-65535

For outbound calls from the Meeting Server:

1. Create an outbound dial rule, leave the **domain** and **sipProxy** fields blank, and set **trunk** type as lync. Also set the appropriate **localContactDomain** and the **localFromDomain** fields.

   If specifying individual domains in outbound dial plan rules, ensure that all domains configured on the Lync side have been added. The domains in use can be read from the Lync Server Topology Builder. Note that if additional domains are later added to Lync, then these should also be added to the outbound dial plan rules.

12.7 Calling into scheduled Lync meetings directly and via IVR

Pre-requisite on Lync deployment: This feature requires a working Lync deployment with telephone dial-in capabilities already enabled. The Lync deployment requires one or more on-prem Lync Front End servers to be configured.

Note: The on-prem Lync Front End servers need to be configured even if your Lync deployment does not support external Lync or Skype for Business clients.

The Meeting Server supports calling into a scheduled Lync meeting from WebRTC or SIP endpoint, using the Lync call ID to join the call; web app users can only be added to a Lync meeting by a Lync client. This feature requires one or more Lync Front End servers to be configured on the Meeting Server for conference lookup. You can configure one via the Web Admin interface under the Lync Edge settings from **Configuration > General**, and one or more via the API (create them as TURN servers with type " lyncEdge"). Refer to Configuration on Meeting Server to use Lync Edge for instructions on how to do this. If there are multiple FE servers in a Pool, use the Pool FQDN as the Server Address.

Note: For Lync meeting resolution, the Meeting Server uses the Lync meeting ID and DNS lookup of _sipinternaltls._tcp.lync-domain, rather than outbound rules. Set DNS SRV record **_sipinternaltls._tcp.lync-domain** on your DNS server or if you do not want to use a DNS SRV record then setup a record on the Meeting Server with the command **dns app add rr <DNS RR>**. For more information on using the dns app command see the MMP Command Line Reference; for a list of DNS records needed for the scalable and resilient type deployment see Appendix 1.

Configure the Lync Front End servers, then follow the task sequence in Table 7 below:

Table 7: Task sequence to configure Lync Front End servers

Sequence	Task	Via the API	On the Web Admin Interface - Do not use for clustered Meeting Servers
1	Configure the Call Bridge IVR(s) to allow entry of Lync conference IDs	If you have set up IVRs through the API: Set **`resolveLyncConferenceIds`** to **`true`** for the configured IVR	If you have set up an IVR via the Web Admin Interface: Go to **Configuration > General** in the IVR section, set **Joining scheduled Lync conferences by ID** to **allowed**
2	Allow direct dialing to Lync conference IDs from standard SIP systems. Note: you may choose to extend an existing configured domain to allow Lync conference access, or to create a new one for this purpose.	Set **`resolveToLyncConferences`** t to **`true`** on the incoming dial plan rule	Go to **Configuration > Incoming calls**, and for one or more configured call matching domains, set **Targets Lync** to **yes**
3	Allow Lync conference ID entry via the Web Bridge call join interface	If you have set up Web Bridges through the API: Set **resolveLync ConferenceIds** to **`true`** on the Web Bridge	If you have set up the Web Bridge via the Web Admin Inteface: Go to **Configuration > General** in the Web bridge settings section ensure that **Joining scheduled Lync conferences by ID** is set to **allowed**

If a call is being matched against Lync conference IDs, the Call Bridge first checks that the call ID does not apply to a space, if it does not then the Call Bridge identifies a Lync Front End server that it has been configured with, that has advertised itself as having the capability to resolve IDs. The Call Bridge queries the Lync Front End server to determine whether the call ID in question corresponds to a Lync conference - if it does, the look up is deemed to have been successful and the call is joined to the Lync call. If the call ID is not recognized as corresponding to a Lync conference then no further Lync Front End servers will be queried.

Note: You may get unexpected results if you add the settings of multiple Lync Front End servers that are in different Lync deployments. For instance, if multiple Lync conferences in different Lync deployments use the same call ID, then more than one Lync Front End server may respond positively to the lookup, in which case the " first" successful Lync resolution is used.

Note: Each participant connecting through a Meeting Server to a Lync meeting is required to have a unique " from:" SIP address to avoid participant conflicts in the Lync AVMCU. Telephone participants connecting through a PSTN gateway are at a high risk of encountering

participant conflicts due to the generic outgoing callerID information. It is recommended that all telephone participants connect to Lync meetings through the Lync PSTN Conferencing/Mediation Server rather than through the Meeting Server Dual Home gateway.

The text in the invitations sent for scheduled Lync meetings can be customized to include the necessary details to allow users to join via the Meeting Server. These details should be placed in the custom footer section. For example **'For SIP/H.323 endpoints, join by calling join@example.com and entering the conference ID above. For WebRTC go to join.example.com and enter the conference ID above.'** The URIs in this must match those configured above. Please see the Microsoft documentation https://technet.microsoft.com/en-us/library/gg398638.aspx for more details.

13 Office 365 Dual Homed Experience with OBTP Scheduling

13.1 Overview

"Office 365 Dual Homed Experience with OBTP (One Button To Push) Scheduling" allows participants to join Office 365 meetings using Cisco endpoints that support OBTP.

The host schedules a meeting using Microsoft Outlook with Skype for Business plugin, and adds participants and conference rooms (including OBTP-enabled endpoints) and a location to meet in.

To join the meeting, participants using a OBTP-enabled endpoint simply push the OBTP button on the endpoint or touchscreen. Skype for Business clients click a link to join the meeting as normal.

Note: If using Office 365, only invited OBTP-enabled endpoints or Skype for Business clients with Office 365 can join the Lync meeting; Cisco endpoints cannot join the meeting manually, via the Meeting Server IVR. This is a key difference to an on-premise Lync deployment,which allows any Cisco endpoint to join manually via the Meeting Server IVR.

Note: "Office 365 Dual Homed Experience with OBTP (One Button To Push) Scheduling" is supported from Version 2.2, and requires Cisco TMS 15.5, and Cisco TMS XE 5.5 or later.

13.2 Configuration

Note: This feature requires the Call Bridge to connect to the public internet in order to contact Office 365. You will need to open TCP port 443 on your firewall for outgoing traffic.

To set up this method of joining Office 365 meetings, configure the Meeting Server with an incoming dial plan rule with request parameter **`resolveToLyncSimpleJoin`** set to **`true`**. This tells the Meeting Server how to resolve the Lync Simple Meet URL sent in the Office 365 invite.

To have the ability to call participants as well as meetings, use an existing outbound dial plan rule to route the outbound calls, or create a new outbound dial plan rule.

13.3 In-conference experience

" Office 365 Dual Homed Experience with OBTP Scheduling" provides the "dual homed experience" with 2-way audio, video and content sharing. Office 365 clients have the familiar

in-conference experience determined by the Lync AVMCU, and participants using OBTP enabled endpoints have a video conferencing experience determined by the Meeting Server. All see the combined participants lists.

Note: Controls on clients do not work conference wide, and can give rise to some strange behavior. For example, if a Skype for Business client mutes an endpoint connected to the Meeting Server then the endpoint will mute, but no notification is sent to the endpoint to say it has been muted; the endpoint cannot unmute itself. If a Skype for Business client mutes all endpoints connected to the Meeting Server and then unmutes them, all the endpoints will remain muted.

Note: ActiveControl functionality such as muting and dropping participants only affect participants on the local Call Bridge and not on the Lync AVMCU.

---

14 Settings for Web Bridge 3

This section explains how to configure the settings through which the Call Bridge communicates with Web Bridge 3. This allows you to use web app video calls and meetings.

If you are testing the web app, follow the instructions in Section 14.2 in the order provided at any time after the initial Meeting Server configuration has been completed. If you are not using web app, skip this chapter.

Note: If your deployment requires the Cisco Expressway Web Proxy to connect to the Web Bridge, then ensure the SAN field of the Web Bridge certificate includes the A record used by the Expressway-C that will connect to the Web Bridge, otherwise the connection will fail. For example, if the Expressway is configured to connect to the Web Bridge on join.example.com, an A record must exist for this FQDN, and the SAN field of the Web Bridge certificate must include join.example.com.

14.1 Web Bridge 3 connections

Table 8 show the ports used for web app connections. Section 14.1.1 describes the call flow between the web app and components in the Meeting Server.

Figure 39: web app port usage

Table 8: Ports required for web app connections

Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
Web Bridge 3	web app	443 (Note 1)	TCP (HTTPS)	Incoming	
Web Bridge 3	web app	80	TCP (HTTP)	Incoming	
Call Bridge	Web Bridge 3				

Note 1: Destination port should be what is configured for Web Bridge 3 https listening port.

14.1.1 Web Bridge 3 call flow

This section describes the call flow between the web app and components in the Meeting Server.

1. The web browser opens an HTTPS connection.
2. User is prompted to **Join meeting** (see step 3) or **Sign in** (see step 4)

3. If **Join meeting** is selected, user is prompted to enter the Call ID/URI and passcode and set their name.
    a. Call details are sent over HTTPS to Web bridge 3; Web Bridge 3 queries the Call Bridge over the C2W connection to validate call details.
    b. If successful, the user is asked to pick media settings.
    c. After choosing media settings, the call details and desired name are sent over HTTPS to the Web Bridge 3; forwarded over C2W to the Call Bridge. Call Bridge will respond with a call access token which is returned to the browser and details theTURN servers to be used by the browser.
    d. Call Bridge requests allocations from its configured TURN server.
    e. Web app requests allocations from the provided TURN server.
    f. The browser opens a websocket connection to the Web Bridge 3, which is forwarded over C2W connection to the Call Bridge. The call access token is sent over this websocket.
    g. The browser and Call Bridge exchange SDP over websocket containing local media IP address/ports as well as media relay address/ports.
    h. ICE negotiation sends UDP packets between all browser media IP address/port combinations and all Call Bridge address/port combinations; attempts TCP connections to TCP media relay address/ports.
    i. Shortest successful media path is used for transmitting media between the browser and Call Bridge, either directly, through a TURN UDP relay, or through a TURN TCP relay (with the TURN server translating media packets between TCP stream and UDP)
4. If **Sign in** is selected, user is prompted to enter Username and Password.
    a. Sent over HTTPS to web bridge, which is forwarded to call bridge to obtain an portal access token if successful.
    b. Enters user portal, all requests are made over HTTPS sending portal access token as header.
    c. If a join call request is made, the flow is the same as above from step 3c onwards, except instead of sending call details and desired name to obtain a call access token, the browser instead sends call details and portal access token.

Useful information: call access tokens and portal access tokens are different, although similar. The portal access token is valid for 24 hours and allows a user to access the user portal. The call access token is only valid for the duration of a user's participation in the call, and is used only to join a call. The only way to obtain a portal access token is by signing in with a user name and password. A call access token can be obtained either by doing a guest join, or by using the portal access token along with the details of the meeting the user wants to join

14.2 Web Bridge 3 settings

Version 3.0 onwards allows you to configure Web Bridge configuration options in a common place rather than solely on a per Web Bridge basis – you can apply the same settings for all, or a specified group of Web Bridges.

The /webBridgeProfiles API object contains the various Web Bridge configuration options. A newly defined Web Bridge profile can be assigned to the individual webBridge objects, or to the top level (global) profile or tenants.

See the section on Web Bridge and Web Bridge Profile Methods in the API Reference Guide for further details on configuring the Web Bridge 3.

14.2.1 How to create and apply a web bridge profile example

Before you begin, ensure that you have installed the Web Bridge 3 certificate and configured the Web Bridge 3 as detailed in Section 4.6. Then follow these steps:

1. To create a webBridgeProfile using the Meeting Server Web Admin interface:
    a. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
    b. From the list of API objects, tap the ► after **/api/v1/webBridgeProfiles**
    c. Click **Create new**.
    d. Set the **name** field to the name you wish to call this web bridge profile.
    e. Set the **resourceArchive** field to the address of any customization archive file that the Meeting Server should use for web bridges using this web bridge profile.
    f. Set the **allowPasscodes** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should allow users to lookup coSpaces (and coSpace access methods) with passcodes in combination with an numeric ID/URI. If this parameter is not supplied, it defaults to **true**.
    g. Set the **allowSecrets** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should allow users to access coSpaces (and coSpace access methods) through a meeting join link with a numeric ID and secret. If this parameter is not supplied, it defaults to **true**.
    h. Set the **userPortalEnabled** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should display the sign-in tab on the index page. If this parameter is not supplied, it defaults to **true**.
    i. Set the **allowUnauthenticatedGuests** field to either **true** or **false**. If set to **true**, guest access is allowed from the landing screen on web bridges using this web bridge profile. If set to false, visitor access is only allowed once users have logged into the User Portal. If this parameter is not supplied, it defaults to **true**.

j. Set the **resolveCoSpaceCallIds** field to either **true** or **false**. This field determines whether or not web bridges using this web bridge profile should accept coSpace and coSpace access method call IDs for the purpose of allowing visitors to join cospace meetings. If this parameter is not supplied, it defaults to **true**.

k. Set the **resolveCoSpaceUris** field to either **off**, **domainSuggestionDisabled** or **domainSuggestionEnabled**. This field determines whether or not this web bridge should accept coSpace and coSpace access method SIP URIs for the purpose of allowing visitors to join cospace meetings. When set to **off**, join by URI is disabled; when set to **domainSuggestionDisabled**, join by URI is enabled but the domain of the URI won't be auto-completed or verified on this web bridge; when set to **domainSuggestionEnabled** join by URI is enabled and the domain of the URI can be auto-completed and verified on this web bridge. If this parameter is not supplied, it defaults to **off**.

l. Click **Create**.

2. Once you've created the profile, you can then add addresses — this is the Web Bridge URI used to generate meeting invites and the cross launch URL for the web app.

Note: From version 3.1 you can also now specify multiple IVR numbers and Web Bridge addresses — up to 32 IVR numbers and up to 32 Web Bridge addresses per Web Bridge profile. These are used when displaying join information, and for generating email invitations.

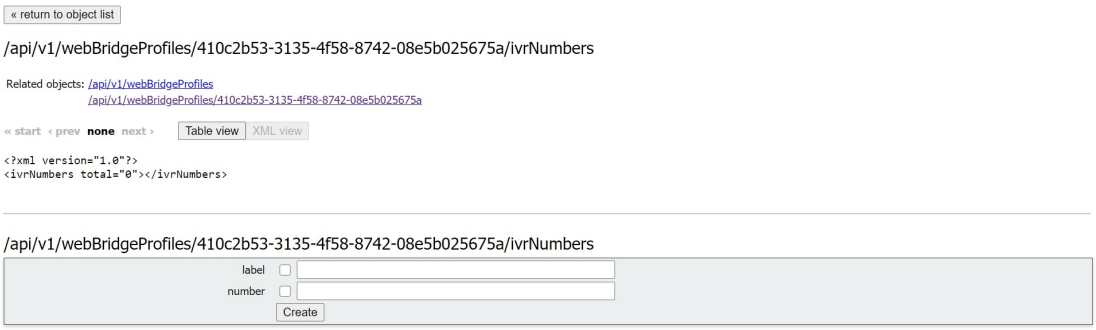
In this example a Web Bridge URI and IVR telephone number are applied to a webBridgeProfile as follows:

a. From the list of API objects tap the ► after **/api/v1/webBridgeProfiles**

b. Click **View or edit**

c. From the resulting " webBridgeProfile object selector window" , click **Select** for the **object id** of the **webBridgeProfile** that you have created in Step 1 that you wish to assign a Web Bridge URI and IVR number to. Enter the **label** and URL **address** for the Web Bridge, and enter the **label** and **number** for the IVR as required.





d. Click **Create**.

3. Assign the ID of the newly created webBridgeProfile to any or all of the following, as required:

    - Top level (global) profile (**/api/v1/system/profiles**)
    - Tenants (**/api/v1/tenants/<*id*>**)
    - WebBridges (**/api/v1/webBridges/<*id*>**)

    In this example an updated webBridgeProfile is assigned to the top level (global) profile as follows:

    a. From the list of API objects tap the ▶ after **/api/v1/system/profiles**

    b. Click **View or edit**

    c. Scroll down the parameters to **webBridgeProfile** and click **Choose**.

    d. From the resulting " webBridgeProfile object selector window" , click **Select** for the **object id** of the **webBridgeProfile** that you have created in Step 1 that you wish to assign to the top level global profile.

    e. Click **Modify**.

    f. The newly assigned webBridgeProfile object id should now be listed under **Object configuration**.

Note: For more information on the web app, see Cisco Meeting Server web app Important Information.

---

15 Recording and Streaming meetings

Prior to 3.0, Meeting Server's internal recorder and streamer components were dependent upon the Meeting Server's internal XMPP server component – in 3.0 this XMPP server is removed. Version 3.0 introduces a new internal recorder and streamer, both SIP-based.

The new internal recorder and streamer components and dialing out to third-party SIP recorders are all configured using SIP URIs, so when recording or streaming is started the administrator-configured SIP URI is called.

15.1 Feature benefits of the new internal SIP recorder and streamer

- The new recorder and streamer support changing layouts. The recorder/streamer get its layout in a similar way to other SIP calls, i.e. from the defaultLayout parameter on the callLegProfile hierarchy or coSpace object. You can also change the layout parameter in the callLeg.
- Custom layouts can be set using the layoutTemplate parameter (you will need a customizations license to implement custom layouts).
- You can control the maximum resolution on a per callLeg basis using the qualityMain parameter in callLegProfiles and callLegs.
- Previously the XMPP streamer only supported 720p resolution, however the new streamer supports up to 1080p resolution and 3.0 allows you to select the streamer resolution using the MMP comand **`streamer sip resolution`**.
- You can choose whether the streamer/recorder receives presentation by changing the presentationViewingAllowed parameter setting in the callLegProfile.
- Improved scalability with the introduction of the new MMP command **`recorder limit`** and **`streamer limit`**.

15.2 Points to note when implementing the new internal SIP recorder and streamer

---

Note: The new internal SIP recorder and streamer service cannot be used as an External recording or streaming service as the services rely on specific SIP header parameters passed by the Meeting Server Call Bridge. When calls from any other source that is not Meeting Server Call Bridge connect, the recorder/streamer will reject the call as it won't locate the specific SIP headers expected.

The recommended deployment for production usage of the recorder is to run it on a dedicated VM with a minimum of 4 vCPU cores and 4GB of RAM. The following table provides an idea of performance and resource usage for each of the recording types.

Table 9: Internal SIP recorder performance and resource usage

Recording Setting	Recordings per vCPU	RAM required per recording	Disk budget per hour	Maximum concurrent recording
720p	2	0.5GB	1GB	40
1080p	1	1GB	2GB	20
audio	16	100MB	150MB	100

Key point to note (applies to new internal recorder component only):

- Performance scales linearly adding vCPUs up to the number of host physical cores.

The recommended deployment for production usage of the streamer is to run it on a dedicated VM with a minimum of 4 vCPU cores and 4GB of RAM. The following table gives an idea of 3 recommended minimum specifications and the number of streams they can handle.

Table 10: Internal SIP streamer recommended specifications

Number of vCPUs	RAM	Number of 720p streams	Number of 1080p streams	Number of audio-only streams
4	4GB	50	37	100
4	8GB	100	75	200
8	8GB	200	150	200

Key points to note (applies to new internal streamer component only):

- Number of vCPUs should not oversubscribe the number of physical cores.
- Maximum number of 720p streams supported is 200 regardless of adding more vCPUs.
- Maximum number of 1080p streams supported is 150 regardless of adding more vCPUs.
- Maximum number of audio-only streams supported is 200 regardless of adding more vCPUs.

15.3 Recording overview

There are two methods of recording meetings when using Meeting Server:

- Third-party external SIP recorder
- Meeting Server internal SIP recorder component

15.3.1 Third-party external SIP recorder support

Meeting Server allows configuration of a third-party external SIP recorder so that when recording is started an administrator-configured SIP URI is called in the same way as the new Meeting Server internal SIP recorder component.

Note: Support for an external third-party SIP recorder still requires Meeting Server recording licenses.

The third-party external SIP recorder feature:

- allows recorders to negotiate BFCP in order to receive separate video and content streams. This gives more flexible options for how recordings are formatted.
- supports the same resolutions as we do for standard SIP calls
- supports the same audio and video codecs as standard SIP calls
- as with the existing Meeting Server internal recorder, any media content sent by the SIP recorder is discarded.

Note: The SIP recorder feature does not support TIP or Active Control.

15.3.2 Meeting Server internal SIP recorder component support

The internal SIP Recorder component (from version 3.0) on the Meeting Server adds the capability of recording meetings and saving the recordings to a document storage such as a network file system (NFS).

The Recorder should be enabled on a different Meeting Server to the server hosting the conferences, see Figure 40. Only locate the Recorder on the same Meeting Server as the Call Bridge which is hosting the conferences (local) for the purposes of testing the deployment.

Where possible it is recommended that the Recorder is deployed in the same physical locality as the target file system to ensure low latency and high network bandwidth. It is expected that the NFS is located within a secure network.

Note: Depending on the mechanism you use to store the recordings you may need to open external firewall ports so that the recorder, uploader and storage system can communicate. For example: NFS running version 2 or 3 of the port mapper protocol uses TCP or UDP ports 2049 and 111.

Note: Do not use the Firewall component on the Meeting Server if using either the Recorder or Uploader.

Note: At the end of recording a meeting, the recording is automatically converted to MP4. The converted file is suitable for placing within a document storage/distribution system, for example, in a network file system (NFS) they are stored in the NFS folder spaces/<space ID; tenant spaces are stored in tenants/<tenant ID>/spaces/<space ID>.

The following figures show the various permitted recording deployments. For information on scalable and resilient deployments, see Section 15.3.3

Figure 40: Permitted deployment for recording: remote mode

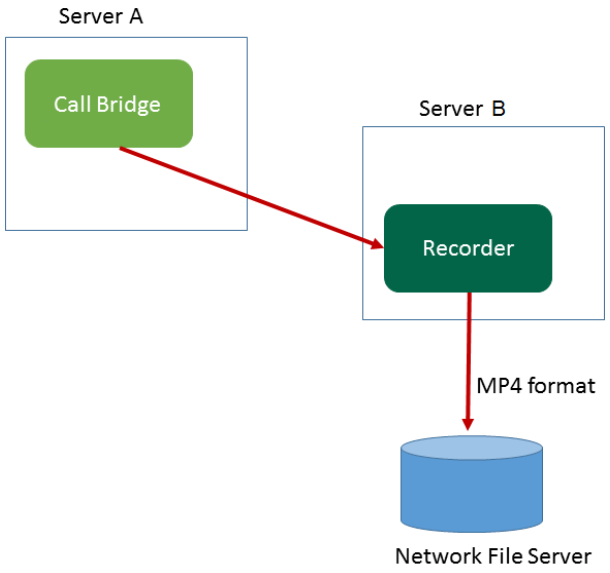
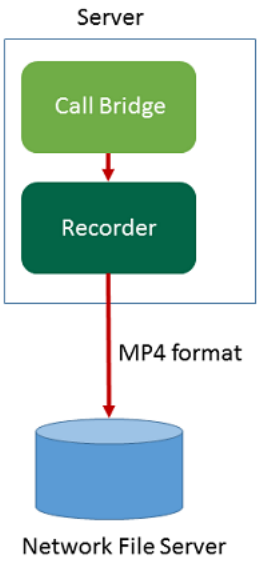


Figure 41: Permitted deployment for testing purposes only: local mode

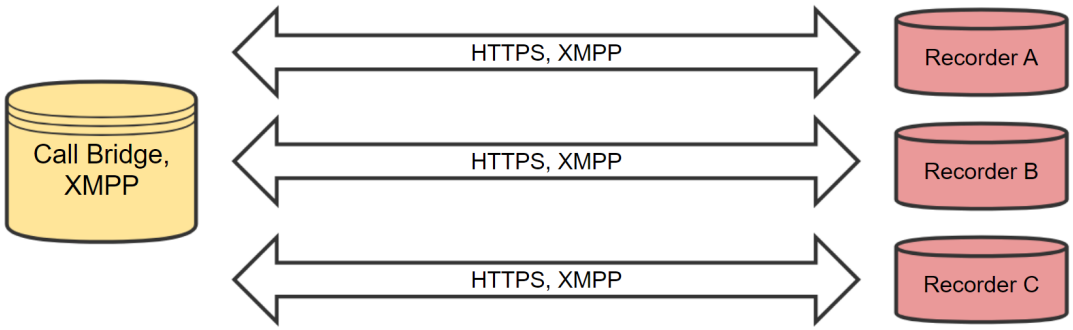


15.3.3 Deploying a Recorder / Streamer for Scalability and Resiliency

When deploying more than one recorder or streamer, we recommend that you deploy them behind a Call Control provider and allow the Call Control provider to give load-balancing and fail-over support. You will need to set the **`sipRecorderUri`** API parameter to point to a dial plan-rule, which can forward calls to the proxy.

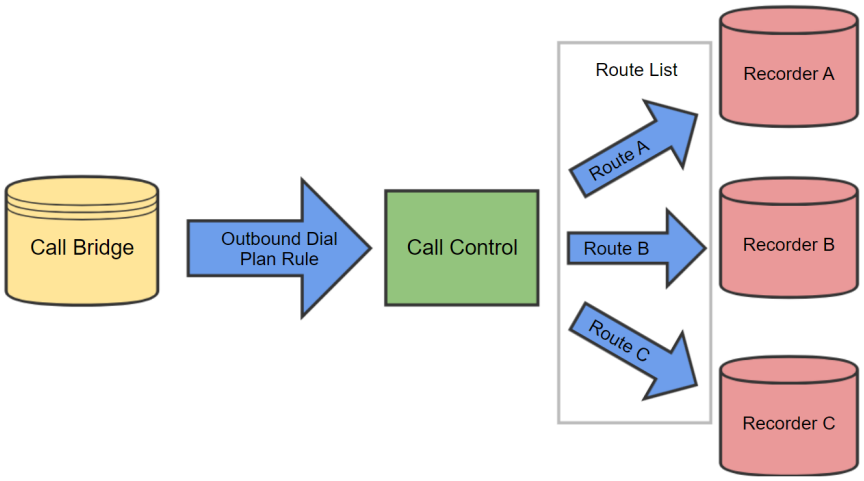
Call Control flow behavior for the new recorder is different to that for the old XMPP recorder. Previously, recorders would connect directly to the Call Bridge as an XMPP client, and send information of availability via an HTTPS link, as shown in Figure 42.

Figure 42: Old XMPP recorder Call Control flow



For the new internal SIP recorder component, Call Control flows through a Call Control provider while media will, in most cases, flow directly between the Call Bridge and the recorder. In some cases, media may also flow through the Call Control provider, depending on the call control provider configuration. This new recorder Call Control flow is shown in Figure 43.

Figure 43: New internal SIP recorder Call Control flow



15.3.3.1 Supported Call Control methods

- **Cisco Unified Communications Manager** – Each recorder in a cluster should be deployed behind a SIP trunk, with each recorder in a cluster associated with the same route list.
- **Cisco Expressway** – Each recorder should be configured to have their own zone, with a route pattern mapping to all recorders in the cluster.

- **Direct Flow** – You may set the dial plan rule's proxy to be the address/FQDN of the recorder / streamer, but this is only recommended when deploying one instance.

For more information, see Cisco Meeting Server 2.x, White Paper on Load Balancing Calls Across Cisco Meeting Servers.

15.4 Example of deploying the new internal SIP recorder component on a VM server

Note: If you plan to save the recordings on a NFS server running Windows 2008 R2 SP1, there is a windows hotfix required to fix permission issues: https://support.microsoft.com/en-us/kb/2485529. Consult your Microsoft Windows Administrator before applying this fix.

This is a two stage process:

- Configuring a Meeting Server recorder via the MMP
- Configuring the recorder URI via the API

Task 1: Configuring a Meeting Server recorder via the MMP

1. Upgrade to version 3.0.
2. SSH into the MMP and login to configure the recorder (enter the MMP command, **`recorder`** to see a list of all available commands).
3. Enter **`recorder nfs <hostname/IP>:<directory>`** to configure the NFS location.
4. Enter **`recorder resolution <audio|720p|1080p>`** to configure the desired resolution (or to only record the audio of calls).
5. Configure the listening interface of the recorder and the SIP TCP and TLS ports to listen on using the MMP command **`recorder sip listen <interface> <tcp-port|none> <tls-port|none>`**. Set the respective port to **`none`** to disable the service:
   a. For example, if you want to only listen on the TLS port and not the TCP port, enter **`recorder sip listen a none 6000`**
   b. Make a note of the ports you've configured if they're not the default TCP/TLS ports (5060/5061) as they will be needed later.

   Note: If you want to listen on the default SIP TCP/TLS ports (5060/5061) you MUST ensure that the Call Bridge is not listening on the same interface, otherwise the ports will clash. You must disable the Call Bridge by removing the corresponding interface, by entering the MMP command **`callbridge listen none`**.

6. Optionally, if TLS is configured, configure the SIP TLS certificates you would like to use:
   a. Enter the MMP command **`recorder sip certs <key-file> <crt-file> [<crt-bundle>]`**

   Note: Note that if SIP TLS certificates are not configured with this option, the SIP TLS service will fail to start.

7. Optionally, if TLS is configured, you can perform TLS verification for SIP on the recorder as follows:
   a. Enter the MMP command **`tls sip trust [<crt-bundle>]`**
   b. Enter the MMP command **`tls sip verify enable`**

   Note: For the TLS connection to be secure we recommend enabling TLS verification.

8. Check the configuration is correct – enter the MMP command **`recorder`** to view the configuration.
9. Enter the MMP command **`recorder enable`** to enable the recorder service.

Task 2: Configuring the recorder URI via the API

Once the new SIP recorder is enabled, it can be configured and used in the Call Bridge in the same way as a third-party SIP recorder, using the **`sipRecorderUri`** API parameter specified in the API call profile object.

If you wish, you can also configure a custom URI that maps to an outboundDialPlan rule (the domain can be anything of your choice, e.g. " recording.com"). You will need to configure an outboundDialPlan rule which tells Meeting Server how to route the domain used in **`sipRecorderUri`** to the recorder. This will allow you to control priority values, encryption, etc. For more information on configuring outboundDialPlan rules, see the " Dial plan configuration – overview" chapter.

Note: The user part of the configured URI (i.e. the part before the '@' symbol) has no special meaning, and for the new internal SIP recorder component, although required, it can usually be anything, e.g. " recording@recorder.com" . However, this may not be the case for third-party SIP recorders which may use the user part of the URI for user credentials, for example. The important part of the URI is the domain part.

To configure the **`sipRecorderUri`** parameter using the Meeting Server Web Admin interface:

1. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
2. From the list of API objects, tap the ▶ after **/api/v1/callProfiles**

3. To configure or modify an existing call profile, select the object id of the required callProfile and fill in the **sipRecorderUri** field with your chosen URI.

   Note: When using the new SIP recorder you only need to use one SIP URI, e.g recording@recorder.com, you don't need to have different SIP URIs on different profiles (it makes no difference).

   ---

4. If you haven't done so already, set the **recordingMode** field to either, **manual** or **automatic** (depending on how you want meetings to be recorded).
5. Click **Modify**.

The updated callProfile can then be assigned to coSpaces, tenants or the top level (global) profile, as required. In this example an updated callProfile is assigned at the global level as follows:

1. Using the Web Admin interface, select **Configuration > API**:
   a. From the list of API objects tap the ► after **/api/v1/system/profiles**
   b. Click **View or edit**
   c. Scroll down the parameters to **callProfile** and click **Choose**.
   d. From the resulting " callProfile object selector window" , click **Select** for the **object id** of the **callProfile** you wish to assign to the top level global profile.
   e. Click **Modify**.
   f. The newly assigned callProfile object id should now be listed under **Object configuration**.

15.4.0.1 callProfile configuration example (if using a matching outbound dial plan rule):

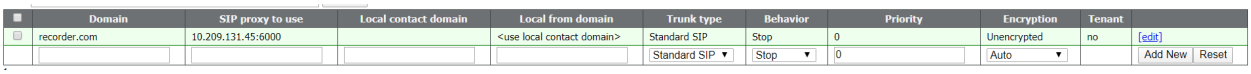
In this example, **recordingMode** is set to **automatic** and **sipRecorderUri** to **recording@recorder.com** using the steps above.

Object configuration	
recordingMode	automatic
sipRecorderUri	recording@recorder.com

From the Meeting Server Web Admin interface select **Configuration > Outbound calls** to see the matching outbound dial plan rule:

	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	
☐	recorder.com	10.209.131.45		<use local contact domain>	Standard SIP	Stop	0	Unencrypted	no	[edit]
					Standard SIP ▾	Stop ▾	0	Auto ▾		Add New Reset

If you configured the recorder in the MMP to use SIP TCP/TLS ports which are different from the default standard ports (5060/5061), you MUST specify the listening port in the **sipRecorderUri** field or in the matching outbound dial plan rule if you are using one, as shown below:



	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	
	recorder.com	10.209.131.45:6000		<use local contact domain>	Standard SIP	Stop	0	Unencrypted	no	[edit]
					Standard SIP	Stop	0	Auto		Add New Reset

If using an outbound dial plan rule, make sure the service of the port specified matches the encryption type, for example, if using the SIP TLS port, set the **Encryption** mode to **Encrypted**.

15.5 Configuring an external third-party SIP recorder

- Specify the SIP recorder – use the sipRecorderUri API parameter for /callProfile objects. If set, this URI is used to dial out to when recording is enabled. If unset, the Meeting Server recorder component (if configured in /recorders) is used.
  a. Use the Web Admin interface of the Meeting Server, select **Configuration**>**API**
  b. From the list of API objects, tap the ► after **/callProfiles**
  c. Either click on the **object id** of an existing call profile or create a new one
  d. Set the **sipRecorderUri** parameter
- Use the recordingMode parameter on the API object /callProfiles or /callProfiles/<call profile id> to select whether a meeting can be recorded or not. Options for this are:
  - **automatic** – recording occurs without any user intervention, if recording cannot occur the meeting still occurs.
  - **manual** – users can manually start and stop the recording using DTMF.
  - **disabled** – no users can record.
- Control which users have permission to start and stop recording by setting the recordingControlAllowed parameter on callLegProfiles.
- Use the startRecording and stopRecording parameters for /dtmfProfiles and /dtmfProfiles/<dtmf profile id> to map the DTMF tones for starting and stopping recording.

Note: The additional API objects are given in the Cisco Meeting Server API Reference guide.

15.6 Finding out recording status

To find out the recording status:

- Use the Web Admin interface of the Meeting Server, select **Configuration** > **API**
- From the list of API objects, tap the ► after **/callLegs**
- Click on the **object id** of an existing call leg

Perform a GET on **`callLegs/<call leg id>`** – the **`recording`** value in the **`status`** output found here indicates whether this callLeg is recording (**`true`**) or not (**`false`**).

15.7 Recording indicator for dual homed conferences

For dual homed conferences, recording should be done using the Microsoft recording method on the Lync/Skype endpoint. We do not recommend using Cisco Meeting Server to record dual homed conferences.

A recording icon indicates to SIP participants connected to the Meeting Server that a Lync/Skype endpoint is recording the conference on the Lync/Skype side.

Meeting Server adds a recording icon to the video pane composed for non-ActiveControl endpoints. Table 11 below shows the icons that Meeting Server will display to indicate that a dual homed conference is being recorded.

Table 11: Recording indicators

Icon displayed	Description
	Meeting is being recorded via the Meeting Server.
	Meeting is being recorded by a Lync/Skype endpoint
	Meeting is being recorded via the Meeting Server and by a Lync/Skype endpoint.
	The meeting is not being recorded (no icon displayed).

Note: web app shows the recording state using its own icons, they do not distinguish between local and remote recording. Meeting Server icons are not overlaid on the web app video pane.

15.8 Recording with Vbrick

Note: This section only applies to the Meeting Server internal Recorder component.

The Uploader component simplifies the work flow for uploading Meeting Server recordings to the video content manager, Vbrick, from a configured NFS connected to a Meeting Server. No manual importing of recordings is required.

Once the Uploader component is configured and enabled, recordings are pushed from the NFS to Vbrick, and an owner is assigned to the recording. The Rev portal applies security configured by your administrator to your video content, only allowing a user to access the content that they are permitted to access. Vbrick emails the owner when the recording is available in the owner's Rev portal. Owners of a recording access the video content through their Rev portal, and can edit and distribute as necessary.

Note: If a file is added to the NFS share within a space directory, the file will be uploaded to Vbrick as though it were a valid recording. Take care to apply permissions to your NFS share so that only the Recorder can write to it.

Note: Depending on the mechanism you use to store the recordings you may need to open external firewall ports so that the recorder, uploader and storage system can communicate. For example: NFS running version 2 or 3 of the port mapper protocol uses TCP or UDP ports 2049 and 111.

Note: Do not use the Firewall component on the Meeting Server if using either the Recorder or Uploader.

15.8.1 Prerequisites for the Meeting Server

Uploader installation. The Uploader component can be installed on the same server as the Recorder component, or on a separate server. If installed on the same server as the Recorder, then add a couple of vCPUs for it to use. If run on a different server, then use the same server specification as for the Recorder: dedicated VM with a minimum of 4 physical cores and 4GB of RAM.

CAUTION: The Uploader must run on a different Meeting Server to the Call Bridge hosting the conferences.

Read and Write access to the NFS share. The Meeting Server running the Uploader will require Read and Write permissions for the NFS. Write permission is required to allow the Uploader to re-write the name of the mp4 file when upload is completed.

Note: If the NFS is set or becomes Read Only, then the Uploader component will continuously upload the same video recording to Vbrick. This is a result of the Uploader being unable to mark the file as upload complete. To avoid this, ensure that the NFS provides read/write access.

API Access to Vbrick Rev. Configure API access for a user on Vbrick Rev.

API Access to Call Bridge. Configure API access for a user on the Meeting Server running the Call Bridge.

Trust Store Store the certificate chains from the Vbrick Rev server, and the Meeting Server running the Web Admin interface for the Call Bridge. The Uploader needs to trust both the Vbrick Rev and the Call Bridge.

Decide who has access to the video recordings. Access to uploaded video recordings can be set to: All Users, Private, and for only space owners and members.

Default state of video recordings. Decide whether the video recordings are immediately available after upload (Active), or that the owner of the video recording needs to publish it to make the recording available (Inactive).

Table 12: Port Requirements

Component	Connecting to	Destination port to open
Call Bridge	NFS (version 3)	2049
Uploader	Web Admin of Call Bridge	443 or port specified in Uploader configuration
Uploader	Vbrick Rev server	443 for video uploads and API access to Vbrick Rev server

15.8.2 Configuring the Meeting Server to work with Vbrick

These steps assume that you have already setup the NFS to store recordings.

1. Establish an SSH connection to the MMP of the Meeting Server where you want to run the Uploader. Log in.
2. For new Vbrick installations, ignore this step. If you are reconfiguring a Vbrick installation then first disable Vbrick access to the Meeting Server.
   `uploader disable`
3. Specify the NFS that the Uploader will monitor.
   `uploader nfs <hostname/IP>:<directory>`
4. Specify the Meeting Server that the Uploader will query for recording information, for example the name of the Meeting Server hosting the space associated with the recording.
   `uploader cms host <hostname>`
5. Specify the Web Admin port on the Meeting Server running the Call Bridge. If a port is not specified, it defaults to port 443.
   `uploader cms port <port>`
6. Specify the user with API access on the Meeting Server running the Call Bridge. The password is entered separately.
   `uploader cms user <username>`
7. Set the password for the user specified in step 6. Type
   `uploader cms password`
   you will be prompted for the password.

8. Create a certificate bundle (crt-bundle) holding a copy of the Root CA's certificate and all intermediate certificates in the chain for the Web Admin on the Meeting Server running the Call Bridge.
9. Add the certificate bundle created in step 8 to the Meeting Server trust store.
   uploader cms trust <crt-bundle>
10. Configure the Vbrick host and the port to which the Uploader will connect.
    uploader rev host <hostname>
    uploader rev port <port>

    Note: The port defaults to 443 unless otherwise specified.

11. Add a Vbrick Rev user who has API permission to upload video recordings.
    uploader rev user <username>
12. Set the password for the user specified in step 11. Type
    uploader rev password
    you will be prompted for the password.
13. Create a certificate bundle (crt-bundle) holding a copy of the Root CA's certificate and all intermediate certificates in the chain for the Vbrick Rev server.
14. Add the certificate bundle created in step 13 to the Vbrick Rev trust store.
    uploader rev trust <crt-bundle>
15. Set access to the video recording.
    uploader access <Private|Public|AllUsers>
16. Give members of the space the ability to view or edit the recordings.
    uploader cospace_member_access <view|edit|none>

    Note: This step requires the listed members to have valid email addresses which are associated with accounts on Vbrick. For example **user1@example.com**

17. Decide whether the owner of the space is the single owner of the video recordings.
    uploader recording_owned_by_cospace_owner <true|false>

    Note: This step also requires the owner of the video recordings to have a valid email address which is associated with an account on Vbrick.

18. If the owner of the space is not listed in Vbrick Rev, then set the username of the fallback owner. If the fallback owner is not specified, then owner will default to the user configured on the MMP.
    uploader fallback_owner <vbrick-user>
19. Enable comments to the video recordings.
    uploader comments enable
20. Enable ratings for the video recordings.
    uploader ratings enable

21. Set the download permission for the video recordings.
    `uploader downloads enable`
22. Set the default state of the video recording when first uploaded to Vbrick Rev.
    `uploader initial_state <active|inactive>`
23. Decide whether to delete the video recording from the NFS after upload is complete
    `uploader delete_after_upload <true|false>`
24. Enable the Uploader to access the Meeting Server
    `uploader enable`

Note: Set **`messageBoardEnabled`** to **`true`**to see the messages being posted in the space indicating that the recording is available.

15.9 Streaming meetings

The internal SIP Streamer component (from version 3.0) adds the capability of streaming meetings held in a space to the RTMP URL configured on the space.

An external streaming server needs to be configured to be listening on this RTMP URL. The external streaming server can then offer live streaming to users, or it can record the live stream for later playback.

Note: The Streamer component supports the RTMP standard in order to work with third party streaming servers that also support the RTMP standard. Vbrick is the officially supported external streaming server, however, other servers have also been tested.

Note: The Streamer component supports the RTMP standard in order to work with third party streaming servers that also support the RTMP standard. Vbrick is the officially supported external streaming server, however, other servers have also been tested.

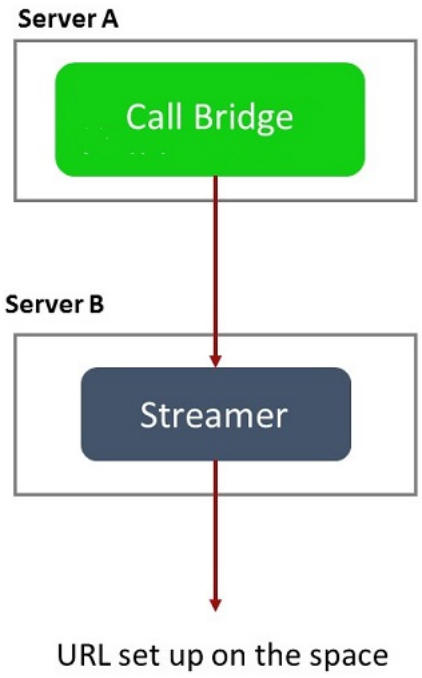
Note: You may need to open firewall ports if the streaming destination RTMP URLs are on the external side of a firewall.

Version 3.1 extends the RTMP support in the internal SIP streamer application to RTMPS — essentially RTMP over a TLS connection. Previously all traffic between the streamer and RTMP server was unencrypted, 3.1 RTMPS support allows this traffic to be encrypted.

The existing **`tls`** MMP command is extended to optionally allow configuration of TLS trusts for RTMPS. This step is optional but recommended. If a TLS trust is not configured then the RTMPS connection will not be secure.

The following figure shows the permitted streamer deployment. For information on scalable and resilient deployments, see Section 15.3.3

Figure 44: Permitted deployment for streaming: remote mode



For testing purposes only, the Streamer can be co-located on the same server as the Call Bridge. This may support between 1 to 2 simultaneous streamings.

15.10 Deploying the new SIP streamer component on a VM server

This is a two stage process:

- Configuring a Meeting Server streamer via the MMP
- Configuring the streamer URI via the API

Task 1: Configuring a Meeting Server streamer via the MMP

1. Upgrade to version 3.0.
2. SSH into the MMP and login to configure the recorder (enter the MMP command, **`streamer help`** to see a list of all available commands).
3. Configure the listening interface of the streamer and the SIP TCP and TLS ports to listen on using the MMP command **`streamer sip listen <interface> <tcp-port|none> <tls-port|none>`**. Set the respective port to **`none`** to disable the service:
    a. For example, if you want to only listen on the TLS port and not the TCP port, enter **`streamer sip listen a none 6000`**

b. Make a note of the ports you've configured if they're not the default TCP/TLS ports (5060/5061), as they will be needed later.

4. Optionally, you can set the maximum resolution that you want the streamer to do (or to only stream the audio of calls) using the MMP command **`streamer sip resolution <audio|720p|1080p>`**, if not specified, the default is 720p.
   a. For example, if you want to set it to 1080p, enter **`streamer sip resolution 1080p`**

   Note: If you want to use 1080p we recommend that you increase your transmit SIP call bandwidth to 3,500,000 bits per second to optimize the video quality. To do this, on the Web Admin UI go to **Configuration > Call settings > Bandwidth settings (SIP)** and set as required.

5. Optionally, if TLS is configured, configure the SIP TLS certificates you would like to use:
   a. Enter the MMP command **`streamer sip certs <key-file> <crt-file> [<crt-bundle>]`**

   Note: Note that if SIP TLS certificates are not configured with this option, the SIP TLS service will fail to start.

6. Optionally, if TLS is configured, you can perform TLS verification for SIP (or LDAP or RTMPS) on the streamer as follows, for example:
   a. Enter the MMP command **`tls sip trust [<crt-bundle>]`**
   b. Enter the MMP command **`tls sip verify enable`**

   Note: For the TLS connection to be secure we recommend enabling TLS verification.

7. Check the configuration is correct – enter the MMP command **`streamer`** to view the configuration.
8. Enter the MMP command **`streamer enable`** to enable the streamer service.

Task 2: Configuring the streamer URI via the API

Once the new SIP streamer is enabled, it can be configured and used in the Call Bridge using the **`sipStreamerUri`** API parameter specified in the API call profile object.

If you wish, you can also configure a custom URI that maps to an outboundDialPlan rule (the domain can be anything of your choice, e.g. " streaming.com"). You will need to configure an outboundDialPlan rule which tells Meeting Server how to route the domain used in **`sipStreamerUri`** to the streamer. This will allow you to control priority values, encryption, etc. For more information on configuring **`/outboundDialPlanRules`**, see the " Dial plan configuration - overview" chapter of your deployment guide.

Note: The user part of the configured URI (i.e. the part before the '@' symbol) has no special meaning, and for the new internal SIP streamer component, although required, it can usually be anything, e.g. " streaming@streamer.com" . The important part of the URI is the domain part.

To configure the **`sipStreamerUri`** parameter using the Meeting Server Web Admin interface:

1. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
2. From the list of API objects, tap the ► after **/api/v1/callProfiles**
3. To configure or modify an existing call profile, select the object id of the required callProfile and fill in the **sipStreamerUri** field with your chosen URI.

   Note: When using the new SIP streamer you only need to use one SIP URI, e.g streaming@streamer.com, you don't need to have different SIP URIs on different profiles.
4. If you haven't done so already, set the **`streamingMode`** parameter to either, **`manual`** or **`automatic`** (depending on how you want meetings to be streamed).
5. Click **Modify**.

The updated callProfile can then be assigned to coSpaces, tenants or the top level (global) profile, as required. In this example an updated callProfile is assigned at the global level as follows:

1. Using the Web Admin interface, select **Configuration > API**:
   a. From the list of API objects tap the ► after **/api/v1/system/profiles**
   b. Click **View or edit**
   c. Scroll down the parameters to **callProfile** and click **Choose**.
   d. From the resulting " callProfile object selector window" , click **Select** for the **object id** of the **callProfile** you wish to assign to the top level global profile.
   e. Click **Modify**.
   f. The newly assigned callProfile object id should now be listed under **Object configuration**.

For each coSpace in the API that you wish to enable streaming for, you must configure the **`streamUrl`** coSpace API field with the RTMPS stream URL to stream to (e.g. " rtmps://mystream.com/live/app"). To configure this:

1. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
2. From the list of API objects, tap the ► after **/api/v1/coSpaces**
3. To configure or modify an existing coSpace, select the object id of the required coSpace

and fill in the **streamUrl** field with the RTMPS stream URL to stream to.

4. Click **Modify**.

15.10.1 Known Limitations

CAUTION: Be warned that the stream URL is sent via SIP headers, so any RTMP stream URLs containing login credentials could potentially be exposed to call control providers which may log them.

16 LDAP configuration

If you plan for users to utilize the web apps to connect to the Meeting Server, then you must have an LDAP server (currently Microsoft Active Directory, OpenLDAP or Oracle Internet Directory LDAP3, see note below). The Meeting Server imports the User accounts from the LDAP server.

You can create user names by importing fields from LDAP, as described in this section. The passwords are not cached on the Meeting Server, a call is made to the LDAP server when a web app authenticates, and therefore passwords are managed centrally and securely on the LDAP server.

Note: When configuring the Meeting Server for LDAP/AD sync, the fields which accept LDAP/AD attributes require that attributes are entered in their case-sensitive format. For example, if the username mapping uses the attribute userPrincipalName then $userPrincipalName$ can result in successful sync but $UserPrincipalName$ will result in sync failure. You are advised to check that each LDAP attribute is entered in the correct case.

Note: From version 2.1, the Meeting Server supports Oracle Internet Directory (LDAP version 3). This must be configured through the API, not the Web Admin interface. To configure the Meeting Server to support Oracle Internet Directory, the Meeting Server should not use the LDAP paged results control in search operations during LDAP sync. POST to **`/ldapServers`** or PUT to **`/ldapServers/<ldap server id>`** the request parameter **`usePagedResults`**set to false.

16.1 Why use LDAP?

Using LDAP to configure the Meeting Server is a powerful and scalable way to set up your environment: defining your organization's calling requirements within the LDAP structure minimizes the amount of configuration required on the Meeting Server.

The server uses the concept of filters, rules and templates, which allow you to separate users into groups, for example:

- Everyone in the HR department
- Staff at grade 11 and above
- Job title = 'director'
- People whose surname starts with 'B'

16.2 Meeting Server settings

The examples in this section explain how to configure a single LDAP server (in this case Active Directory), using the Web Admin interface on the Meeting Server. However, the Meeting Server supports multiple LDAP servers which can be configured via the API, see the LDAP Methods section in the API Reference guide.

When configuring a cluster of Call Bridges, the simplest method is to use the API. If configuring multiple Call Bridges via the Web Admin interface, each must have identical configuration.

Note: The Web Admin Interface only allows you to configure one LDAP server.

To set up the Meeting Server to work with Active Directory, follow these steps:

1. Sign in to the Web Admin Interface and go to **Configuration > Active Directory**.
2. Configure the connection to the LDAP server in the first section with the following:
   - Address = this is the hostname or IP address of your LDAP server
   - Port = usually 636
   - Username = the Distinguished Name (DN) of a registered user. You may want to create a user specifically for this purpose.
   - Password = the password for the user name you are using
   - Secure Connection = tick this box for a secure connection

     For example:

     ```
     Address:  ldap.example.com
     Port:     636
     Username: cn=Fred Bloggs,cn=Users,OU=Sales,dc=YourCompany,dc=com
     Password: password
     ```

     Note: For further details of the permissions required by the user name and password credentials, see Section 16.4.

     Note: The Meeting Server supports secure LDAP. By default the LDAP server runs on port 636 for secure communications and port 389 for insecure communications. The Meeting Server supports both, but we recommend using 636. Note that you must select Secure Connection (see above) for communications to be secure: using port 636 alone is not enough.

     Note: When LDAP servers are configured with secure connection, connections are not fully secure until TLS certificate verification has been configured using the `tls ldap` command on the MMP.

     ---

3. Type the Import Settings which will be used to control which users will be imported.

   - Base Distinguished Name = the node in the LDAP tree from which to import users. The following is a sensible choice for base DN to import users

     ```
     cn=Users,dc=sales,dc=YourCompany,dc=com
     ```

   - Filter = a filter expression that must be satisfied by the attribute values in a user's LDAP record. The syntax for the Filter field is described in rfc4515.

     A rule for importing people into the main database might reasonably be 'import anyone with an email address', and this is expressed by the following filter:

     ```
     mail=*
     ```

     For testing purposes you may want to import a named user (e.g. fred.bloggs)and a group of test users whose mail address starts with “test”; for example:

     ```
     (|(mail=fred.bloggs*)(mail=test*))
     ```

     If you wanted to import everyone apart from one named user (e..g. fred.bloggs), use this format:

     ```
     (!(mail=fred.bloggs*))
     ```

     To import users that belong to a specific group, you can filter on the memberOf attribute. For example:

     ```
     memberOf=cn=apac,cn=Users,dc=Example,dc=com
     ```

     This imports both groups and people that are members of the APAC group.

     To restrict to people (and omit groups), use:

     ```
     (&(memberOf=cn=apac,cn=Users,dc=Example,dc=com)(objectClass=person))
     ```

     Using an extensible matching rule (LDAP_MATCHING_RULE_IN_CHAIN / 1.2.840.113556.1.4.1941), it is possible to filter on membership of any group in a membership hierarchy (below the specified group); for example:

     ```
     (&(memberOf:1.2.840.113556.1.4.1941:=cn=apac,cn=Users,dc=Example,
     dc=com)(objectClass=person))
     ```

     Other good examples which you can adapt to your LDAP setup include:

     Filter that adds all Person and User except the ones defined with a !

     ```
     (&(objectCategory=person)(objectClass=user)(!(cn=Administrator))(!
     (cn=Guest))(!(cn=krbtgt)))
     ```

     Filter that adds same as above (minus krbtgt user) and only adds if they have a sAMAccountName

```
(&(objectCategory=person)(objectClass=user)(!(cn=Administrator))(!
(cn=Guest))(sAMAccountName=*))
```

Filter that adds same as above (Including krbtgt user) and only adds if they have a sAMAccountName

```
(&(objectCategory=person)(objectClass=user)(!(cn=Administrator))(!
(cn=Guest))(!(cn=krbtgt))(sAMAccountName=*))
```

This filter only imports specified users within (|(tree

```
(&(objectCategory=person)(objectClass=user)(|(cn=accountname)
(cn=anotheraccountname)))
```

Global Catalog query to import only members of specified security group (signified with =cn=xxxxx

```
(&(memberOf:1.2.840.113556.1.4.1941:=cn=groupname,cn=Users,
dc=example,dc=com)(objectClass=person))
```

4. Set up the Field Mapping Expressions

   The field mapping expressions control how the field values in the Meeting Server's user records are constructed from those in the corresponding LDAP records. Currently, the following fields are populated in this way:

   - Display Name
   - User name
   - space Name
   - space URI user part (i.e. the URI minus the domain name)
   - space Secondary URI user part (optional alternate URI for space)
   - space call id (unique ID for space for use by WebRTC client guest calls)

   Field mapping expressions can contain a mixture of literal text and LDAP field values, as follows:

   ```
   $<LDAP field name>$
   ```

   As an example, the expression

   ```
   $sAMAccountName$@example.com
   ```

   Generates:

   ```
   fred@example.com
   ```

For more information see " More information on LDAP field mappings" on page 184.

Note: Each imported user must have a unique user ID (JID), constructed using the JID field in the Field Mapping Expressions section of the **Configuration > Active Directory**. In order to construct a valid JID, any LDAP attribute used in the JID field mapping expression must be present in each LDAP record that is to be imported. To ensure that only records that have these attributes present are imported, we recommend that you include presence filters (i.e. those of the form (<attribute name>=*)) using a '&' (AND) in the Filter field under Import Settings for each attribute used in the JID field mapping expression.

For example, suppose your JID field mapping expression is `$sAMAccountName$@company.com`, and you wish to import users who are members of the group cn=Sales,cn=Users,dc=company,dc=com, an appropriate import filter would be:

```
(&(memberOf=cn=Sales,cn=Users,dc=company,dc=com)(sAMAccountName=*))
```

5. To synchronize with Active Directory, select **Sync now** or activate the synchronization by using the appropriate API call (see the Cisco Meeting Server API Reference Guide).

   Note: that you must manually resynchronize whenever entries in the LDAP server change.

6. View the result of the synchronization by going to **Status > Users**.

It is possible to choose whether to use OU separation when importing from the LDAP server. In the Web Admin Interface, go to **Configuration > Active Directory** and in the **Corporate Directory Settings** section select **Restrict Search to Searcher OU** to enable the search only within the OU of the user account.

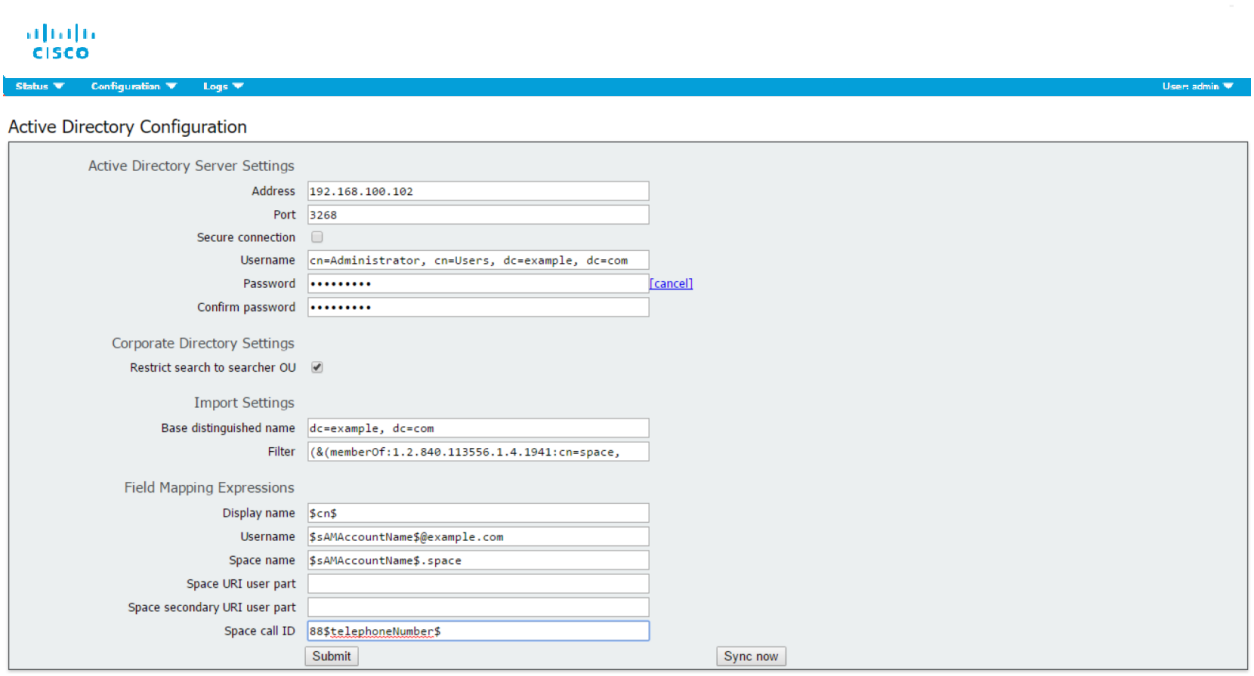
16.3 Example

This example assigns a space to a particular group of users and a Call ID for this space using an 88 prefix in front of the regular telephone number.

1. Create the group in the LDAP structure called "space" and assign the required members to that group.

2. Use the following filter which uses the extensible matching rule (LDAP_MATCHING_RULE_IN_CHAIN / 1.2.840.113556.1.4.1941) to find all the users that are a member of the "space" group:

```
(&(memberOf:1.2.840.113556.1.4.1941:=cn=space,cn=Users,dc=lync,
dc=example,dc=com)(objectClass=person))
```



3. Then synchronizing a particular user in the directory called:

   ```
   cn = Fred Blogs
   TelePhoneNumber = 7655
   sAMAccountName = fred.blogs
   ```

   creates the following space which can be viewed on the **Status > Users** page.

Name	Username
Fred Blogs	`fred.blogs@example.com`

   And the following space that can be viewed on the **Configuration > space** page.

Name	URI user part
fred.blogs	`fred.blogs.space`

16.4 More information on LDAP field mappings

This section provides additional information for LDAP field mappings that you set up for the Meeting Server.

Parts of an LDAP field value can be substituted by means of a sed-like construction, as follows:

```
$<LDAP field name>|'/<regex>/<replacement format>/<option>'$
```

where:

<option> can be g, to replace every match of **<regex>** with **<replacement format>**, or blank to match only the first

parts of **<regex>** can be tagged for use in **<replacement format>** by enclosing them in round brackets

tagged matches can be referenced in **<replacement format>** as \x where x is a digit from 0 to 9. Match 0 corresponds to the entire match, and matches 1-9 the 1st to 9th tagged sub-expressions

single quotes inside the substitution expression must be escaped with a backslash, as must backslash characters themselves

any character other than a single quote, a backslash, or the digits 0-9 can be used in place of the forward slash that separates the components of the substitution expression

if the separating character is to be used as a literal within the expression, it must be escaped with a backslash.

As an example, the following would convert addresses in the format:

```
firstname.lastname@test.example.com
```

into the format:

```
firstname.lastname@example.com JIDs
$mail|'/@test/@xmpp/'$
```

and the following would remove every lower case 'a' from the user's full name:

```
$cn|'/a//g'$
```

A sensible set of expressions for use might be:

```
Full name:              $cn$
JID:                    $mail|'/@test/'$
space URI:            $mail|'/@.*//'$.space
space dial-in number: $ipPhone$
```

Note: The LDAP server credentials are used to read the following fields (for security reasons you may want to restrict the fields and permissions available using those credentials):

- mail
- objectGUID
- entryUUID
- nsuniqueid
- telephoneNumber
- mobile

- sn
- givenName

16.5 Enforcing passcode protection for non-member access to all user spaces

When spaces are auto-generated via an LDAP sync, they are all created without a passcode. By default **nonMemberAccess** is set to **true** so that the existing behavior remains unchanged, no passcode is required to access the space and non-members are able to access the created spaces.

Setting **nonMemberAccess** to **false** allows a company to enforce passcode protection for non-member access to all user spaces.

To ensure the member must configure non-member access and set a passcode as part of the LDAP sync:

- Either POST to **/ldapSources** or PUT to **/ldapSources/<ldap source id>** the request parameter **nonMemberAccess** set to **false**.
- To retrieve the **nonMemberAccess** setting, use GET on **/ldapSources/<ldap source id>**.

Note: Spaces created before version 2.4 (when this parameter was introduced) are unaffected by any LDAP syncs.

17 Single Sign On (SSO) for Cisco Meeting Server web app

This feature allows a web app user to login using an SSO provider to verify their identity.

SSO means the web app user doesn't need to enter their password every time they sign in as they can now have a single session with an identity provider (the entity responsible for authenticating users at a single place and maintaining a single session for each, for example, OAuth, gmail).

It allows the web app user to login with different SSO providers on the same Web Bridge.

This SSO mechanism uses SAML (Security Assertion Markup Language) 2.0 which is an open standard and a widely used industry standard protocol.

Note: Currently Meeting Server supports only HTTP-POST bindings in the SAML 2.0 protocol. This means it will only accept messages on its HTTP-POST AssertionConsumerService and it will reject Identity Providers with no HTTP-POST bindings available

---

Note: If you enable SSO login, you can no longer use LDAP login.

17.1 Configuring SSO for use on Meeting Server web app

To use SSO requires some configuration for the identity provider and on the Meeting Server (regarded as the Service Provider in the SAML 2.0 exchange) as detailed below.

Task 1: Mapping between Identity provider and Meeting Server users

So that Meeting Server can correctly map users on your Identity provider to its own users you will need to setup an authenticationId for every user authenticated via SSO. This can be done as part of the standard ldap sync process. The contents of this field will be verified against a custom parameter passed from the Identity provider with successful responses (see Task 2).

We recommend that you choose a unique identifier for each user (e.g. $sAMAccountName$). Empty values for the authenticationIds are not accepted.

To setup the authenticationId as part of an ldapSync you can create a new ldapSync or modify an existing one.

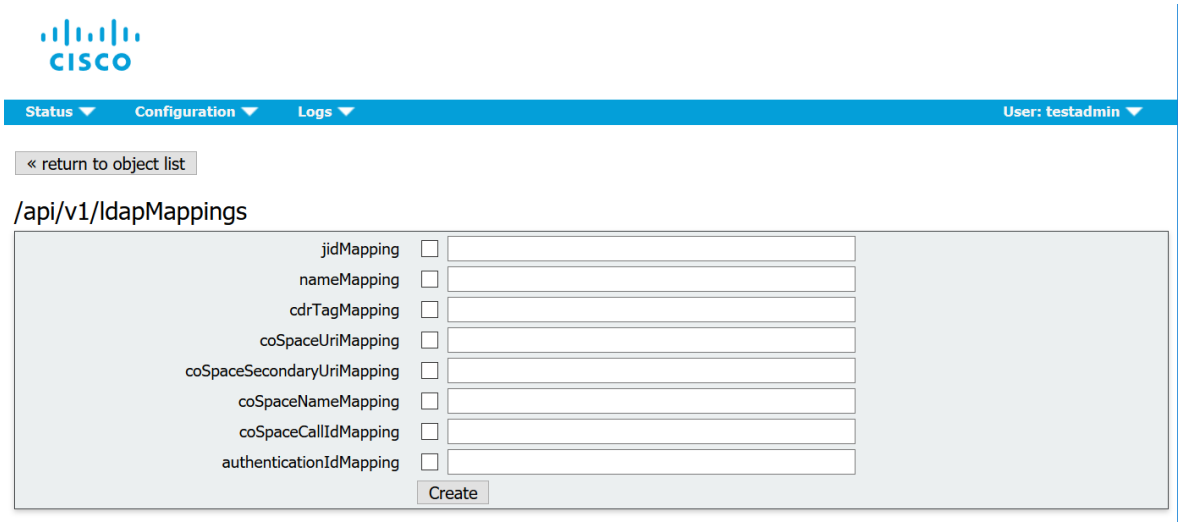
You then need to create/modify an ldapMapping and populate the **authenticationIdMapping** parameter with an appropriate value (e.g. $sAMAccountName$).

Using the Meeting Server Web Admin interface:

a. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:

b. From the list of API objects, tap the ▶ after **/api/v1/ldapMappings**

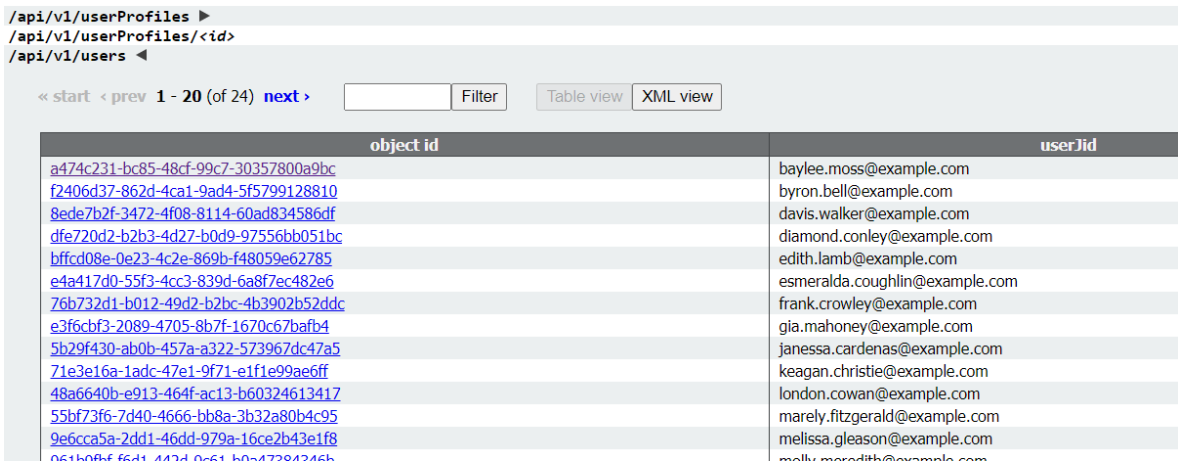
c. Click **Create new** or select the ID for an existing ldap mapping to modify.



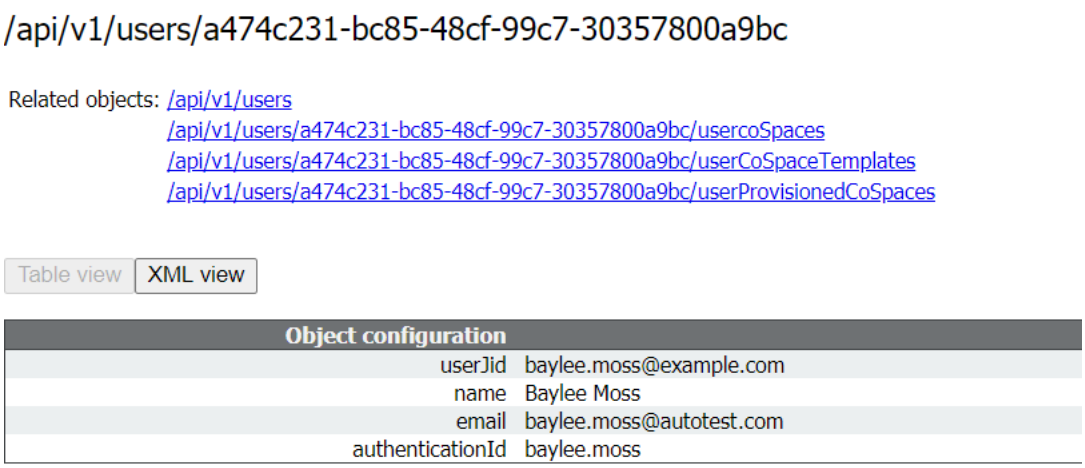
d. Populate the **authenticationIdMapping** parameter with an appropriate value (e.g. $sAMAccountName$) and click **Create** or **Modify**, as appropriate.

e. For the changes to take effect on the Meeting Server you now need to trigger an ldapSync. From the list of API objects, tap the ▶ after **/api/v1/ldapSyncs** and select the object ID or **Create new**, as appropriate. Once the ldapSync has finished you can verify that the process has succeeded by examining one of your Meeting Server users.

f. Firstly, from the list of API objects, tap the ▶ after **/api/v1/users**, to display a list of users as seen in this example:



g. Select one of the users that should now have authenticationId set up (you may need to use the Filter field). The user entry should now include an **authenticationId** field with the correct value from the ldapSync as shown in this example:



Task 2: Identity Provider configuration

1. All identity providers let you upload a metadata xml file representing the Service Provider being registered with them (i.e. the Meeting Server in this instance). Some identity providers simplify the process by allowing you to configure the most important pieces of information. Metadata xml file examples can be found here.

   The values to include in the metadata xml file to be uploaded to the identity provider are:

   a. entityID – this is the Web Bridge 3 address (i.e. https://<domain>:port). This address must be a valid Web Bridge 3 address reachable from the browsers of web app users.

      Note: If there are multiple Web Bridge 3s in a deployment this should be a load-balanced address.

   b. An HTTP-POST AssertionConsumerService for the Web Bridge address defined as the entityId following the format " https://<domain>:<port>/api/auth/sso/idpResponse" .

   c. Optional. A public key for signing with which the identity provider will verify AuthnRequest signatures.

   d. Optional. A public key for encryption with which the identity provider will encrypt information sent back to one of the Web Bridge 3s routable through the address provided above.

Note: Meeting Server requires that messages sent to it are signed by the identity provider on the Response and/or Assertion level. Unsigned communication will be discarded.

2. You need to configure a custom parameter passed from your identity provider with a successful response. For each user its contents should match the value already configured as authenticationId for that Meeting Server user (e.g. $sAMAccountName$). Usually identity providers will have a special form or dialog for that as part of creating the Service Provider entry. This parameter can be any name of your choosing, although we recommend you choose something easy to remember, such as " uid" (you will need the name in Task 3).

Task 3: Creating SSO archive zip file

1. To configure the Meeting Server, you need to create an archive zip file named sso_<name>.zip for each SSO you want to configure for the Web Bridge 3 on that Meeting Server. The file name must start with " sso_" followed by a meaningful name of your choice.

   Create a zip archive file containing these files:

   a. idp_config.xml – This is a file that the administrator will receive from the identity provider.

   b. config.json – includes:

      - supportedDomains (array of strings) – a list of all domains for Meeting Server users which will be authenticated against this identity provider. I.e. using the examples from Task 1, supportedDomains would contain the single entry of " example.com" .
      - authenticationIdMapping (string) – name of the parameter from the identity provider responses configured as part of Task 2 (e.g. " uid") that matches to the authenticationIds in the Meeting Server. Web app users for SSO must have authenticationIds setup for them(see Task 1.)
      - ssoServiceProviderAddress (string) – the address on which the identity providers will send the responses, this will match the Web Bridge 3 specified in the entityID in Task 2.

   c. Optional. sso_sign.key – private key for the public signing key configured on the identity provider side. It will be used to sign outgoing AuthnRequests from Meeting Server which can then be verified using the public key on the identity provider's side.

   d. Optional. sso_encrypt.key – private key for the public encryption key configured on the identity provider side. It will be used to decrypt on the Meeting Server messages encrypted with the public key on the identity provider's side.

Note: You will need different named zip files for different identity providers.

2. Create an archive (zip) file containing the SSO files.

Note: When you zip the files, do not zip the folder containing the SSO files. If this is done, this will create an extra layer of folder (zipped file > folder > SSO files). Instead, highlight the SSO files and right-click to zip them (or open a zip application and zip the files together). This will create a zipped file with the SSO files without creating an extra layer of folder (e.g. zipped file > SSO files).

Task 4: Uploading the SSO archive zip

The SSO archive zip now needs to be uploaded and hosted on the local Web Bridge 3.

Note: The commands in the following steps are for console/terminal environments (i.e. command prompt or terminal) and not for SFTP clients such as WinSCP.

1. For each Meeting Server with an enabled Web Bridge 3 which will locally host this zip archive:
2. a. Connect your SFTP client to the IP address of the MMP.
   b. Log in using the credentials of the MMP admin user.
   c. Upload the zip file **`sso_<name>.zip`**. For example:
      ```
      PUT sso_<name>.zip
      ```
   d. Connect your SSH client to the IP address of the MMP.
   e. Log in using the credentials of the MMP admin user.
   f. Restart the Web Bridge 3
      ```
      webbridge3 restart
      ```
3. The new SSO archive file will be picked up after the restart.

Note: Once a web app user is logged in they will have a separate session on the web app application from the one with the identity provider. This means that if they logout/sign out from the web app application but not from the identity provider once they enter the same username they will automatically be allowed into the web app application again. However, if they sign out from the identity provider it doesn't sign them out from the web app application and they will have to also sign out from the web app application. To ensure that you cannot log in for this browser session again you must sign out from both the web app application and the identity provider.

17.1.1 Example 1 config.json file

This is an example config.json file:

```
{
    "authenticationIdMapping" : "<parameter_from_task_2>",
    "ssoServiceProviderAddress" : "https://<domain>:<port>",
    "supportedDomains" : ["<domain1>","<domain2>"]
}
```

17.1.2 Example 2 Simple service provider metadata file.

This is an example simple service provider metadata file – note that administrators will have to modify <domain> and <port> with their relevant values.

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="https://<domain>:<port>" entityID="https://<domain>:<port>">
  <md:SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</md:NameIDFormat>
    <md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://<domain>:<port>/api/auth/sso/idpResponse" index="0"/>
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```

17.1.3 Example 3 Comprehensive service provider metadata file.

This is a comprehensive metadata file example which includes an xml for the signing and encryption keys.

Note: The keys should be placed in the X509Certificate sub-elements of their corresponding KeyDescriptor elements according to the use parameter (" encryption" or " signing"). You must substitute " ..." with the text contents of the key (e.g. ds:X509CertificateMIID**<omitted_key_text>**+gb</ds:X509Certificate>)

Note: If you include a signing certificate, the value AuthnRequestsSigned is set to " true" (it is set to " false" in the simpler metadata file in example 2).

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="https://<domain>:<port>" entityID="https://<domain>:<port>">
  <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:X509Data>
        <ds:X509Certificate>...</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
```

```
          <ds:X509Certificate>...</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</md:NameIDFormat>
    <md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://<domain>:<port>/api/auth/sso/idpResponse" index="0"/>
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```

18 Support for ActiveControl

The Meeting Server supports ActiveControl for hosted calls. For participants using a Cisco SX, MX or DX endpoint with CE 8.3+ software installed, ActiveControl allows the meeting participant to receive details of the meeting and perform a few administrative tasks during the meeting, using the endpoint interface.

18.1 ActiveControl on the Meeting Server

The Meeting Server supports sending the following meeting information to ActiveControl enabled endpoints:

- Participant list (also known as the roster list) so that you can see the names of the other people in the call and the total number of participants,
- indicator of audio activity for the currently speaking participant,
- indicator of which participant is currently presenting,
- Indicators telling whether the meeting is being recorded or streamed, and if there are any non-secure endpoints in the call,
- on screen message which will be displayed to all participants,

and supports these administrative tasks on ActiveControl enabled endpoints:

- select the layout to be used for the endpoint,
- disconnect other participants in the meeting.

18.2 Limitations

- If an ActiveControl enabled call traverses a Unified CM trunk with a Unified CM version lower than 9.1(2), the call may fail. ActiveControl should not be enabled on older Unified CM trunks (Unified CM 8.x or earlier).
- ActiveControl is a SIP only feature. H.323 interworking scenarios are not supported.

18.3 Overview on ActiveControl and the iX protocol

ActiveControl uses the iX protocol, which is advertised as an application line in the SIP Session Description Protocol (SDP). The Meeting Server automatically supports ActiveControl, but the feature can be disabled, see section Section 18.4 . In situations where the far end network is not known or is known to have devices that do not support the iX protocol, it may be safest to disable iX on SIP trunks between the Meeting Server and the other Call control or Video Conferencing devices. For instance:

- for connections to Unified CM 8.x or earlier systems the older Unified CM systems will reject calls from ActiveControl-enabled devices. To avoid these calls failing, leave iX disabled on any trunk towards the Unified CM 8.x device in the network. In cases where the 8.x device is reached via a SIP proxy, ensure that iX is disabled on the trunk towards that proxy.
- for connections to third-party networks. In these cases there is no way to know how the third-party network will handle calls from ActiveControl-enabled devices, the handling mechanism may reject them. To avoid such calls failing, leave iX disabled on all trunks to third-party networks.
- for Cisco VCS-centric deployments which connect to external networks or connect internally to older Unified CM versions. From Cisco VCS X8.1, you can turn on a zone filter to disable iX for INVITE requests sent to external networks or older Unified CM systems. (By default, the filter is off.)

18.4 Disabling UDT within SIP calls

ActiveControl uses the UDT transport protocol for certain features, for example sending roster lists to endpoints, allowing users to disconnect other participants while in a call, and inter-deployment participation lists. UDT is enabled by default. You can disable UDT for diagnostic purposes, for example if your call control does not use UDT, and you believe this is the reason the call control does not receive calls from the Meeting Server.

Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:

1. From the list of API objects, tap the ► after **/compatibilityProfiles**
2. Either click on the **object id** of an existing compatibility profile or create a new one
3. Set parameter **sipUDT** = **false**. Click **Modify**.
4. From the list of API objects, tap the ► after **/system/profiles**
5. Click the **View or edit** button
6. Click **Choose** to the right of parameter **compatiilityProfile**. Select the **object id** of the compatibilityProfile created in step 3 above
7. Click **Modify**.

18.5 Enabling iX support in Cisco Unified Communications Manager

Support for the iX protocol is disabled by default on the Cisco Unified Communications Manager for some SIP profiles. To enable iX support in Unified CM, you must first configure support in the SIP profile and then apply that SIP profile to the SIP trunk.

Configuring iX support in a SIP profile

1. Choose **Device > Device Settings > SIP Profile**. The Find and List SIP Profiles window displays.
2. Do one of the following:
   a. To add a new SIP profile, click **Add New**.
   b. To modify an existing SIP profile, enter the search criteria and click **Find**. Click the name of the SIP profile that you want to update.

   The SIP Profile Configuration window displays.
3. Check the check box for **Allow iX Application Media**
4. Make any additional configuration changes.
5. Click **Save**

Applying the SIP profile to a SIP trunk

1. Choose **Device > Trunk**.

   The Find and List Trunks window displays.
2. Do one of the following:
   a. To add a new trunk, click **Add New**.
   b. To modify a trunk, enter the search criteria and click **Find**. Click the name of the trunk that you want to update.

   The Trunk Configuration window displays.
3. From the SIP Profile drop-down list, choose the appropriate SIP profile.
4. Click **Save**.
5. To update an existing trunk, click **Apply Config** to apply the new settings.

18.6 Filtering iX in Cisco VCS

To configure the Cisco VCS to filter out the iX application line for a neighbor zone that does not support the protocol, the zone must be configured with a custom zone profile that has the SIP UDP/IX filter mode advanced configuration option set to On.

To update advanced zone profile option settings:

1. Create a new neighbor zone or select an existing zone (**Configuration > Zones > Zones**).
2. In the Advanced parameters section, for **Zone profile**, choose *Custom* if it is not already selected. The zone profile advanced configuration options display.

3. From the **SIP UDP/IX filter mode**drop-down list, choose **On**.
4. Click **Save**.

18.7 iX troubleshooting

Table 13: Call handling summary for calls that contain an iX header

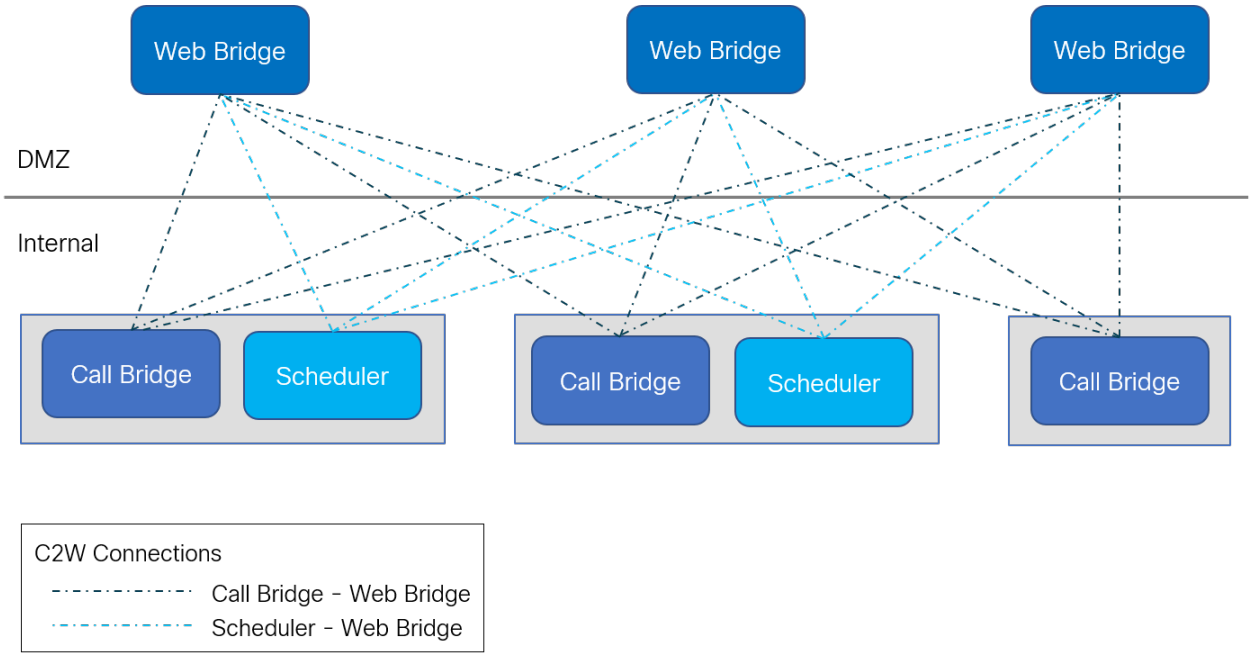
Scenario	Outcome
Unified CM 8.x or earlier	Calls fail
Unified CM 9.x earlier than 9.1(2)	Calls handled normally but no ActiveControl
Unified CM 9.1(2)	Calls handled normally plus ActiveControl
Endpoint - no support for iX and no SDP implementation	Endpoint may reboot or calls may fail

19 Scheduler - Deployment

The scheduler is deployed as a new component using the Meeting Server MMP. When the scheduler is enabled, it makes API requests to the Call Bridge over the loopback interface. It is therefore a requirement that the scheduler is deployed on a Meeting Server which is also hosting a Call Bridge. It is not possible to configure the scheduler to use a remote Call Bridge.

The list of configured Web Bridges is retrieved by the scheduler using the Call Bridge APIs. Persistent C2W connections are established to each Web Bridge similar to how the Call Bridge also establishes a C2W connection to each Web Bridge. No explicit configuration is required to enable connection between the scheduler and Call Bridge, because this happens automatically over the loopback interface. Similarly, the C2W connections are all automatic but it is necessary to configure a trust bundle between the scheduler and Web Bridges.

Note: The scheduler will need to be able to establish a C2W connection to all Web Bridges in a cluster. For call bridge deployments with large databases, disable the scheduler before rebooting the server and re-enable it only after the database sync completes, which can take up to 30 minutes.



It is not necessary to deploy a scheduler alongside every Call Bridge. A scheduler on a Meeting Server 1000/ Small and Meeting Server on VM deployments supports 150, 000 meetings and a scheduler on Meeting server 2000 supports 200,000 meetings. Two or three schedulers can be added to provide resiliency. Scheduled meeting data is stored in the Meeting Server database and both clustered and single box database deployments are supported.

The Call Bridge may log API requests from the scheduler as user " scheduler" . This is for logging purposes only and not a real account. There is no built in account and the scheduler user does not need to explicitly create an account. The scheduler uses the Call Bridge API over the loopback interface and is automatically a trusted source to issue API commands.

19.1 Deploying the Scheduler

To enable connection between the scheduler and Call Bridge, no explicit configuration is needed. This happens automatically over the loopback interface. Similarly, the C2W connections are all automatic, but it is necessary for a trust bundle to be configured between the scheduler and Web Bridges.

1. Configure C2W Trust.

   C2W is a TLS-based WebSocket connection established from the scheduler to each Web Bridge. Each scheduler needs to be able to connect to each Web Bridge in a cluster. The scheduler requires configuration of a client certificate and key to be used for this connection. To do this, create a certificate and upload it to the Meeting Server via SFTP or use the **pki** MMP commands to create a certificate.

   Configure the scheduler to use the certificate:

   scheduler c2w certs <key-file> <crt-fullchain-file>

   For example:

   scheduler c2w certs scheduler_c2w.key scheduler.cer

   It is necessary for the scheduler to be able to trust each Web Bridge it connects to. Upload a trust bundle which contains each Web Bridge certificate, via SFTP.

   Configure the scheduler using the command:

   scheduler c2w trust webbridge_bundle.cer

   It is also necessary for the Web Bridge to be able to trust the scheduler. So it is important to include the scheduler certificate in the bundle configured using the command:

   webbridge3 c2w trust <crt-bundle>

   All the necessary certs for both schedulers and Call Bridges should be included in the <crt-bundle>.

2. (Optional) Configure scheduler's HTTPS interface.

   The scheduler has its own HTTPS interface which if enabled, can be used to configure scheduler meetings using the scheduler APIs. The Web Bridge however, does not communicate with the scheduler using the management API. Though it is not mandatory to enable the HTTPS server, it is recommended that you do so because it provides some diagnostic and troubleshooting functionality.

   Configure the HTTPS server listen interface using the command:

```
scheduler https listen <interface> <port>
```

For example:

```
scheduler https listen a 8443
```

Configure a certificate key pair for the server using the command:

```
scheduler https certs <key-file> <crt-fullchain-file>
```

For example:

```
scheduler https certs scheduler_https.key scheduler_https.cer
```

3. (Optional) Configure the email server.

   Scheduler supports sending the email notifications via configuration of an SMTP email server. Email notifications are sent to the participants when a meeting is scheduled, canceled, or modified.

   The configuration of the server address and port, enabling email protocol, and configuring a username for authentication are specified via the following scheduler MMP commands:

   ```
   scheduler email server <hostname|address> <port>
   scheduler email server none
   scheduler email username <smtp username>
   scheduler email protocol <smtp|smtps>
   scheduler email auth <enable|disable>
   scheduler email starttls <enable|disable>
   ```

   Email will not be configured on a scheduler if no server address is configured on it. At least one email server must be configured for the scheduler to send email invites. Emails can be sent from any scheduler and not necessarily from the scheduler which was used to schedule the meeting. If an email server is down, then a different scheduler sends the email.

   If the Meeting Server does not receive a response from the Exchange email server within 5000 milliseconds (5 seconds), the Scheduler will automatically reject the request.

   Scheduler supports the following types of email configurations:

   a. SMTP
   b. SMTP with Authenticated Login (Auth Login)
   c. SMTP and STARTTLS
   d. SMTP with Auth Login and STARTTLS
   e. SMTPS (end to end TLS Encryption for the entire SMTP transaction)
   f. SMTPS with Auth Login

Note: It is recommended to use Exchange Server 2016 CU22 - 15.1.2375.7 and Exchange Server 2019 CU11 - 15.2.986.5.

From version 3.4 meeting invites can be sent to all the participants from a common email address. The MMP command **`scheduler email common-address <address@mail.domain> "<Display name>"`** configures the common email address and a display name on the Meeting Server. The Scheduler sends the meeting invites from the common email address to the participants.

If the common email address is left blank, the Scheduler sends the email invites from the organizer's email address.

Note: If common email address is not configured, authentication with the SMTP server requires an email address to be configured using the MMP command **`scheduler email username <smtp user-name>`**. This account configured on the MMP must have appropriate permissions to be able to send emails on behalf of web app users.

The organizer's name can also be included to appear as display name besides the email address to identify the sender. When a meeting is scheduled using web app, web app sends the name of the user scheduling the meeting as the organizer display name, to the scheduler. A name of choice can be set as display name by including the optional parameter **`organizerDisplayName`** in the scheduler API.

Note: On the Meeting Server 2000 appliance, after configuring and enabling the Scheduler, it is required to execute the MMP commands 'webbridge3 restart' and 'scheduler restart'. The 'Schedule Meeting' option will appear in the Meeting Server Web App only after these services have successfully restarted.

To enable the Scheduler to send email notifications via the SMTP, configure the email server to listen on a specified port for the SMTP protocol.

a. Disable the Scheduler component if it is currently running:

   `scheduler disable`

b. Configure the email server and port:

   `scheduler email server <hostname|address> <port>`

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

c. Enable the Scheduler:

   `scheduler enable`

To enable the Scheduler to send email notifications via the SMTP with Auth Login, configure the email server to listen on a specified port for the SMTP protocol, enable the SMTP server to support Auth Login, and configure a user account for authentication. This account configured on the MMP must have appropriate permissions to be able to send emails on behalf of web app users.

a. Disable the Scheduler component if it is currently running:

```
scheduler disable
```

b. Configure the email server and port:

```
scheduler email server <hostname|address> <port>
```

For example,

```
scheduler email server exchange.example.com 25
scheduler email server 10.27.33.55 25
```

c. Enable the Auth Login option:

```
scheduler email auth enable
```

d. Set the username to be used for authentication:

```
scheduler email username <username>
```

Enter the password:

```
scheduler email username test@test.com
Please enter password:
Please enter password again:
```

e. Enable the Scheduler:

```
scheduler enable
```

To enable the Scheduler to send email notifications via the SMTP and STARTTLS, configure the email server to listen on a specified port for the SMTP protocol and enable STARTTLS.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

a. Disable the Scheduler component if it is currently running:

```
scheduler disable
```

b. Configure the email server and port:

```
scheduler email server <hostname|address> <port>
```

For example,

```
scheduler email server exchange.example.com 25
scheduler email server 10.27.33.55 25
```

c. Enable the STARTTLS option:

```
scheduler email starttls enable
```

d. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

```
scheduler email trust <cert or bundle name>
```

The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

e. Enable the Scheduler component:

```
scheduler enable
```

To enable the Scheduler to send email notifications via the SMTP using Auth Login and STARTTLS, configure the email server to listen on a specified port for the SMTP protocol. Additionally, enable the SMTP server to support Auth Login, configure a user account that will be used for authentication, and enable STARTTLS.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

a. Disable the Scheduler component if it is currently running:

```
scheduler disable
```

b. Configure the specified email server and port:

```
scheduler email server <hostname|address> <port>
```

For example,

```
scheduler email server exchange.example.com 25
scheduler email server 10.27.33.55 25
```

c. Enable the Auth Login option:

```
scheduler email auth enable
```

d. Set the username to be used for authentication:

scheduler email username <username>

Enter the password:

```
scheduler email username test@test.com
Please enter password:
Please enter password again:
```

e. Enable the STARTTLS option:

scheduler email starttls enable

f. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

scheduler email trust <cert or bundle name>

The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

g. Enable the Scheduler component:

scheduler enable

To enable the Scheduler to send email notifications via the SMTPS, configure the email server to support end to end SMTP encryption on a specific port.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

a. Disable the Scheduler component if it is currently running:

scheduler disable

b. Configure the specified email server and port:

scheduler email server <hostname|address> <port>

For example,

```
scheduler email server exchange.example.com 25
scheduler email server 10.27.33.55 25
```

c. Set the email protocol to SMTPS:

scheduler email protcol smtps

d. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

   scheduler email trust <cert or bundle name>

   The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

e. Enable the Scheduler component to complete the email configuration using SMTPS:

   scheduler enable

To enable the Scheduler to send email notifications via the SMTPS using Auth Login, configure the email server to support end to end SMTP encryption on a specific port. Additionally, enable the SMTPS server to support Auth Login and configure a user account that will be used for authentication.

To establish a TLS connection, the TLS handshake involves a certificate exchange between the email server and the Scheduler. By default, the Scheduler is set to trust all certificates and establishes a successful TLS connection by accepting any certificate coming from the email server. However, there is an additional option on the scheduler to configure a specific certificate. In this mode, the Scheduler accepts and trusts only the configured certificate.

a. Disable the Scheduler component if it is currently running:

   scheduler disable

b. Configure the specified email server and port:

   scheduler email server <hostname|address> <port>

   For example,

   ```
   scheduler email server exchange.example.com 25
   scheduler email server 10.27.33.55 25
   ```

c. Enable the Auth Login option:

   scheduler email auth enable

d. Set the username of the user which will be used for authentication:

   scheduler email username <username>

   Enter the password:

   ```
   scheduler email username test@test.com
   Please enter password:
   ```

```
Please enter password again:
```

e. Set the email protocol to SMTPS:

```
scheduler email protcol smtps
```

f. To use a specific certificate, first import and upload the certificate to the Meeting Server VM via SFTP. Then, configure the certificate by running the command:

```
scheduler email trust <cert or bundle name>
```

The configured certificate must be a valid certificate. For example, the common name or SAN names must match the FQDN of the email server, the certificate must not have expired, and so on. Likewise, if the certificate is issued by a Certificate Authority or there are intermediate certificates in the chain, configure the Root CA certificate or alternatively a certificate bundle containing the root certificate, intermediate certificate 1, intermediate certificate 2 and onwards, in that order.

g. Enable the Scheduler component to complete the email configuration using SMTPS with Auth Login:

```
scheduler enable
```

19.1.1 Scheduler detailed logging

The Scheduler supports the option to enable detailed logging for Web Bridge connections, email notifications, and API using the scheduler timedLogging MMP command.

When timedLogging is not enabled, Meeting Server displays the following output:

```
cms-vm> scheduler timedLogging
{
"webBridge": "0",
"api": "0",
"email": "0"
}
```

To enable any of the timedLogging options, use the command:

```
scheduler timedLogging (webBridge|api|email) <time>
```

For example,

```
cms-vm> scheduler timedLogging webBridge 600
SUCCESS
```

The time variable is expressed in seconds, and enables timedLogging for the set duration.

```
cms-vm> scheduler timedLogging
{
"webBridge": "594",
"api": "0",
"email": "0"
}
```

After the set duration expires or the specific investigation or troubleshooting step is completed download the log files using SFTP.

The configuration of the server address and port, enabling email protocol, and configuring a username for authentication are specified via the following scheduler MMP commands:

```
scheduler email server <hostname|address> <port>
scheduler email server none
scheduler email username <smtp username>
scheduler email protocol <smtp|smtps>
scheduler email auth <enable|disable>
scheduler email starttls <enable|disable>
```

Email will not be configured on a scheduler if no server address is configured on it. At least one email server must be configured for the scheduler to send email invites. Emails can be sent from any scheduler and not necessarily from the scheduler which was used to schedule the meeting. If an email server is down, then a different scheduler sends the email.

If the Meeting Server does not receive a response from the Exchange email server within 5000 milliseconds (5 seconds), the Scheduler will automatically reject the request.

4. After configuring the email server, enable the scheduler using the command:

   ```
   scheduler enable
   ```

5. Check the configuration and status of the service using the command:

   ```
   scheduler status
   ```

   Sample output of a successful configuration:

```
1 cms> scheduler status
2 Status: enabled
3 Running
4 Database responsive at start
5 HTTPS configured
6 C2W configured
7 Email server configured
```

```
 8
 9  Scheduler application status:
10  {
11      "status": "UP",
12      "components": {
13          "c2w": {
14              "status": "UP",
15              "details": {
16                  "guid": "dc06c10f-a220-42d8-b4eb-f9be3d07faf4",
17                  "webbridges": "webbridge1.mycompany.com:4443:CONNECTED,
    webbridge1.mycompany.com:8443:CONNECTED,
    webbridge3.mycompany.com:8443:CONNECTED"
18              }
19          },
20          "db": {
21              "status": "UP"
22          },
23          "mail": {
24              "status": "UP",
25              "details": {
26                  "location": "smtp.mycompany.com:25"
27              }
28          },
29          "ping": {
30              "status": "UP"
31          }
32      }
33  }
```

20 Additional security considerations & QoS

This chapter discusses other security features available on the Meeting Server that are in addition to authentication provided through X.509 certificates and public keys.

Note: The commands listed in this chapter are also listed in the MMP Command Reference guide.

20.1 Common Access Card (CAC) integration

The Common Access Card (CAC) is used as an authentication token to access computer facilities. The CAC contains a private key which cannot be extracted but can be used by on-card cryptographic hardware to prove the identity of the card holder.

The Meeting Server supports administrative logins to the SSH and Web Admin Interface using CAC. Use the MMP commands in Table 14 below to configure CAC for your deployment.

Table 14: MMP commands to configure CAC logins

MMP commands	Description
`cac enable|disable [strict]`	Enables/disables CAC mode with optional strict mode removing all password-based logins
`cac issuer <ca cert-bundle>`	Identifies trusted certificate bundle to verify CAC certificates
`cac ocsp certs <keyfile> <certificatefile>`	Identifies certificate and private key for TLS communications with OCSP server, if used
`cac ocsp responder <URL>`	Identifies URL of OCSP server
`cac ocsp enable|disable`	Enables/disables CAC OCSP verification

20.2 Online Certificate Status Protocol (OCSP)

OCSP is a mechanism for checking the validity and revocation status of certificates. The MMP can use OCSP to work out whether the CAC used for a login is valid and, in particular, has not been revoked.

20.3 FIPS

You can enable a FIPS 140-2 level 1 certified software cryptographic module, then cryptographic operations are carried out using this module and cryptographic operations are restricted to the FIPS approved cryptographic algorithms.

Table 15: MMP commands to configure FIPS

MMP commands	Description
`fips enable|disable`	Enables/disables the FIPS-140-2 mode cryptography for all cryptographic operations for network traffic. After enabling or disabling FIPS mode, a reboot is required
`fips`	Displays whether FIPS mode is enabled
`fips test`	Runs the built-in FIPS test

20.4 TLS certificate verification

You can enable Mutual Authentication for SIP and LDAP in order to validate that the remote certificate is trusted. When enabled, the Call Bridge will always ask for the remote certificate (irrespective of which side initiated the connection) and compare the presented certificate to a trust store that has been uploaded and defined on the server.

Table 16: MMP commands to configure TLS

MMP commands	Description
`tls <sip|ldap> trust <crt bundle>`	Defines Certificate Authorities to be trusted
`tls <sip|ldap> verify enable|disable|ocsp`	Enables/disables certificate verification or whether OCSP is to be used for verification
`tls <sip|ldap>`	displays current configuration

20.5 User controls

MMP admin users can:

- Reset another admin user's password
- Set the maximum number of characters that can be repeated in a user's password – and there are a number of other user password rule additions
- Limit MMP access by IP address
- Disable MMP accounts after configurable idle period

20.6 Firewall rules

The MMP supports the creation of simple firewall rules for both the media and admin interfaces. Note that this is not intended to be a substitute for a full standalone firewall solution and therefore is not detailed here.

Firewall rules must be specified separately for each interface. After setting up a firewall rule on an interface, remember to enable the firewall on that interface. See the MMP Command Reference for full details and examples.

CAUTION: We recommend using the serial console to configure the firewall, because using SSH means that an error in the rules would make the SSH port inaccessible. If you must use SSH then ensure that an allow **`ssh rule`** is created for the ADMIN interface before enabling the firewall.

20.7 DSCP

You can enable DSCP tagging for the different traffic types on the Meeting Server (see the MMP Command Reference).

1. Sign in to the MMP.
2. Use **`dscp (4|6) <traffic type> (<DSCP value>|none)`** to set the DSCP values as required. For example: **`dscp 4 oa&m 0x22`** which sets operations, administration and management for IPv4.
3. Alternatively, use the **`dscp assured (true|false)`** command to force the use of the assured or non-assured DSCP values for the " voice" and " multimedia" traffic types. For example: **`dscp assured true`**

Note: DSCP tagging is for all packets being sent from the Meeting Server only. For PC Client DSCP tagging, Group Policy must be used to define desired DSCP values because Windows controls this, and normal user accounts have no permissions to set DSCP.

20.8 Verifying SSH fingerprints

Administrators connecting to theMeeting Server for the first time via SSH or SFTP, can verify the keys prompted by the Meeting Server by retrieving the fingerprints of the keys installed on the meeting server before logging in.

Table 17: MMP command to retrieve the keys

MMP Command	Description
ssh server_key list	The output displays a list of keys along with the size, type, and fingerprints for all existing keys in the Meeting Server host, among the following keys: • ssh_host_dsa_key.pub • ssh_host_ecdsa_key.pub • ssh_host_ed25519_key.pub • ssh_host_key.pub • ssh_host_rsa_key.pub

21 Diagnostic tools to help Cisco Support troubleshoot issues

In addition to using Syslog records (see Section 3.1.4) to help diagnose deployment issues, the following features are available on the Meeting Server:

- SIP tracing
- log bundle
- generate keyframe for specific call leg
- regular reporting of registered media modules

21.1 SIP Tracing

You can enable additional SIP tracing using the **Logs > Detailed tracing** page in the Web Admin Interface. These logs may be useful when investigating call setup failure issues for SIP endpoints and should be disabled at all other times. To prevent the verbose logging being enabled for longer than necessary, it automatically shuts off after a choice of 1 minute, 10 minutes, 30 minutes or 24 hours. Refer to the Meeting Server Support FAQs on the Cisco website for more troubleshooting information.

Diagnostics for failed login attempts include:

- the IP address of the far end included in event log messages relating to logins
- audit messages generated for unsuccessful logins (minus the user name) and log in session timeouts. They are also generated for successful logins.

21.2 Log bundle

Meeting Server can produce a log bundle containing the configuration and state of various components in the Meeting Server. This log bundle includes the syslog and live.json files. If you need to contact Cisco support with an issue, these files will aid them to speed up their analysis.

The Meeting Server log bundle is generated in the following ways:

- Meeting Server admin can initiate the log bundle download process by connecting the SFTP client to the MMP IP address using the MMP admin user credentials. The system generates and downloads a log bundle with file name **logbundle.tar.gz**.
- Alternatively, the admin can generate the log bundle before initiating the download process using the **`generate_logbundle`** command. A log bundle with file name **generatedlogbundle.tar.gz** is generated.

Command/Examples	Description/Notes
`generate_logbundle`	Generates the log bundle with the file name generatedlogbundle.tar.gz on the respective meeting server. **Note:** Each time this command is executed the latest log bundle replaces the log bundle that was generated earlier.

Download the log bundle using the steps mentioned below:

1. Connect your SFTP client to the IP address of the MMP.
2. Log in using the credentials of an MMP admin user.
3. Run one of these commands in the location where the log bundle must be downloaded:
   a. `sftp get logbundle.tar.gz`
   b. `sftp get generatedlogbundle.tar.gz`
4. Copy the file **logbundle.tar.gz/generatedlogbundle.tar.gz** to a local folder.
5. Rename the file, changing the logbundle part of the filename to identify which server produced the file. This is important in a multi-server deployment.
6. Send the renamed file to your Cisco Support contact for analysis.

Initial file size of the log bundle.tar.gz is 1 Kb, after transfer via SFTP the size will increase depending on the number of files and their size.

Note: In the event that you are not able to download the logbundle due to a slow network connection between a computer and the Meeting Server, you can download the log and live.json files to send to Cisco Support.

21.3 Ability to generate a keyframe for a specific call leg

A **generateKeyframe** object has been added to **/callLegs/<call leg id>**. This is a debug facility, and Cisco Support may ask you to use the feature when diagnosing an issue.

Using the Web Admin interface, select **Configuration** > **API**, then

1. From the list of API objects, tap the ▶ after **/callLegs**
2. Click on the **object id** of the call leg
3. From the list of **Related objects** at the top of the page, click **/callLegs/<call leg id>/generateKeyframe**
4. Click **Create**

This will trigger the generation of a new keyframe in the outgoing video streams for the call leg in question

21.4 Reporting registered media modules in syslog

syslog can print a message every 15 minutes to allow people to monitor whether all media modules are alive and well.

An example from a Meeting Server 2000:

```
2020-08-06T13:21:39.316Z user.info cms2kapp host:server INFO : media module
status 1111111 (1111111/1111111) 7/7 (full media capacity)
```

22 Additional licensing information

Meeting Management is mandatory with Meeting Server 3.0 or later for licensing purposes. If you are using Smart Licensing, then you must connect to the Cisco Smart Software Manager. The support for local license files (traditional licensing mode) has been deprecated and license reservation is introduced.

Note: In an environment where you cannot use Meeting Management or connect to the internet due to security reasons, contact your Cisco Account team for alternate options.

22.1 Licensing

You can find the following information in this chapter:

- How Smart licences work in Meeting Server
- Expired license feature enforcement actions
- How to retrieve licensing information (Smart licensing)
- Smart licensing registration process
- Assigning Personal Multiparty licenses to users
- How Cisco Multiparty licenses are assigned
- Determining Cisco Multiparty licensing usage
- Calculating SMP Plus license usage
- Retrieving license usage snapshots from a Meeting Server
- License reporting

22.1.1 How Smart licenses work in Meeting Server – overview

Meeting Management is mandatory for licensing to work on Meeting Server. A trust and interaction between Meeting Server and Meeting Management supports the licensing using Smart or for existing customers use of installed licensing files – it's this trusted link that enables Meeting Management to license Meeting Server.

Note: For full details on using Cisco Meeting Management to administer Smart Licensing, see the Meeting Management Administrator Guide.

A high level work flow for implementing Smart Licensing is as follows:

1. Register your Meeting Management to Smart Licensing Virtual Account.
2. When a Meeting Server first starts up it will have no license status values defined.

   Note: You can use Trial Mode for a 90 day full featured period without licenses.

3. When Meeting Server first connects to a Meeting Management instance set up to administer Smart Licensing, it checks to see if the Meeting Server has previously had a license applied. If not, it will set the license expiry date to 90 days in the future.

   The expiry date for a license is shown in Meeting Management and also returned in the clusterLicensing API, as shown in Appendix B.5.

   Note: The expiry date for any feature license will only ever be up to a maximum of 90 days in the future.

4. Meeting Management collates Meeting Server licensing usage for the cluster and reports to your Smart Account on a daily basis to check that it has the licenses required to ensure the Meeting Server is in compliance. The Smart Account responds to Meeting Management to indicate if the Meeting Server is compliant or not. Meeting Management then sets the expiry dates as appropriate as follows:

   a. If the Meeting Management identifies that a license exists and is below entitlement for a particular feature, the expiry date will be extended to 90 days in the future.

      Note: If Meeting Server doesn't connect to Meeting Management and send usage data for a period of 90 days then the Meeting Server's license won't get refreshed and will therefore expire. For information on the enforcement actions when a license expires, see Section 22.1.2.

   If a license usage is higher than the entitlement, or a license is not found, then enforcement occurs as follows.

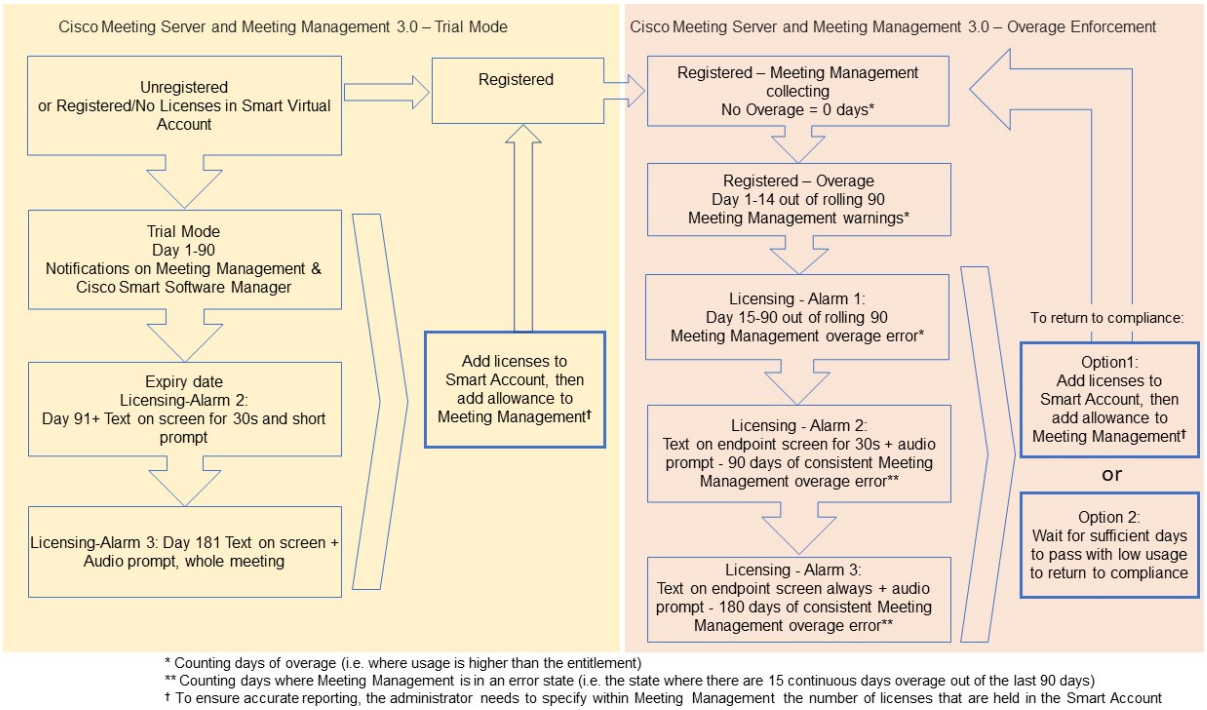
   b. If Meeting Management identifies that less than 15 out of the last 90 days are non-compliant, it will allow this and reset the Meeting Server expiry date to 90 days in the future from that point. The admin will get a visual warning to notify " Insufficient licenses" .

   c. If Meeting Management identifies that more than 15 of the last 90 days are non-compliant, the first level of enforcement (Alarm 1) will occur, i.e. out of compliance notifications on the Meeting Management interface.

   d. If overage continues, Meeting Management does not reset the 90 day clock, it gives you a countdown in xx days in which to add new licenses otherwise Alarm levels 2 and 3 will be enabled for all participants joining a meeting as shown in Figure 45.

Figure 45 shows the enforcement flow from initial start up in trial mode on the left-hand side through to overage enforcement as shown on the right-hand side.

Figure 45: Cisco Meeting Server and Cisco Meeting Management Smart Licensing enforcement flow



* Counting days of overage (i.e. where usage is higher than the entitlement)
** Counting days where Meeting Management is in an error state (i.e. the state where there are 15 continuous days overage out of the last 90 days)
† To ensure accurate reporting, the administrator needs to specify within Meeting Management the number of licenses that are held in the Smart Account

22.1.2 Expired license feature enforcement actions

Previously, Meeting Server would evaluate its license file on restart only. From 3.0 the current status of whether a feature is licensed or not can change dynamically, for example, because a feature license expires (previously this would not have been evident until a restart), or there has been an API change. Meeting Management will calculate enforcement actions with Smart Licensing.

Note: You can use the Smart Licensing portal to enable email notifications for " insufficient licenses" .

When a license feature has expired the actions described in Table 18 will occur.

Table 18: Expired license enforcement actions

Feature	Action
callBridge	When expired: a visual text message displays on screen lasting 30 seconds and an audio prompt plays on joining a meeting for all participants/all meetings. (Alarm level 2)
callBridgeNoEncryption	When expired more than 90 days ago or no license present: the same as before but the visual message is permanent. The audio prompt plays " Your deployment is out of licensing compliance, please contact your administrator" . (Alarm level 3) . However, encrypted calls are not processed in the unlicensed state. Note: you only need callBridge or callBridgeNoEncryption to prevent the above action.
PMP/SMP	
customizations	When expired or not present, customization features will not be active during a meeting.
recording	When expired or not present you will not be able to start a new recording (regardless of whether it is a 3rd party recorder or not). This license represents recording and streaming so the same restrictions also apply to streaming.

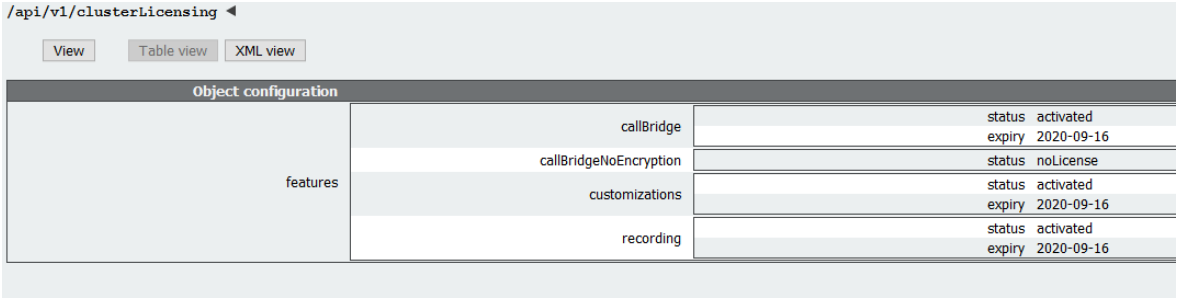
To turn off Alarms 2 and 3, simply add more licenses to your Smart Account.

22.1.3 How to retrieve licensing information (Smart Licensing)

To retrieve licensing information for a cluster using the Meeting Server Web Admin interface:

1. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
2. From the list of API objects, tap the ▶ after **/api/v1/clusterLicensing**
3. The current license status for the cluster is displayed as shown in this example:

Figure 46: clusterLicensing API – license status



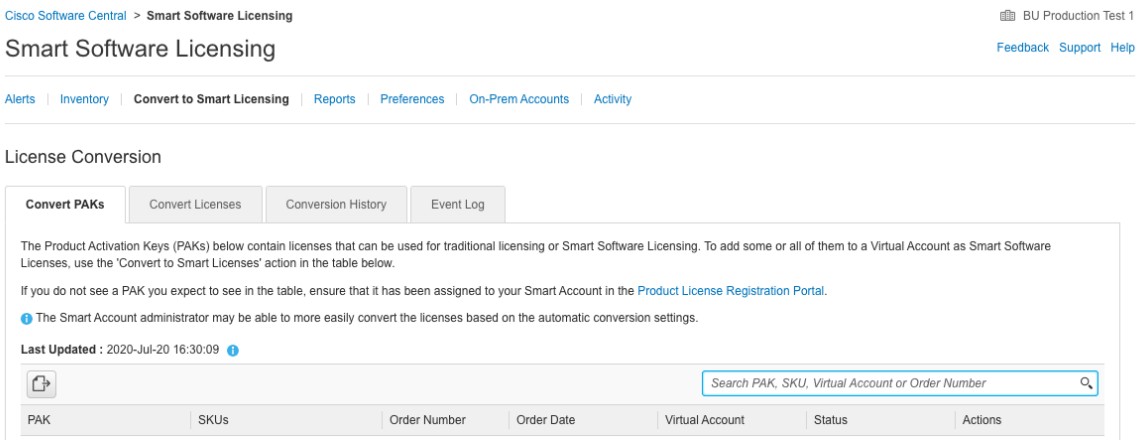
22.1.4 Smart Licensing registration process

To enable Smart Licensing:

1. Sign in to Cisco Smart Software Manager (CSSM) portal and choose Virtual Account with Meeting Server Licenses.
2. Generate a registration token.
3. Copy the token to your clipboard.
4. Open the instance of Meeting Management that you want to use for license reporting.
5. Go to the **Settings** page, **Licensing** tab.
6. Click **Change**.
7. Choose **Smart Licensing** and **Save**.
8. Click **Register**.
9. Paste the registration token (this allows Meeting Management to connect to the Smart Licensing portal).
10. Click **Register**.
11. When you have registered, check how many licenses you have in your Virtual Account.
12. In Meeting Management, go to the **Licenses** page.
13. Enter the license information for the licenses you have in your Virtual Account.

    If any licenses are not shown in your Virtual Account, use the **Convert Licenses** tab, search by PAK to find them, then choose **Convert Licenses** as shown in Figure 47. (If you can't find a license(s), open a case by sending an email to licensing@cisco.com.)

Figure 47: License conversion for Smart Licensing



22.1.5 Multiparty licensing

22.1.5.1 Personal Multiparty plus licensing

Personal Multiparty Plus (PMP Plus) provides a named host license assigned to each specific user who frequently hosts video meetings. This can be purchased through Cisco UWL Meeting

or Flex Meetings (which includes PMP Plus). Personal Multiparty Plus is an all-in-one licensing offer for video conferencing. It allows users to host conferences of any size (within the limits of the Cisco Meeting Server hardware deployed). Anyone can join a meeting from any endpoint, and the license supports up to full HD 1080p60 quality video, audio, and content sharing.

Note: Using Unified Communications Manager, the initiator of an Ad Hoc conference can be identified and if they have been assigned a PMP Plus license then that is used for the conference.

Note: To determine the number of active calls using the PMP Plus licence of an individual, use the parameter **`callsActive`** on API object **`/system/multipartyLicensing/activePersonalLicenses`**. We generally allow 2 calls to be active allowing for one starting and other finishing. If the call is on a cluster of Call Bridges then use the parameter **`weightedCallsActive`** on API object **`/system/multipartyLicensing/activePersonalLicenses`** for each Call Bridge in the cluster. The sum of **`weightedCallsActive`** across the cluster matches the number of distinct calls on the cluster using the individual's PMP Plus license. If a PMP Plus licence is exceeded, then SMP Plus licences are assigned, see Section 22.1.1.

22.1.5.2 Shared Multiparty plus licensing

Shared Multiparty Plus (SMP Plus) provides a concurrent license that is shared by multiple users who host video meetings infrequently. Shared Multiparty Plus enables all employees who do not have PMP Plus host license to access video conferencing. It is ideal for customers that have room systems deployed that are shared among many employees. All users with PMP Plus or using SMP Plus licenses have the same great experience, they can host a meeting with their space, initiate an ad-hoc meeting or schedule a future one. Each shared host license supports one concurrent video meeting of any size (within the limits of the hardware deployed).

Note: To determine the number of SMP Plus licences required, use the parameter **`callsWithoutPersonalLicense`** on API object **`/system/multipartyLicensing`**. If the calls are on a cluster of Call Bridges then use the parameter **`weightedCallsWithoutPersonalLicense`** on API object **`/system/multipartyLicensing`** for each Call Bridge in the cluster. The sum of **`weightedCallsWithoutPersonalLicense`** across the cluster matches the number of distinct calls on the cluster which require an SMP Plus license.

22.1.6 Assigning Personal Multiparty licenses to users

This process requires that users are imported from a single LDAP source. See the " Provisioning – Import users" chapter in the Meeting Management Administrator Guide for full details.

22.1.6.1 To determine whether a specific user has a license:

1. From the list of API objects, tap the ▶ after **/users**
   a. Select the **object id** of the specific user
   b. Identify the **object id** of the **userProfile** associated with this user
2. From the list of API objects, tap the ▶ **/userProfiles**
   a. Select the **object id** of the specific userProfile
   b. Find the setting for parameter **hasLicence**. If set to **true** then the user identified in step 1 is associated with a Cisco Multiparty user license. If set to **false** the user is NOT associated with a Cisco Multiparty user license.

   Note: If the userProfile is deleted, then the userProfile is unset for the ldapSource and the imported users.

22.1.7 How Cisco Multiparty licenses are assigned

When a meeting starts in a space, a Cisco license is assigned to the space. Which license is assigned by the Cisco Meeting Server is determined by the following rules:

- if the space owner is defined and corresponds to a Meeting Server imported LDAP user with an assigned Cisco PMP Plus license, the license of that owner is assigned irrespective of whether the person is active in the conference, if not, then
- if the meeting was created via ad hoc escalation from Cisco Unified Communications Manager, then Cisco Unified Communications Manager provides the GUID of the user escalating the meeting. If that GUID corresponds to a Meeting Server imported LDAP user with an assigned Cisco PMP Plus license, the license of that user is assigned, if not, then
- if the meeting was scheduled via Cisco TMS version 15.6 or newer, then TMS will provide the owner of the meeting. If that user corresponds to a Meeting Server imported LDAP user by user ID/email address with an assigned Cisco PMP Plus license, the license of that user is assigned to the meeting, if not then,
- a Cisco SMP Plus license is assigned.

22.1.8 Determining Cisco Multiparty licensing usage

We recommend you use Meeting Management to view your Multiparty licensing usage. However, the API can be used.

Table 19 below lists the API objects and parameters that can be used to determine the consumption of Multiparty licenses.

Table 19: Objects and parameters related to Multiparty license usage

API object	Parameter (s)	Use to .....
/system/licensing	personal, shared	determine whether components of the Cisco Meeting Server have a Multiparty license and are activated. Values are: noLicense, activated, grace, expired. Also provides date of expiry and number limit.
/system/multipartyLicensing	personalLicenseLimit, sharedLicenseLimit, personalLicenses, callsWithoutPersonalLicense, weightedCallsWithoutPersonalLicense	indicates the number of licenses available and in use
/system/multipartyLicensing/ activePersonalLicenses	callsActive, weightedCallsActive	indicates the number of active calls that are using a Personal Multiparty Plus user license,
/userProfiles	hasLicense	indicates whether or not a user is associated with a Cisco Multiparty user license

For more information on these additional object and fields to support Cisco Multiparty licensing, refer to the Cisco Meeting Server API Reference Guide.

22.1.9 Calculating SMP Plus license usage

For the following specific scenarios, the SMP Plus license consumed for a meeting is reduced to 1/6th of a full SMP Plus license:

- an audio-only conference where no attendees are using video,
- a Lync gateway call unless the Meeting Server is recording or streaming, at which point it is considered a full conference and a full SMP Plus license is consumed,
- a point to point call involving a web app and a SIP endpoint, or two web apps, unless the Meeting Server is recording or streaming, at which point it is considered a full conference and a full SMP Plus license is consumed.

A full SMP Plus license is consumed for any audio-video conference instantiated from a space with the owner property undefined, owned by an imported LDAP user without a PMP Plus license, or owned by an imported LDAP user whose PMP Plus license has already been consumed, this is irrespective of the number of participants.

Note: A point to point call is defined as:

- having no permanent space on the Meeting Server,
- two or less participants, including the recorder or streamer
- no participants hosted on the Lync AVMCU,

This includes Lync Gateway calls as well as other types of calls: point-to-point web app to web app, web app to SIP and SIP to SIP.

22.1.10 Retrieving license usage snapshots from a Meeting Server

An administrator can retrieve license usage from the Meeting Server. These cannot be accessed though the Web Admin Interface, instead use an API tool such as POSTMAN:

Use GET on **`/system/MPLicenseUsage/knownHosts`** to retrieve host ids of the Meeting Servers in the deployment. Supply an offset and limit if required to retrieve host ids other than those on the first page of the list.

Use GET on **`/system/MPLicenseUsage`** to retrieve license usage from the Call Bridge of the Meeting Server with the specified host id. Supply a start and end time for the snapshot. Provides information on number of personal licenses in use, number of shared licenses in use which are audio only, point to point, or neither audio or point to point, number of calls being recorded and number of streamed calls.

Note: Note: personal and shared licenses are normalized over the number of Call Bridges that the call spans.

22.1.11 License reporting

Meeting Management has license reporting/usage information for the last 90 days, and Cisco Smart Software Manager also contains license reporting information. The usage of recording licenses indicates the number of conferences recording concurrently, similarly the streaming license usage indicates the number of conferences streaming concurrently.

22.1.12 Legacy licensing file method

This section only applies if you are using the traditional licensing method. From version 3.4, the support for traditional licensing has been deprecated. The existing local licenses will still be supported until the license expires.

22.1.12.1 Obtaining Cisco user licenses using the traditional licensing method

This section assumes that you have already purchased the licenses that will be required for your Meeting Server from your Cisco Partner and you have received your PAK code(s).

Follow these steps to register the PAK code with the MAC address of your Meeting Server using the Cisco License Registration Portal.

Note: You need a license file for each individual Call Bridge – licenses can only be shared amongst servers in the same cluster. Each license file should have all of the required features that you purchased for that cluster; such as PMP Plus, SMP Plus, Recording, Streaming.

1. Obtain the MAC address of your Meeting Server by logging in to the MMP of your server, and enter the MMP command: **`iface a`**

   Note: This is the MAC address of your VM, not the MAC address of the server platform that the VM is installed on.

2. Open the Cisco License Registration Portal and register the PAK code(s) and the MAC address of your Meeting Server(s).
3. If your PAK does not have an R-CMS-K9 activation license, you will need this PAK in addition to your feature licenses.
4. The license portal will email a zipped copy of the license file. Extract the zip file and rename the resulting xxxxx.lic file to **`cms.lic`**.
5. Using your SFTP client, log into Meeting Server and copy the **`cms.lic`** file to the Meeting Server file system.
6. Restart the Call Bridge(s) using the MMP command **`callbridge restart`**
7. After restarting the Call Bridge(s), check the license status by entering the MMP command **`license`**

   The activated features and expirations will be displayed.

23 Obtaining information on hosted conferences

There are two mechanisms for obtaining information on conferences hosted on the Meeting Server which remove the need to constantly poll the API: Call Detail Records and Events.

Note: You can configure Cisco Meeting Management as a CDR (Call Detail Record) receiver and events client on each Call Bridge to get information about active meetings via API requests, CDRs, and Meeting Server events. For more information, see the Meeting Management User Guide for Administrators.

23.1 Call Detail Records (CDRs)

The Meeting Server generates Call Detail Records (CDRs) internally for key call-related events, such as a new SIP connection arriving at the server, or a call being activated or deactivated.

The server can be configured to send these records to a remote system to be collected and analyzed. There is no provision for records to be stored on a long-term basis on the Meeting Server, nor any way to browse CDRs on the Meeting Server itself.

The CDR system can be used in conjunction with the Meeting Server API, with the call ID and call leg IDs values being consistent between the two systems to allow cross referencing of events and diagnostics.

The Meeting Server supports up to four CDR receivers, enabling you to deploy different management tools or multiple instances of the same management tool, such as Cisco Meeting Management. For more information, see the Cisco Meeting Server Call Detail Records Guide.

23.2 Events

Meeting Server can notify an " events client" in real-time of changes that are occurring on the Meeting Server. The Meeting Server acts as a server for the events, and the events client could be for example, a web-based management application. Cisco Meeting Management acts as an events client.

Note: You can construct your own events client, which is similar to constructing an API client. The events client needs to support HTTP and WebSocket libraries, both are available in common scripting languages like Python. The events port on the Meeting Server is the same port as you configured for the Web Admin, typically TCP port 443 on interface A.

Rather than continually poll an API resource on the Meeting Server, an events client can subscribe to an event resource to receive updates. For example, after establishing a WebSocket connection between the events client and the Meeting Server, the events client can subscribe to the event resource `callRoster` and receive updates on the participant list of

an active conference to find out when a new participant joins, or an existing participant changes layout etc.

For more information, see the Cisco Meeting Server Events Guide.

Appendix A DNS records needed for the deployment

Note: You can configure the DNS resolver(s) to return values which are not configured in external DNS servers or which need to be overridden; custom Resource Records (RRs) can be configured which will be returned instead of querying external DNS servers. (The RR is not available to clients.) See the MMP Command Reference for details.

Note: Verify that no A or SRV records already exist for any Meeting Servers before defining the records below.

Table 20: DNS records required for deployment

Type	Example, Description and Resilience Considerations
A / AAAA	**ukedge1.example.com** Resolves to: IP address of Web Bridge. Description: This record is used by the Meeting Server to connect to the Web Bridge.
A / AAAA	**join.example.com** Resolves to: 1) a single DNS entry of a Web Bridge (if using one), or 2) if network load balancing on DNS, then a single DNS entry based on IP or location, or 3) a single IP of a network load balancer, or 4) one or more IP addresses of Cisco Expressways. Description: It is common practice to provide an end user with an FQDN to type into the browser which resolves to this record.
A / AAAA	**ukedges.example.com** **nyedges.example.com** Resolves to: IP addresses of Web Bridges. Description: Used by Call Bridge to connect to Web Bridge 3 (for c2w).

Type	Example, Description and Resilience Considerations
A / AAAA	**ukcore1.example.com** **nycore1.example.com** Resolves to: IP address of the Call Bridge. Description: Used by the Lync FE server to contact the Call Bridge. Resilience considerations: One record per Call Bridge. Each Call Bridge must have a unique FQDN.
A / AAAA	**ukcore1admin.example.com** **ukedge1admin.example.com** **ukcoreadmin.example.com** Resolves to: IP address of the MMP interface. Description: This record is used purely for admin purposes; when system administrators prefer a FQDN to remember for each MMP interface. Resilience considerations: One record per Web Admin Interface. Each MMP interface must have a unique FQDN.
SRV(*)	**_sipinternaltls._tcp.<yourLyncdomain>** Resolves to: The A record of the Lync FE server or FE Pool. Description: If you have an FE pool, you can have multiple FE records pointing to individual FE servers within the pool. You also need this record if you want the Meeting Server to resolve Lync meetings by Lync meeting IDs.
A / AAAA	**fe.<yourLyncdomain>** Resolves to: IP address of the Lync FE server. Description: You will need one record for each individual FE server.
SRV(*)	**_sipfederationtls._tcp.<yourSIPdomain>** Resolves to: The FQDN of the Meeting Server. Description: This record is required for Lync federation.

Type	Example, Description and Resilience Considerations
A	**`callbridge.example.com`** Resolves to: IP address of the Call Bridge. Description: Required for Lync federation as the Call Bridge will need to have a public IP address, and NAT is not supported in this scenario.

(*) SRV records do not resolve directly to IP addresses. You need to create associated A or AAAA name records in order to satisfy the SRV requirements.

Appendix B Ports required for the deployment

The following diagram shows the connections to the Meeting Server and location of the firewall in a scalable and resilient server deployment. Use the tables below the diagram to identify which ports to open.

Figure 48: Ports that must be open in a scalable and resilient server deployment with Expressway in the DMZ

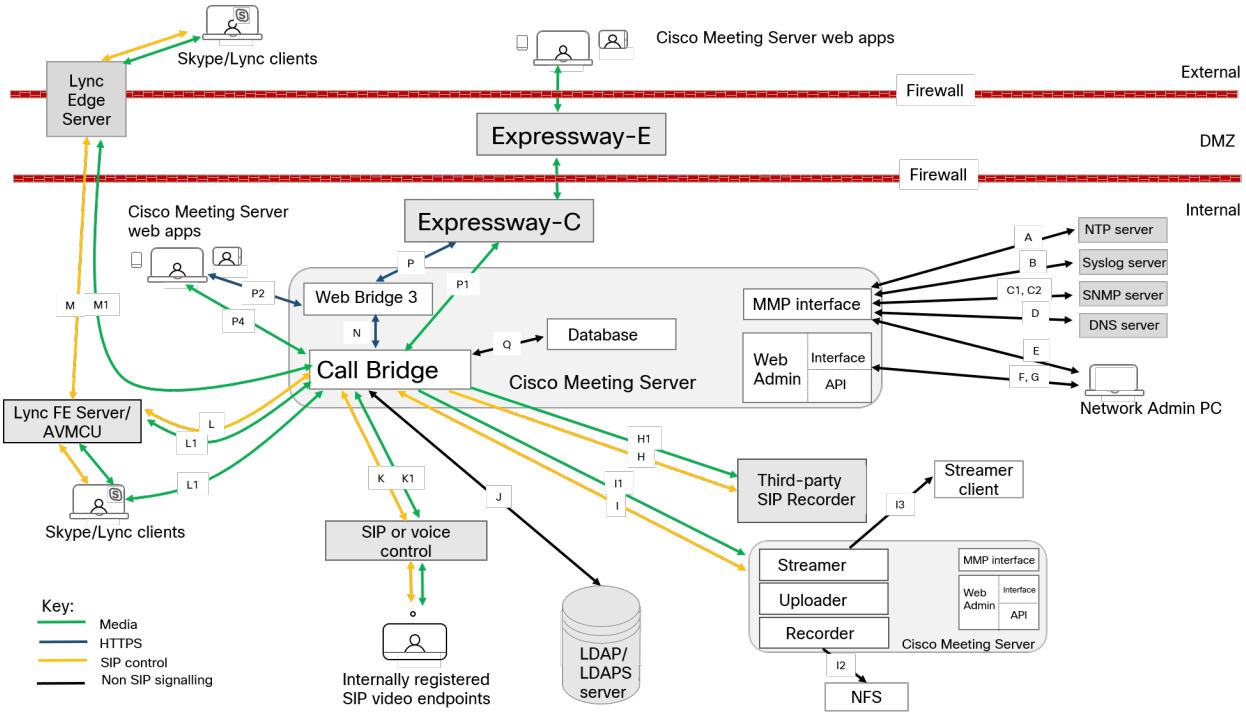
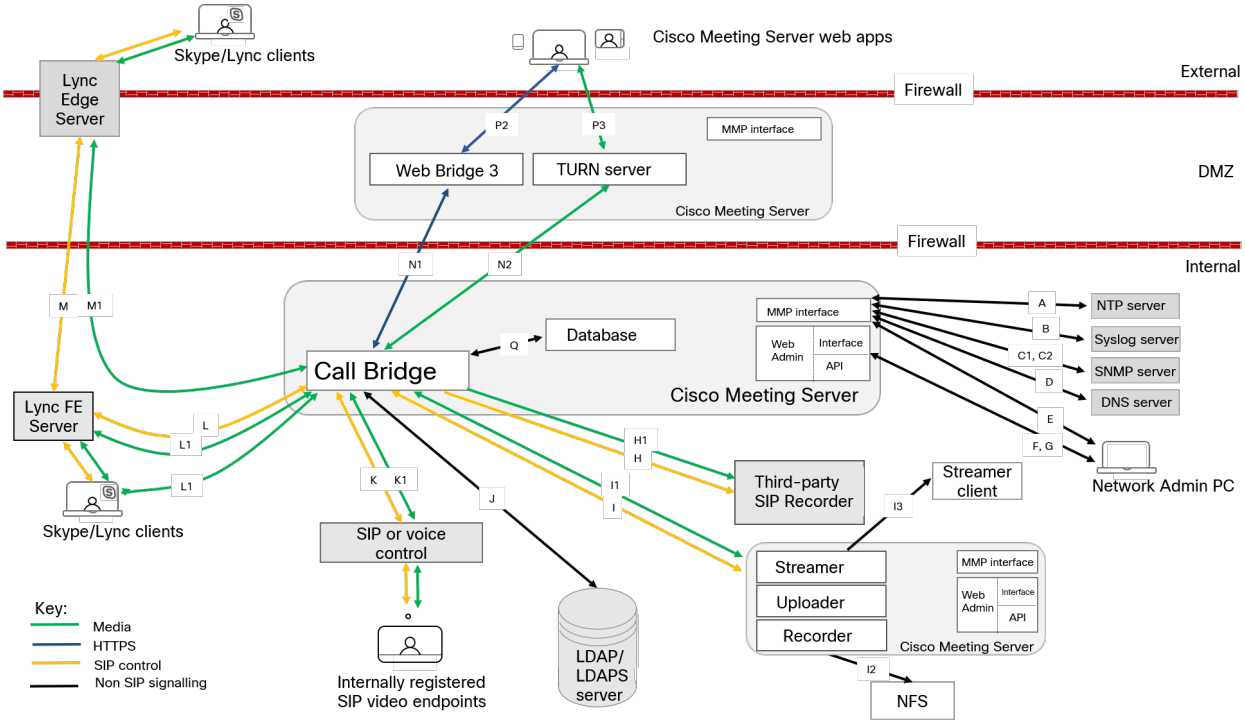


Figure 49: Ports that must be open in a scalable and resilient server deployment using the TURN server and Web Bridge 3 components in the DMZ



B.1 Configuring the Meeting Server

Table 21 lists the ports to use to configure the Meeting Server.

Table 21: Ports for administration of the Meeting Server

Code	Connect to	Destination port to open	Method	Traffic type	Traffic direction with respect to Meeting Server	Additional information
E	MMP	22	SSH	TCP	Incoming	Secure login to MMP
F	API or Web Admin	80	HTTP	TCP	Incoming	Port enabled/disabled through MMP
G	API or Web Admin	443	HTTPS	TCP	Incoming	Port configurable through MMP

B.2 Connecting services

Use Table 22 to identify which ports are used to connect different services to the web app.

Table 22: Ports to open to connect services

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
A	MMP	NTP server	123	TCP or UDP	Outgoing	
B	MMP	Syslog server	514	TCP	Outgoing	Default port, different port configurable through MMP
C1	MMP	SNMP server	161	UDP	Incoming	
C2	MMP	SNMP TRAP	162	TCP or UDP	Outgoing	
D	MMP/Call Bridge/Web Bridge	DNS server	53	TCP or UDP	Outgoing	
	Call Bridge	CDR recipient device		TCP	Outgoing	set URI of CDR recipient in Web Admin interface, or API using API object /system/cdrReceivers/

B.3 Using Meeting Server components

Use Table 23 to identify which ports are used to connect to the components in the Meeting Server and the ports that need to be open through the firewall.

Table 23: Ports to open to use Meeting Server components

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
H	Call Bridge	3rd party SIP recorder	5060	TCP (SIP)	Outgoing	
			5060	UDP (SIP)		
			5061	TLS (SIP)		
H1	Call Bridge	3rd party SIP recorder		Media	Outgoing	ports determined by 3rd party SIP recorder
			32768-65535	UDP (STUN, RTP, BFCP)	Incoming	

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
I	Call Bridge	Recorder/ Streamer	5060	TCP (SIP)	Outgoing	Ports configurable through MMP. For a local recorder use the loopback interface, e.g. lo:8443
			5061	TLS (SIP)		
			5060	TCP (SIP)	Incoming	
			5061	TLS (SIP)		
I1	Call Bridge	Recorder/ Streamer	32768-65535	Media	Outgoing	
			32768-65535	UDP (STUN, RTP, BFCP)	Incoming	
I2	Recorder	Network File Server (NFS)				Use the MMP command recorder nfs <hostname/IP<directory> to specify where to store the recordings on the NFS
I3	Streamer	Streamer client	1935	RTMP	Outgoing	
J	Call Bridge	LDAP/LDAPS (Active Directory)	389/636 (Note 1)	TCP/TCP (SIP TLS)	Outgoing	Port configurable through Web Admin interface
K	Call Bridge	Internal registered SIP endpoint or voice call control	5060	UDP (SIP), TCP (SIP)	Incoming and outgoing	
			5061	TCP (SIP TLS)		
K1	Call Bridge	Internal registered SIP endpoint or voice call control	32768-65535	UDP (STUN, RTP, BFCP)	Incoming	
L	Call Bridge	Lync FE server/ AVMCU	5061	TCP (SIP TLS)	Incoming and outgoing	

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
L1	Call Bridge	Lync client, Lync FE server / AVMCU	1024-65535 (Note 2)	UDP (STUN, RTP)	Outgoing	
			32768-65535	UDP (STUN, RTP)	Incoming	
			1024-65535 (Note 2)	TCP (RDP)	Outgoing	
			32768-65535	TCP (RDP)	Incoming	
M	Call Bridge	Lync Edge server	3478	UDP	Outgoing	
			443	TCP	Outgoing	
M1	Call Bridge	Lync Edge server	32768-65535	UDP (STUN, RTP)	Incoming	
N	Call Bridge	Web Bridge 3		TCP (C2W)	bidirectional data flow	Note: every Call Bridge in a cluster needs to connect to every Web Bridge.
N1	Call Bridge	Web Bridge 3	9999	TCP (C2W)	bidirectional data flow	
N2	Call Bridge	TURN Server	50000-62000 (Note 4)	UDP (RTP, STUN)	Outgoing	Firewall must allow return UDP traffic
P	Web Bridge 3	Expressway	443	TCP (HTTPS)	Incoming and outgoing	
			80	TCP (HTTP)	Incoming	Port 80 optional for HTTP > HTTPS redirect
P1	Call Bridge	Expressway	1024-65535	UDP (STUN, RTP)	Incoming and outgoing	Port 3478 always in use, ephemeral ports will be allocated within the range as needed per call
P2	Web Bridge 3	Cisco Meeting Server web app	443	TCP (HTTPS)	Incoming and outgoing	Port 80 optional for HTTP > HTTPS redirect

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
P3	TURN server	Cisco Meeting Server web app	3478 (Note 3) (Note 4)	UDP (RTP, STUN)	Incoming	Firewall must allow return UDP traffic
P4	Call Bridge	Cisco Meeting Server web app	1024-65535	Media TCP/UDP (STUN RTP)	Incoming and outgoing	
Q	Call Bridge	Database				Internal to Meeting Server, does not require open ports on the firewall

Note:

Note 1: Port 636 (secure) and 389 (non-secure) are commonly used for this function but the port is configurable through the Web Admin interface. The same applies to 3268 and 3269 (non-secure and secure) global catalog LDAP requests.
Note 2: Exact range depends on configuration of Lync server.
Note 3: Admin may optionally enable 3478 TCP or another customer TCP port for TURN.
Note 4: TURN and Media ranges assume web app allocates TURN relay and Call Bridge does not create TURN relays as documented in this guide.

B.4 Additional ports required for Scalability and Resilience

Figure 50: Additional ports required to be open in a scalable and resilient multiple server deployment

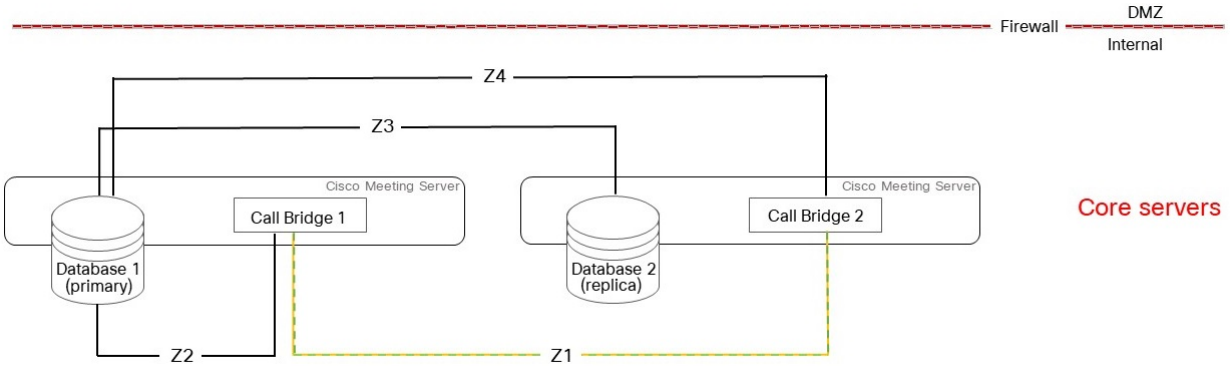


Table 24: Additional ports required to be open in a scalable and resilient multiple server deployment

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
Z1	Call Bridge 1	Call Bridge 2	443	TCP	Incoming and outgoing	for Call Bridge clustering
Z1	Call Bridge 1	Call Bridge 2	5060	TCP/UDP (SIP)	Incoming and outgoing	Figure 50 shows direct flow of SIP and media traffic between Call Bridges. Note that all SIP calls can be done via call control (i.e. CUCM or VCS) and the media could also flow via other devices including Turn servers within the deployment.
Z1	Call Bridge 1	Call Bridge 2	5061	TCP (SIP) TLS	Incoming and outgoing	
Z1	Call Bridge 1	Call Bridge 2	32768-65535	UDP (SIP) BFCP	Incoming	
Z1	Call Bridge 1	Call Bridge 2	1024-65535 (note 1)	UDP (SIP) BFCP	Outgoing	
Z1	Call Bridge 1	Call Bridge 2	32768-65535	UDP STUN/RTP	Incoming and outgoing	
Z2	Call Bridge 1	Database 1 (primary)	n/a	TCP		
Z3	Database 1 (primary)	Database 2 (replica)	5432	TCP	Incoming and outgoing	Database cluster / replication ports
Z4	Call Bridge 2	Database 1 (primary)	5432	TCP	Incoming and outgoing	

Note:
1) Exact range depends on far end

B.5 Ports open on loopback

The ports listed in Table 25 are open on the loopback interface.

Table 25: Ports on loopback

Port	Usage	Notes
53	DNS	
123	NTP	
1234	HTTP	Not applicable to Cisco Meeting Server 2000
2829, 2830	Server to media internal connection	

Port	Usage	Notes
3521	configd	
5432	postgres	
5060	SIP	always open
5061	encrypted SIP	only if certificates applied to Call Bridge
5070	BFCP	only on IPv6
8080	HTTP	always open
8081	HTTP	if webadmin enabled
3478	STUN	

Appendix C Sharing Call Bridge licenses within a cluster

Note: This section only applies to the traditional licensing method.

This section assumes that you have already purchased the licenses that will be required for your Call Bridge cluster and you have received your PAK codes.

Note: You need a license file for each individual Call Bridge – licenses can only be shared amongst servers in the same cluster. Each license file should have all of the required features that you purchased for that cluster; such as PMP Plus, SMP Plus, Recording/Streaming.

Note: ACUs are no longer supported from Meeting Server version 3.0 onwards.

C.1 Registering your Cisco Meeting Server activation PAK codes

Follow these steps to register and license a Call Bridge cluster. Using the Cisco License Registration Portal:

1. Register your Meeting Server activation PAK code against the MAC address of one of your Meeting Server nodes that will be used in the Call Bridge cluster. You will receive a single **cms.lic** file with licenses for the Call Bridge, TURN server and Web Bridge for this primary node.
2. Register your remaining PAK codes for the cluster against the same MAC address used in step 1. The PAK codes are associated with the features you purchased for the cluster (branding, SMP Plus, PMP Plus, Recording, Steaming etc). If you have multiple PAK codes, you will receive a new **cms.lic** file each time you register a PAK code, the new file being an aggregate of the previous files.

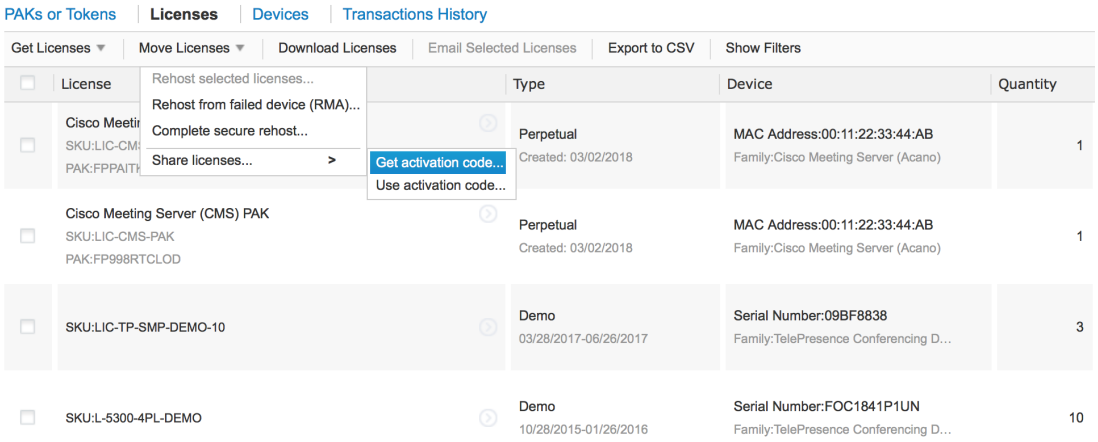
   Note: You can open the **cms.lic** in a text editor to check on the features purchased.

3. Once you have registered all of the features, you need to register the other Call Bridges in the cluster. Register a Meeting Server activation PAK code against the MAC address of each of the other Meeting Server nodes that will be used in the Call Bridge cluster. You will receive a **cms.lic** file for each secondary node.

C.1.1 Sharing feature licenses across the cluster

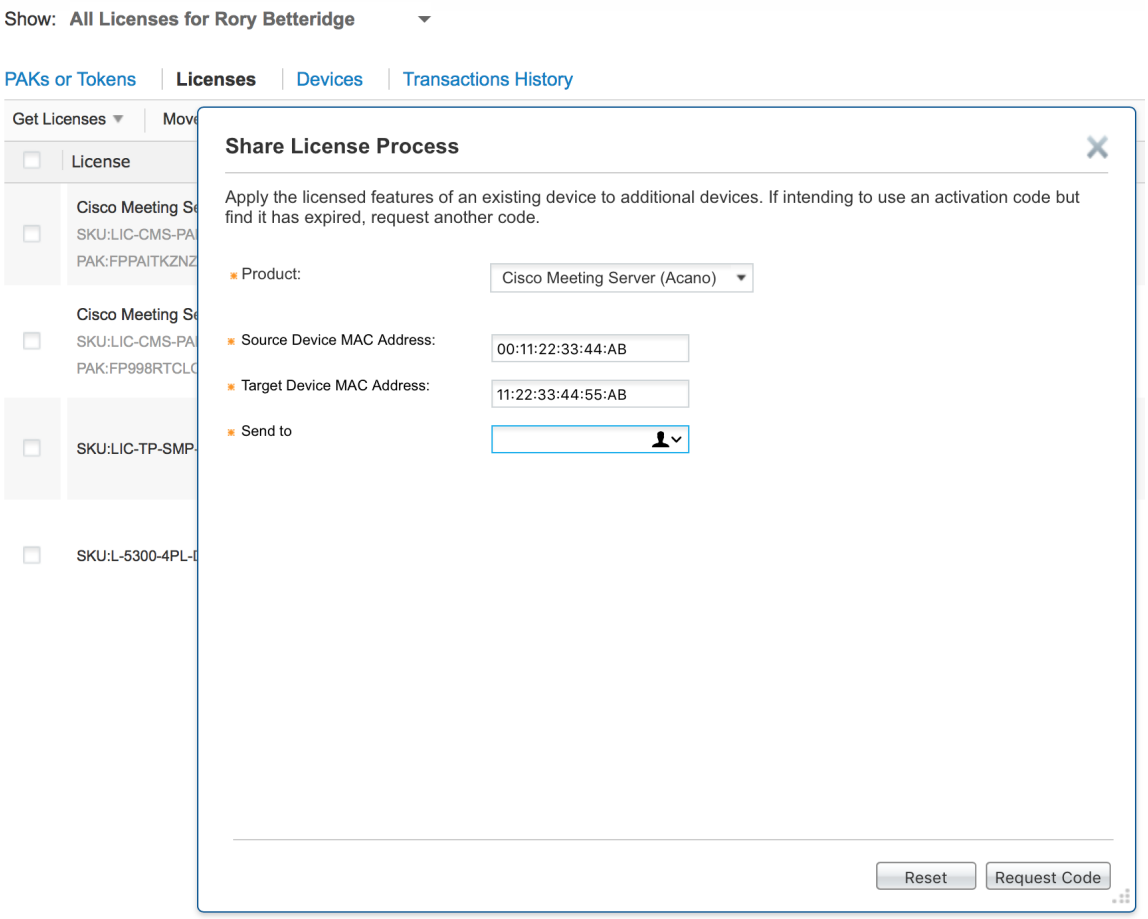
4. You now need to share the feature licenses registered to the primary node with the other secondary nodes. For each Meeting Server node that you registered in step 3, complete steps a to k below.

   a. In the Licence Registration Portal, select the **Licenses** tab, and navigate to **Move Licenses** > **Share Licenses** > **Get Activation Code**.

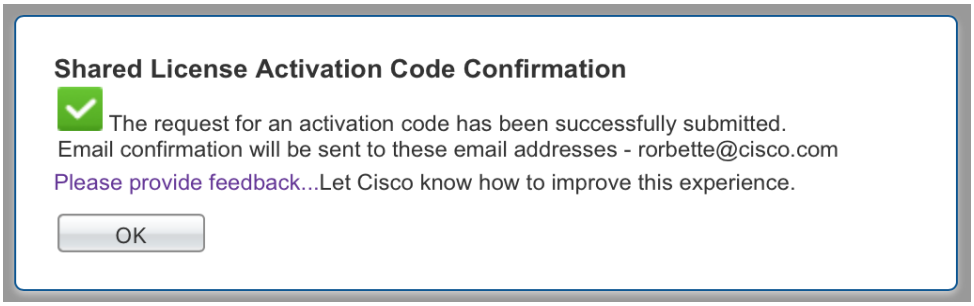


   b. Select **Cisco Meeting Server** and enter the MAC address of your first node (used in steps 1 and 2) in the **Source Device MAC Address:** field.

   c. Enter the MAC address of one of the other nodes in the **Target Device MAC Address:** field. Enter your email address in the **Send to** field.



d. A Shared License Activation Code Confirmation message displays.



e. You will receive an activation code by email for the target node you entered in step c.

```
PLEASE DO NOT DISCARD THIS EMAIL.

You have received this email because your email address was provided to Cisco Systems during the registration pr

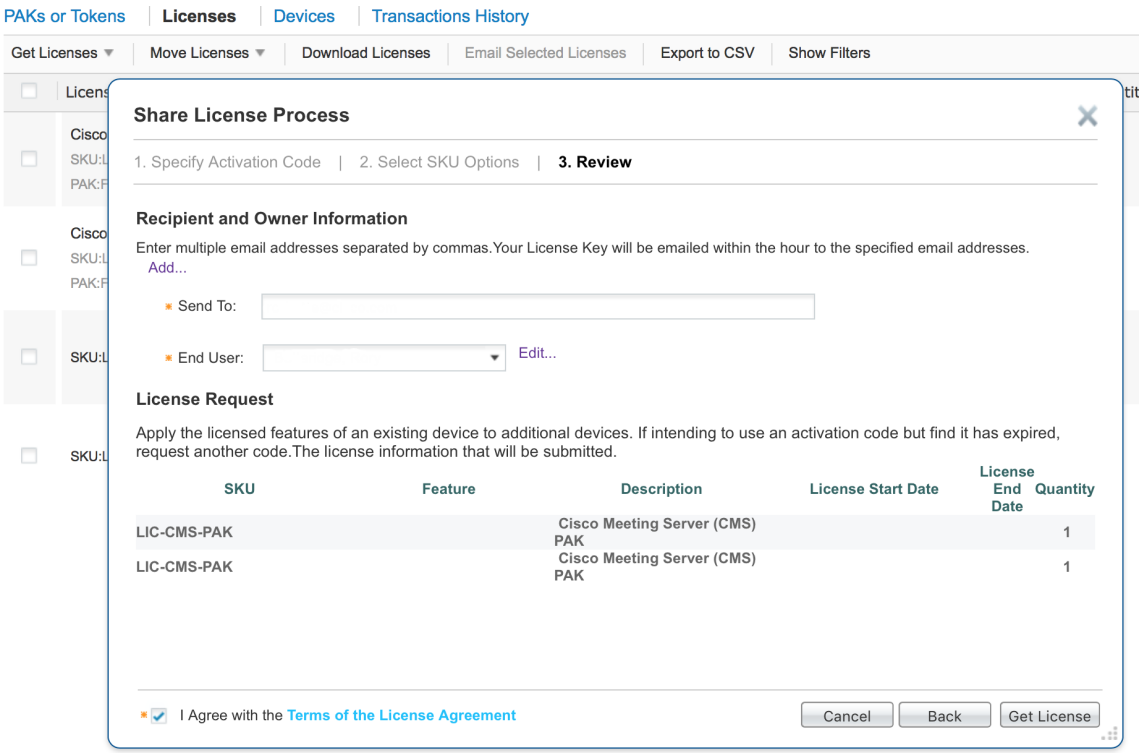
Below, you will find the Activation Code:
Activation Code    : 8NF7R5E1

Here is the device registration information:
Existing Device Serial # : 00:11:22:33:44:AB
New Device Serial # : 11:22:33:44:55:AB

Please click the below link and follow the instructions given below to continue the registration process:
https://slexui.cloudapps.cisco.com/SWIFT/LicensingUI/Quickstart
1) Click on Other Licenses drop down and Select 'Share License Process' option.
2) Select 'Use Activation code' option and Enter the above activation code in 'Specify Activation Code' tab and
```

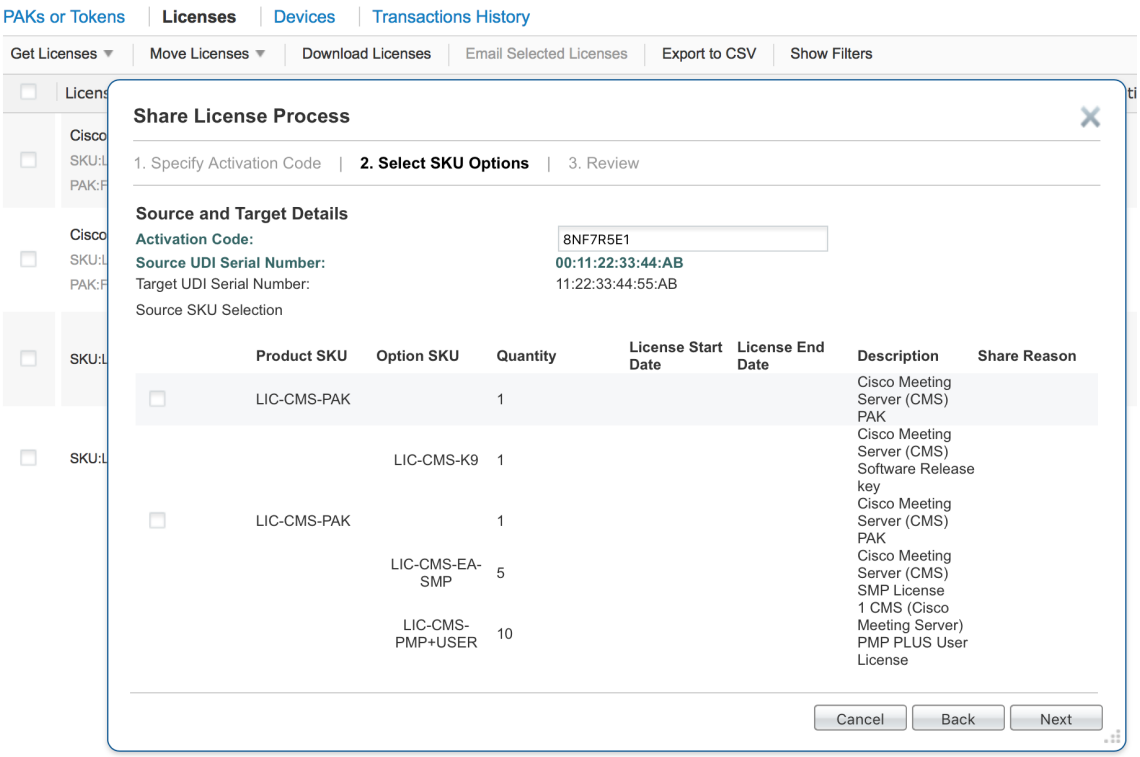
f. Return to the **Licenses** tab, and navigate to **Move Licenses** > **Share Licenses** > **Use Activation Code**.

g. Enter the activation code you received in step e and click **Next**

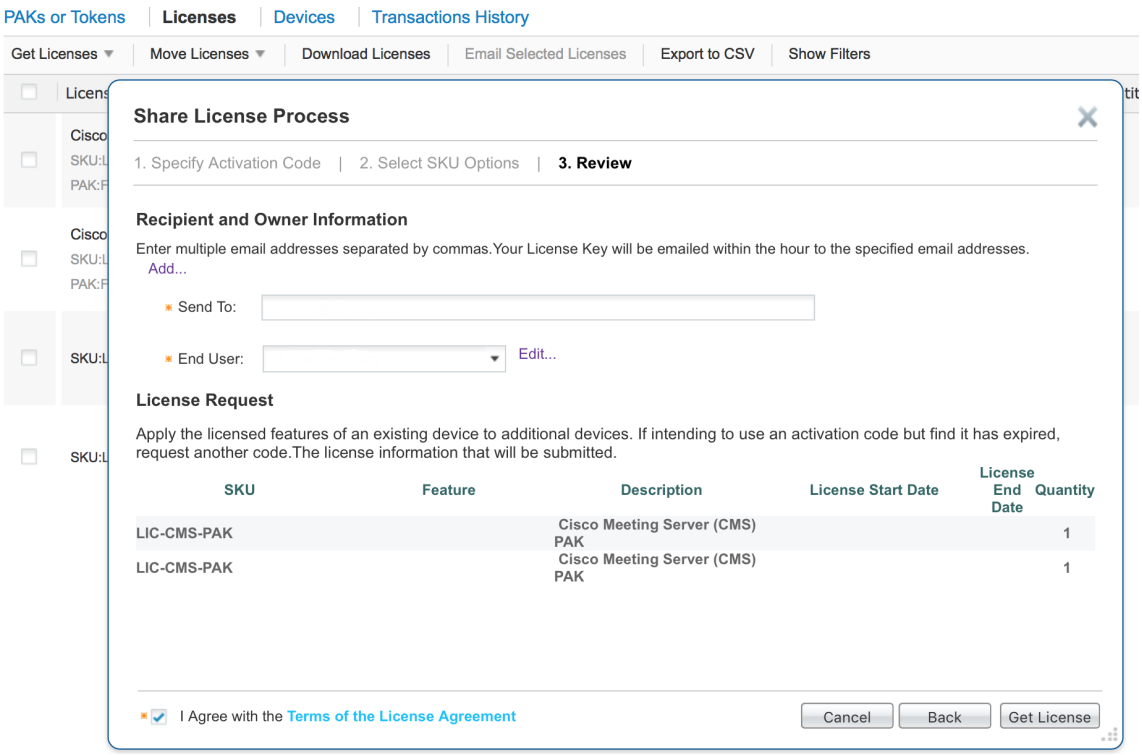


h. Tick the boxes for the features you registered in step 2, and click **Next**.

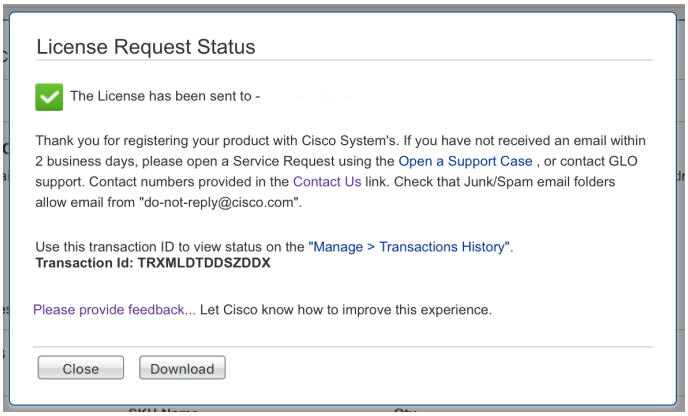
Note: You don't need to tick the box for the **software release key** as this secondary node has already been registered.



i. Enter the email addresses to receive the license for this secondary node. Tick the **I agree with the Terms of the License Agreement** at the bottom of the screen. Click **Get License**.

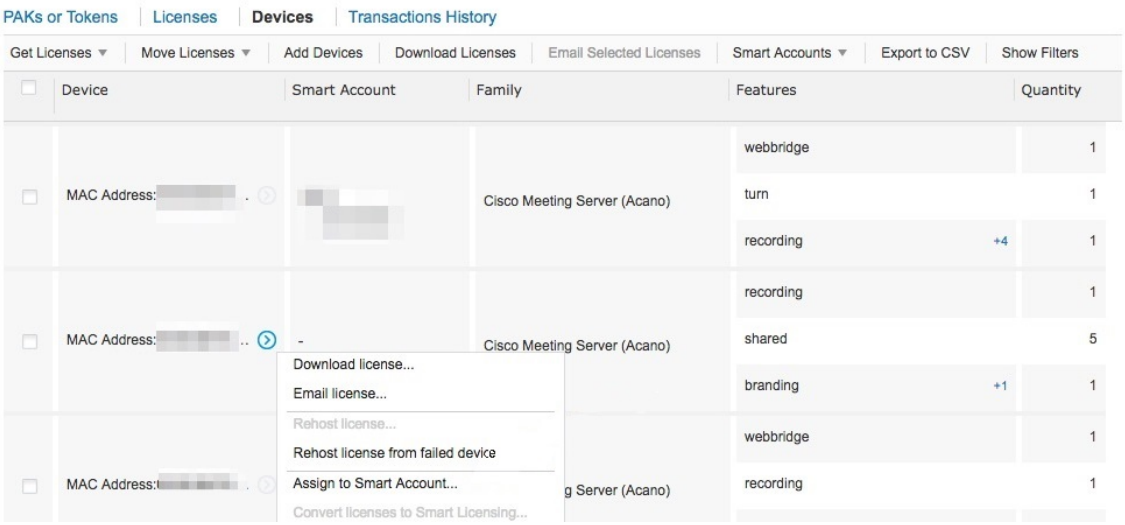


j. Make a note of the " transaction ID" in case there is an issue with the license being released.



k. You will receive via email, a new **cms.lic** file for this secondary node, containing the licenses for the Call Bridge, TURN server and Web Bridge on this Meeting Server, and the registered features.

5. **Optional step**: After completing steps 4a to 4k for each secondary node, you can assign the licenses to your Smart Account so you can easily manage and control Cisco licenses across your entire organization. Your Smart Account is accessible through the **Devices** tab in the Cisco License Registration Portal.



C.2 Adding licenses to an existing Call Bridge cluster

Register your newly acquired PAK code, as detailed in step 1, to one of the Meeting Server MAC addresses—this can be on any node. This action will add the new licenses to any existing licenses. You then need to repeat step 4 to share the new licenses across the cluster.

Appendix D Unclustering

D.1 Unclustering Call Bridges

It is possible to have some Call Bridges clustered and accessing a database cluster; however any Call Bridges that are not in the Call Bridge cluster cannot access the clustered databases and will use a local database.

1. Remove a Call Bridge from the cluster:
   a. On that Call Bridge's Core Server sign in to the Web Admin Interface and go to **Configuration > Cluster**.
   b. Select the check box next to the Call Bridge's entry and click **Delete**.

   This takes that Call Bridge out of the Call Bridge cluster.

2. Then remove the co-located database (if any) from the database cluster.
   a. Sign in to the Call Bridge server's MMP.
   b. Enter the **`database cluster remove`** command.

   The database is disconnected from the database cluster and the space database cluster's contents are no longer accessible to the Call Bridge on this Core server; however the original database contents become accessible again. (Note that they may be out of date compared to the contents of the clustered databases.)

Appendix E Call capacities by Cisco Meeting Server platform

Table 26 below details maximum call capacities on Meeting Servers by upgrading to later software versions. Note that there are different capacities for a single or cluster of Meeting Servers compared to load balancing calls within a Call Bridge Group.

Table 26: Meeting Server call capacity for clusters and Call Bridge groups

Cisco Meeting Server platform		Cisco Meeting Server Small M7 (per node)	Cisco Meeting Server 2000 M6(per node)
Individual Meeting Servers or Meeting Servers in a cluster (notes 1, 2, 3, and 4) and Meeting Servers in a Call Bridge Group	1080p30 720p30 SD Audio calls	120 240 480 3000	648 1296 1875 3200
	HD participants per conference per server		
	web app call capacities (internal calling & external calling on CMS web edge):		
	Full HD HD SD Audio calls	120 240 480 3000	648 1296 1875 1875
Meeting Servers in a Call Bridge Group	Call type supported		
	Load limit	240,000	1,296,000

Points to Note:

- Maximum of 24 Call Bridge nodes per cluster; cluster designs of 8 or more callbridge nodes need to be approved by Cisco, contact Cisco Support for more information.
- Clustered Cisco Meeting Server 2000's without Call Bridge Groups configured, support integer multiples of maximum calls, for example integer multiples of 700 HD calls.
- Up to 21,000 HD concurrent calls per cluster (24 nodes x 875 HD calls) applies to SIP or web app calls.

- A maximum of 2600 participants per conference per cluster depending on the Meeting Servers platforms within the cluster.
- Table 26 assumes call rates up to 2.5 Mbps-720p5 content for video calls and G.711 for audio calls. Other codecs and higher content resolution/framerate will reduce capacity. When meetings span multiple call bridges, distribution links are automatically created and also count against a server's call count and capacity. Load limit numbers are for H.264 only.
- The call setup rate supported for the cluster is up to 40 calls per second for SIP calls and 20 calls per second for Cisco Meeting Server web app calls.
- Up to 16,800 HD concurrent calls per cluster (24 nodes x 700 HD calls) applies to SIP or web app calls.
- Table 26 assumes call rates up to 2.5 Mbps-720p5 content for video calls and G.711 for audio calls. Other codecs and higher content resolution/framerate will reduce capacity. When meetings span multiple call bridges, distribution links are automatically created and also count against a server's call count and capacity. Load limit numbers are for H.264 only.
- Meeting Server 1000 M7(Meeting Server Small) variants support a maximum of 94 vCPU and 128 GB RAM.

E.1 Cisco Meeting Server web app call capacities

This section details call capacities for deployments using Web Bridge 3 and web app for external and mixed calling. (For internal calling capacities, see Table 26.)

E.1.1 Cisco Meeting Server web app call capacities – external calling

Expressway (Large OVA or CE1200) is the recommended solution for deployments with medium web app scale requirements (i.e. 800 calls or less). Expressway (Medium OVA) is the recommended solution for deployments with small web app scale requirements (i.e. 200 calls or less). However, for deployments that need larger web app scale, from version 3.1 we recommend Cisco Meeting Server web edge as the required solution which will scale up to SIP capacity (see Table 26).

External calling is when clients use Cisco Expressway as a reverse proxy and TURN server to reach the Web Bridge and Call Bridge.

When using Expressway to proxy web app calls, the Expressway will impose maximum calls restrictions to your calls as shown in Table 27.

Note: Deployments of Web Bridge 3 and the web app have been validated on Expressway version X14.3 or later, earlier Expressway versions may not be supported by Web Bridge 3.

Table 27: Cisco Meeting Server web app call capacities – external calling

Setup	Call Type	CE1200 Platform	Large OVA Expressway	Medium OVA Expressway
Per Cisco Expressway (X14.3 or later)	Full HD	150	150	50
	Other	200	200	50

The Expressway capacity can be increased by clustering the Expressway pairs. Expressway pairs clustering is possible up to 6 nodes (where 4 are used for scaling and 2 for redundancy), resulting in a total call capacity of four times the single pair capacity.

Note: The call setup rate for the Expressway cluster should not exceed 6 calls per second for Cisco Meeting Server web app calls.

E.1.2 Cisco Meeting Server web app capacities – mixed (internal + external) calling

Both standalone and clustered deployments can support combined internal and external call usage. When supporting a mix of internal and external participants the total web app capacity will follow Appendix E for Internal Calls, but the number of participants within the total that can connect from external is still bound by the limits in Table 27.

For example, a single standalone Meeting Server 2000 with a single Large OVA Expressway pair supports a mix of 1000 audio-only web app calls but the number of participants that are external is limited to a maximum of 200 of the 1000 total.

Appendix F Activation key for unencrypted SIP media

You have the choice of purchasing an activation key with SIP media encryption enabled or SIP media encryption disabled (unencrypted SIP media) for the Cisco Meeting Server 1000/ Small, Cisco Meeting Server 2000 and the VM software image. Choose either encrypted or unencrypted options under the software pids R-CMS-K9 and R-CMS-2K-K9. Media includes audio, video, content video and ActiveControl data.

Note: Current Call Bridge activations are unaffected, unless an activation key is uploaded with SIP media encryption disabled.

F.1 Unencrypted SIP media mode

If the activation key for " SIP media encryption disabled" is uploaded to the Meeting Server, then the following occurs:

- media sent between the Meeting Server and SIP devices is unencrypted,
- media sent over distribution links between clustered Call Bridges is unencrypted,
- call signalling remains encrypted,
- media in calls between the Meeting Server and web app, on any platform, remains encrypted,
- an error message is returned if the **sipMediaEncryption** parameter is set to anything other than **prohibited** on the following API objects:

  /calls/<call id>/participants

  /calls/<call id>/callLegs

  /callLegs/<call leg id>

  /callLegProfiles and **/callLegProfiles/<call leg profile id>**

  /callLegs/<call leg id>/callLegProfileTrace

- an error message is displayed if the **SIP media encryption** field on the the **Configuration**>**Call settings** web page of the Web Admin interface is set to anything other than **disabled**.

Note: If SIP media encryption is disabled, call signaling can still be encrypted on outbound calls, if required, by setting the **sipControlEncryption** parameter on **/outboundDialPlanRules**.

F.2 Determining the Call Bridge media mode

To determine whether the Call Bridge uses encrypted or unencrypted SIP media use the Web Admin interface, select **Configuration** > **API**, then:

1. From the list of API objects, tap the ► after **/api/v1/system/licensing**

If the **features**object **callBridgeNoEncryption**has the **status** set to **activated**then an activation key for unencrypted media is loaded on the Call Bridge. Other valid settings for the status of **callBridgeNoEncryption** are **noLicense grace** or **expired**.

callBridgeNoEncryption also has an **expiry** field in the form of a string.

Appendix G Dual Homed Conferencing

G.1 Overview

Dual homed conferencing also improves the user experience for both Lync client users and web app users in Lync scheduled meetings and in Lync drag and drop style meetings (also known as ad hoc calls). Lync participants can use drag and drop to add web app users to a Lync meeting, and can use conference controls to mute web app users or disconnect them. For web app users joining a Lync scheduled conference, they will see the video from up to five Lync participants, as well as video from the web app users. Lync users see video in a gallery format from all of the web app users, as well as the Lync users in the meeting. Both Lync users and web app users receive a full combined list of participants in the meeting.

Note: The " Add Participant" button on the Lync/Skype for Business client does not work in ad hoc dual homed conferences. Do not use the " Meet Now" button as a workaround, as this will this will leave an active call between the Meeting Server and the AVMCU.

Lync participants can also directly dial into a Meeting Server space or use drag and drop to add a Meeting Server space to a Lync meeting. These are useful if a large meeting is being held in a Cisco Meeting Server space which the Lync user wants to join. In the first case they will receive a composed layout of multiple participants. When adding a complete space to a Lync meeting, the Lync user will receive only one video stream from the space (the main speaker) and will not receive a full combined participant list. They can continue to add additional Lync participants as normal.

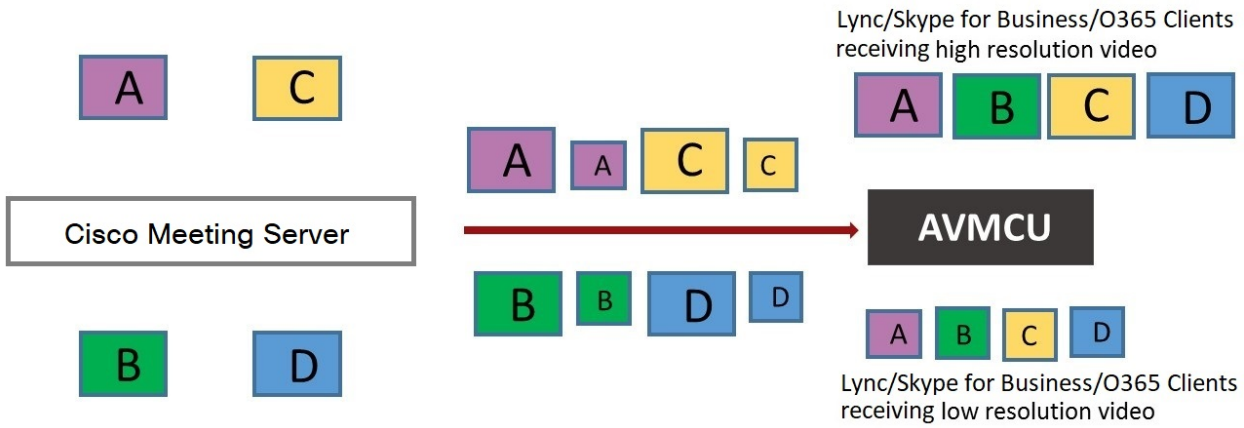
Note: Dual-homed conferences with a Meeting Server cluster are not currently supported with Expressway X8.11 as the edge for the Meeting Server, unless at least some of the Microsoft traffic flows directly between one of the Meeting Servers in the cluster and the Microsoft infrastructure (and not through Expressway). Dual-homing is supported with Expressway X8.11 as the edge for standalone Meeting Servers.

G.2 Consistent meeting experience in dual homed conferences

The Meeting Server sends two H.264 video streams stream per video participant to the AVMCU, a high resolution video stream and a low resolution video stream, see Figure 51. Lync, Skype for Business and O365 clients that support the high resolution, subscribe to and receive the high quality video stream. Clients that select a lower quality, because of bandwidth restrictions, window size, layout, CPU power or being on a mobile device, subscribe to and receive the lower quality streams, and do not reduce the video quality nor degrade the video experience for other participants.

Note: Ensure that the bandwidth of the SIP trunk is set sufficiently high to accommodate the two video streams. We recommend 8MB for LANs and 2.5MB for WANs.

Figure 51: Dual media streams to AVMCU



Note: Any devices using Microsoft RTVideo will not benefit from this feature.

G.2.1 Summary of user experiences

Dual homed conferencing combined with support for RDP and multiple video encoders, results in a richer meeting experience for both Lync and web appusers.

- Both Lync client users and web app users see familiar screen layouts.
- Both Lync client users and web app users receive a full combined list of all participants in the meeting, regardless of where they are connected.
- Lync client users see a non-square aspect ratio for video from SIP endpoints and web apps.
- Lync client users see content in a separate area of their screen rather than in the main video area.
- The Meeting Server sends video using the best quality codec supported by each participant in Lync meetings. This optimizes the experience for all Lync client users in a meeting, when a mixture of Lync client versions are used by participants.

- The Meeting Server sends two H.264 video streams stream per video participant to the AVMCU, a high resolution video stream and a low resolution video stream, to preserve the high resolution experience for clients that support it, when clients that can only support low resolution join the meeting.
- Chat works in Lync AVMCU conferences with web app users in spaces. and in direct calls between a web app user and a Lync client.

Note: For the best user experience during meetings, use Lync 2013, Skype for Business 2015 or later, which allow multiple video streams to be transmitted to the Meeting Server. This enables an endpoint or web app user connecting to the Meeting Server to view multiple Lync participants. Lync 2010 only provides a single loudest speaker stream, if the loudest speaker is on the Meeting Server side of the conference already, then web app users and SIP endpoint users will not view the Lync participants.

For more information on RDP and multiple video encoder support, see these FAQs:

- RDP support,
- multiple video encoder support.

G.3 Mute/unmute meeting controls in dual homed conferences

Version 2.4 of the Meeting Server software introduced improved mute/unmute meeting controls in dual homed conferences for:

- on-premise and Office 365 Lync/Skype for Business clients,
- end point users,
- web app users.

Note: This section assumes that muting and unmuting is enabled using the API of the Meeting Server.

Muting/unmuting:

- Lync clients can mute and unmute anyone in the dual homed conference, this means themselves and others, and they can mute and unmute the audience too.
- All endpoint users can now mute Lync clients,
- Endpoint users on the Lync side of the AVMCU can now mute and unmute themselves (self) and other endpoints (either on the Lync clients/endpoints connected to the AVMCU or on the Meeting Server side). Prior to version 2.4, only endpoint users on the Meeting Server side of the AVMCU could mute and unmute themselves (self) and others.

  - For non-ActiveControl endpoints, the Meeting Server sends DTMF key sequences for each mute and unmute, and overlays an icon on the media stream to the endpoint to indicate whether the endpoint is muted or unmuted.
  - For ActiveControl endpoints running CE 9.2.1or later software, the endpoint handles the icons and messages (the Meeting Server does not overlay icons).
- Once an ActiveControl endpoint is muted it has to be unmuted locally so as to ensure the privacy of any local conversation. For example, when a remote participant mutes an ActiveControl endpoint and then tries to unmute it, the ActiveControl endpoint will mute itself again until it is locally unmuted.
- When a remote participant tries to unmute a non-ActiveControl endpoint, the non-ActiveControl endpoint will be unmuted.
- web app users and Cisco Meeting Management users can mute and unmute Lync clients. They also see the correct mute state of all participants in the meeting.

Muting/unmuting web app **users:**

- Information on local muting and unmuting of a web app user is not passed to Lync clients in dual homed conferences. However, if a Lync client remotely mutes a web app user and the web app unmutes itself, the Meeting Server tells the Lync clients about the unmuting.
- When a remote participant tries to unmute a web app user, the web app user will remain locally muted. Note: other participants will still see them as unmuted, although they are actually muted.
- The web app shows the mute/unmute state using its own icons. Meeting Server icons are not overlaid on the web app video pane.

G.4 Configuring the Dual Homed Lync functionality

If you already have an on-prem Lync deployment or Lync Federation deployment working with the Meeting Server deployment, then no additional configuration is required on the Meeting server.

If this is a new deployment, then make sure that you configure the Lync Edge settings on the Meeting Server, see Section 1.5.

G.4.1 Troubleshooting

If users are unable to join a Lync conference via the IVR or using a dial plan rule that resolves to “Lync”, the first thing to do is to verify that the “Lync Edge” settings have been set up - the same mechanism is used to resolve Lync conferences as is used to find the Edge server. The Meeting Server must query the Lync FE server to find both of these.

If this fails, a message will be logged in the event log to say that the conference ID cannot be found:

`lync conference resolution: conference "1234" not found`

This may mean that the conference does not exist, but there are also other possible causes.

If SIP traffic tracing is enabled, there should be a 'SERVICE' message sent to the Lync FE server just before the above message is logged, which should be replied to with a 200 OK. Check that this message is sent to the correct IP, which should be that of a Lync FE server.

If this message is not sent (it does not show up in the logs), then it is possible that the Call Bridge is unable to find the Lync server using a DNS SRV lookup for the **`_sipinternaltls._tcp.lyncdomain`** record, and so does not know where to send it. Enabling DNS tracing and retrying should confirm this. However this can also happen if the Lync Edge settings have not been configured on the Meeting Server.

If the Service message is sent but the Lync server replies with "403 unauthorized", then the most likely cause of this is that the local contact domain in the outbound dial plan rule for this Lync domain is not set correctly. It should be set to the FQDN of the Meeting Server, which should be the same as the FQDN supplied in the CN of the Call Bridge's certificate.

Appendix H Using TURN servers behind NAT

The TURN server can be deployed behind a NAT, and the NAT address specified using the MMP command **`turn public-ip`**. However, due to how Interactive Connectivity Establishment (ICE) works, careful configuration of the NAT is required to ensure connectivity always works.

This appendix provides an overview of how ICE works. It explains:

- how candidates are identified,
- how connectivity is checked,
- the effect of NAT in front of the TURN server,
- how NAT affects external web app users.

Note: Issues can arise when the only available path includes both relay candidates. This requires the firewall to be correctly configured, so that all clients are able to send and receive video and audio.

H.1 Identifying candidates

ICE works by gathering a list of candidate addresses and ports, and then finding which pairs of these candidates allow media to be exchanged. When multiple candidate pairs are available then a priority scheme is used to determine which pair is used.

Typically, three candidates might exist:

1. Host candidate
2. Server Reflexive candidate
3. Relay candidate

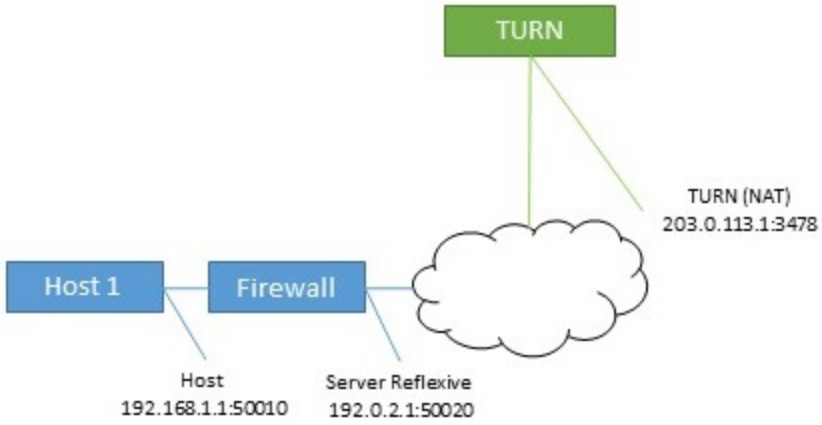
H.1.1 Host candidate

The most simple candidate is the host candidate. This is the address used by the host interface. This is often on a local network and not routable.

H.1.2 Server Reflexive candidate

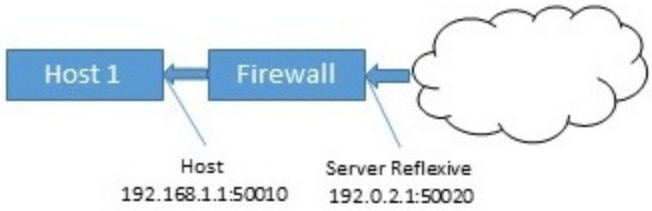
The server reflexive candidate is the address that the TURN server sees incoming packets coming from. To determine this, the host sends packets to a defined port on the TURN server (normally port 3478) and the TURN server replies with information about where the packets came from.

Figure 52: Server Reflexive candidate



In cases where the host is behind a firewall carrying out NAT, then this is different to the host candidate. In many cases, packets sent to this port and address will be forwarded back to the host.

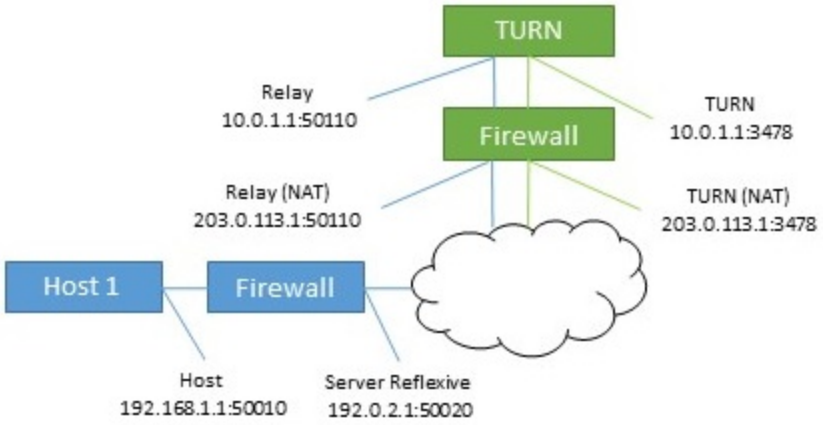
Figure 53: Effect of a host behind a firewall carrying out NAT



H.1.3 Relay candidate

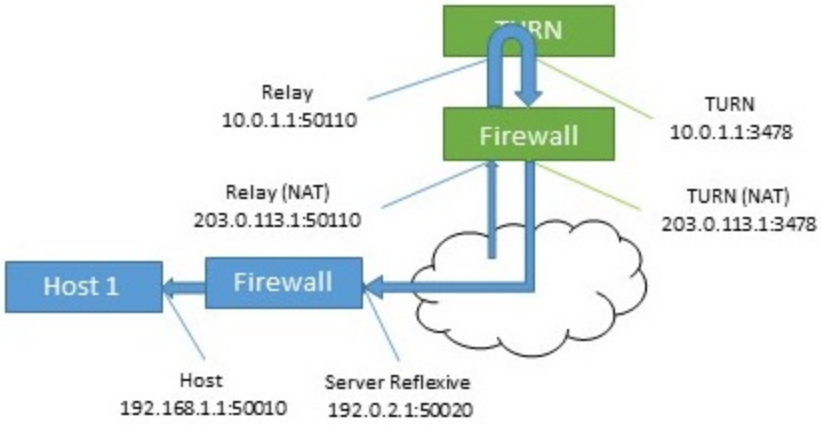
The final candidate is the relay candidate. This candidate is created by the TURN server in response to requests from the host. The relay address of this candidate is the TURN server interface address, when NAT is used the relay address is changed to an address from NAT.

Figure 54: Relay candidate



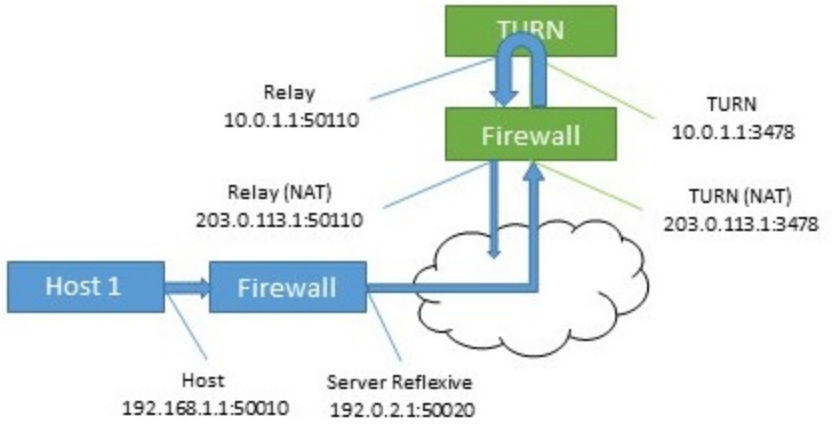
Data sent to this relay address is then sent back to the host via the TURN server.

Figure 55: TURN server returns relay address to host



This relay candidate has a second use. It can also be used by the host to send packets to the far end. This occurs when there is no other path possible. Note that these packets come from the TURN server itself, so will only get their NAT address when rewritten by the firewall.

Figure 56: Host sending packets to the far end



H.2 Checking connectivity

Once candidates are known then connectivity checks are undertaken. Each host tries to contact the far end host, server reflexive and relay addresses directly. It then also uses its relay to attempt connections to the same far end candidates.

Table 28: Candidates for two hosts (using same TURN server)

Host	Type	Address:port
1	Host	192.168.1.1:50010
1	Server Reflexive	192.0.2.1:50020
1	Relay	203.0.113.1:50110
2	Host	172.16.1.1:50100
2	Server Reflexive	198.51.100.1:50040
2	Relay	203.0.113.1:50510

Table 29: Candidate pairs formed by host 1

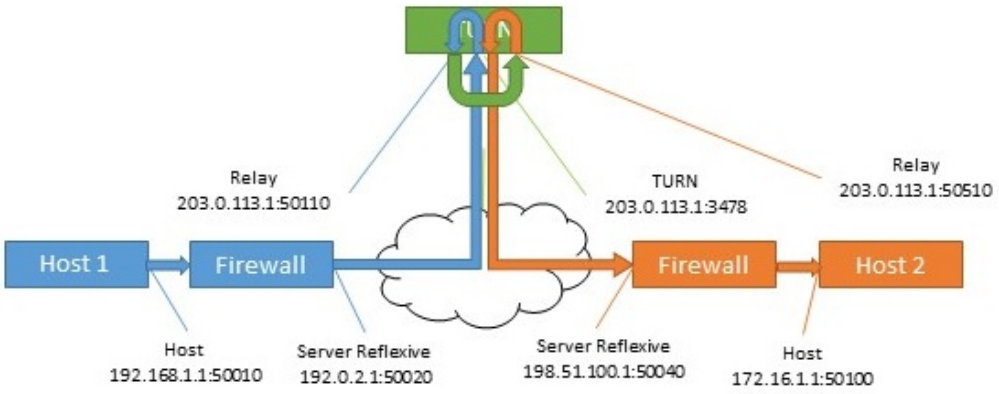
Source	Destination Type	Destination address
Host (192.168.1.1:50010)	Host	172.16.1.1:50100
Host (192.168.1.1:50010)	Server Reflexive	198.51.100.1:50040
Host (192.168.1.1:50010)	Relay	203.0.113.1:50510
Relay (10.0.1.1:50110)	Host	172.16.1.1:50100

Source	Destination Type	Destination address
Relay (10.0.1.1:50110)	Server Reflexive	198.51.100.1:50040
Relay (10.0.1.1:50110)	Relay	203.0.113.1:50510

Typically, the relay addresses are only required when the hosts have limited network access. For example, a user in a coffee shop or hotel may not be able to access any higher numbered ports.

When both hosts have restricted access then a path that involves both relay candidates can be formed. In this case, the traffic flows out of one relay candidate and into the other before being forwarded on to the far end.

Figure 57: Host to host media path using relay to relay path (no NAT)



H.3 NAT in front of the TURN server

When NAT is present in front of the TURN server, the flow becomes more complicated. The relay candidates are expecting to receive traffic from one of the other hosts candidates. If the packets are sent from the TURN server's interface, and are not rewritten by the firewall, then they will appear to be coming from an unknown address. This prevents a succesful connectivity check and in cases where the other paths are not available, there are no routes for media to take.

Figure 58: Host to host media path using relay to relay path (with NAT)

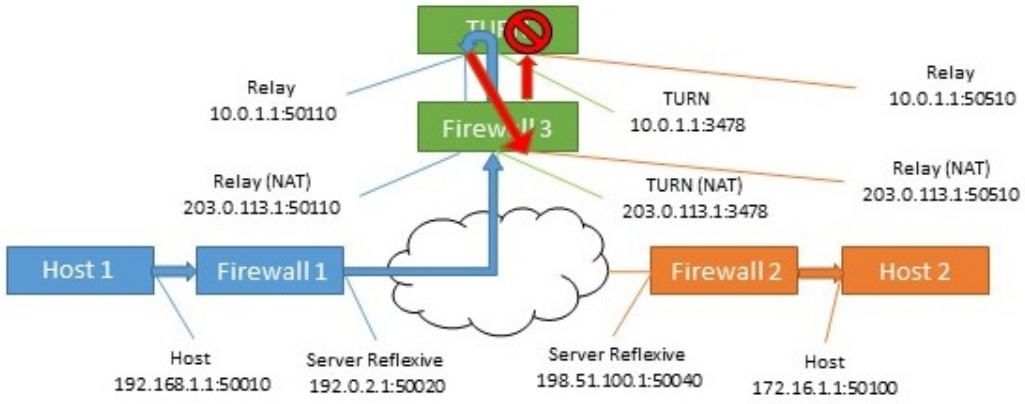


Table 30: Host to host media path using relay to relay path (with NAT)

Source address (in packets)	Destination	Action at destination
192.168.1.1:50010	203.0.113.1:3478 via Firewall	Firewall 1 rewrites source address
192.0.2.1:50020	203.0.113.1:3478	Firewall 3 rewrites destination address and forwards to the TURN server
192.0.2.1:50020	10.0.1.1:3478	TURN serevr internally maps this to the relay address for this source, and sends to far end's relay.
10.0.1.1:50110	203.0.113.1:50510 via Firewall	Firewall 3 rewrites destination address
10.0.1.1:50110	10.0.1.1:50510	TURN server sees unexpected source address and drops traffic.

The solution for this is known as hairpin NAT, loopback NAT or NAT reflection. In this the source address of the traffic is rewritten as well as the destination. The source address is then the address of the firewall, which means it matches one of the candidates.

Table 31: Host to host media path using relay to relay path (with hairpin NAT)

Source address (in packets)	Destination	Action at destination
192.168.1.1:50010	203.0.113.1:3478 via Firewall	Firewall 1 rewrites source address
192.0.2.1:50020	203.0.113.1:3478	Firewall 3 rewrites destination address and forwards to the TURN server.

Source address (in packets)	Destination	Action at destination
192.0.2.1:50020	10.0.1.1:3478	TURN server internally maps this to the relay address for this source, and sends to far end's relay.
10.0.1.1:50110	203.0.113.1:50510 via Firewall	Firewall 3 rewrites both source and destination addresses.
203.0.113.1:50110	10.0.1.1:50510	TURN server internally maps traffic from relay to assigned host.
10.0.1.1:3478	198.51.100.1:50040 via Firewall	Firewall 3 rewrites source address.
203.0.113.1:3478	198.51.100.1:50040	Firewall 2 rewrites destination address.
203.0.113.1:3478	172.16.1.1:50100	Arrives at final destination.

For details on how to enable this functionality, refer to your firewall documentation.

Appendix I Web Admin Interface – Configuration menu options

The **Configuration** tab on the Call Bridge's Web Admin interface allows you to configure the following options:

- General
- Active Directory
- Call settings
- Outbound calls and Incoming calls
- CDR settings
- Spaces
- Cluster
- API

I.1 General

Use the **Configuration > General** page to set up and configure:

- **TURN server settings**. Use these settings to allow the Call Bridge and external clients to access the TURN server. Use MMP commands to configure the TURN server itself. While you can still configure a single TURN server via the Web Admin Interface, we strongly suggest that if you have multiple TURN servers you use only the API to configure them, as described in Section 8.1.
- **Lync Edge settings**. Use these settings if you are integrating your Call Bridge with Lync Edge. See *Configuration on Meeting Server to use Lync Edge*.
- **IVR**. Use these settings if you are using an Interactive Voice Response (IVR) to manually route to pre-configured calls, so callers are greeted by a prerecorded voice message inviting them to enter the ID number of the call or space that they want to join. See *IVR configuration*.

I.2 Active Directory

If you want users to use web apps to connect to the Meeting Server, then you must have an LDAP server. The Meeting Server imports the User accounts from the LDAP server.

Note: You can use OpenLDAP and Oracle Internet Directory (LDAP version 3), however, this needs to be configured via the API—it cannot be configured through the Web Admin interface.

Use the **Configuration > Active Directory** page to set up the Meeting Server to work with Active Directory. See LDAP configuration.

I.3 Call settings

Use the **Configuration > Call settings** page to:

- Allow media encryption for SIP calls (including Lync).
- Specify whether participant label overlays are shown on SIP calls.
- Specify the preferred size (in milliseconds) for outgoing audio packets; 10ms, 20ms, or 40ms.
- Enable TIP support. (You need to enable TIP support if you use endpoints such as the Cisco CTS range.)
- Allow presentation video channel operations—if this is set to **prohibited** then no content channel video or BFCP capability will be advertised to the far end.
- If presentation video channel operations are allowed for SIP calls, this setting determines the Call Bridge's BFCP behavior, one of:
  - **server role only**—this is the normal option for a conferencing device, and is intended for use with BFCP client mode devices (for instance, SIP endpoints).
    or
  - **server and client role**—this option allows the Call Bridge to operate in either BFCP client or BFCP server mode in calls with remote devices.

  This setting allows improved presentation video sharing with a remote conference-hosting device.
- Set the value for the Resource-Priority header field in outgoing SIP calls. This setting tells the Meeting Server how much priority you will allow the bandwidth to allocate for presenting. This depends on the bandwidth capability of the network environment and other factors such as if there are any immersive systems that push HD, for example.
- Enable and disable UDP signaling for SIP. Set to one of:
  - **disabled|enabled**: disable if you use SIP over TCP, or require that all of your network traffic is encrypted.
  - **enabled, single address** mode corresponds to the SIP over UDP behavior in versions prior to 2.2 and is the default.
  - **enabled, multi address** if the Call Bridge is configured to listen on more than one interface.
- Enable Lync presence support. This setting determines whether or not this Call Bridge should supply information on destination URIs it serves to Lync presence subscribers.

- Leave the Lync packet pacing mode set to **default**. Do not change the setting to **delay** unless instructed to do so by Cisco Support.

Note: For more information on each field, you can use the hover-over text that displays for each individual field, or see *Dial plan configuration–SIP endpoints*Dial plan configuration – SIP endpoints.

The **Call settings** page also allows you to change the bandwidth settings for SIP, Cisco Meeting Server (web app), Server reflexive, Relay, VPN, and Lync content. The settings are measured in bits-per-second, for example, 2000000 is 2Mbps. We dedicate at least 64kbps for audio, and recommend 2Mbps for a 720p30 call, or around 3.5Mbps for a 1080p30 call. More bandwidth would be required for 60fps.

You may need to change some of the bandwidth settings if you allow SIP media encryption, or enable TIP support, for example. In the case of 3 screen TIP calls, the bandwidth numbers seen on the **Call settings** page get automatically tripled, so you do not need to manually set them to 6Mbps for example. However, we would normally recommend (3x) 4Mbps for most CTS calls.

I.4 Outbound calls and Incoming calls

Use the **Configuration > Outbound calls / Incoming calls** pages to determine how the Meeting Server handles each call.

The **Outbound calls** page controls how outbound calls are handled; the **Incoming calls** page determines whether incoming calls are rejected, or matched and forwarded. If they are matched and forwarded, then information about how to forward them is required. The **Incoming calls** page has two tables—one to configure matching/rejection and the other to configure forwarding behavior.

For more information on completing these fields, see *Web Admin Interface configuration pages that handle calls*.

I.5 CDR settings

Use the **Configuration > CDR settings** page to enter the URI of the CDR receivers.

The Meeting Server generates Call Detail Records (CDRs) internally for key call-related events, such as a new SIP connection arriving at the server, or a call being activated or deactivated. It can be configured to send these CDRs to a remote system to be collected and analyzed. You can not store records on a long-term basis on the Meeting Server, or browse CDRs on the Meeting Server.

For more information on completing these fields, see Call Detail Record support and the Call Details Record Guide.

You can also use the API to configure Meeting Server with the URI of the CDR receivers. See the API Reference guide.

I.6 Spaces

Use the **Configuration > Spaces** page to create a space on the Meeting Server to dial into. This allows, for example, endpoints and web app to dial in.

Add a space with:

- **Name** for example. `Call 001`
- **URI** for example. `88001`

On this page you can also optionally specify Secondary URI user part, Call ID, Passcode, and Default Layout.

You can also use the API to create spaces. See the API Reference guide.

Note: The Call ID parameter supports only a numeric value, therefore should be configured with a number.

I.7 Cluster

Note: The **Configuration > Cluster** page only appears in the Web Admin interface if all the databases are running as a cluster and all the Call Bridges have been connected to the database cluster.

Within your Meeting Server deployment, you can enable Call Bridge clustering which will allow multiple Call Bridges to operate as a single entity and scale beyond the capacity of any single Call Bridge.

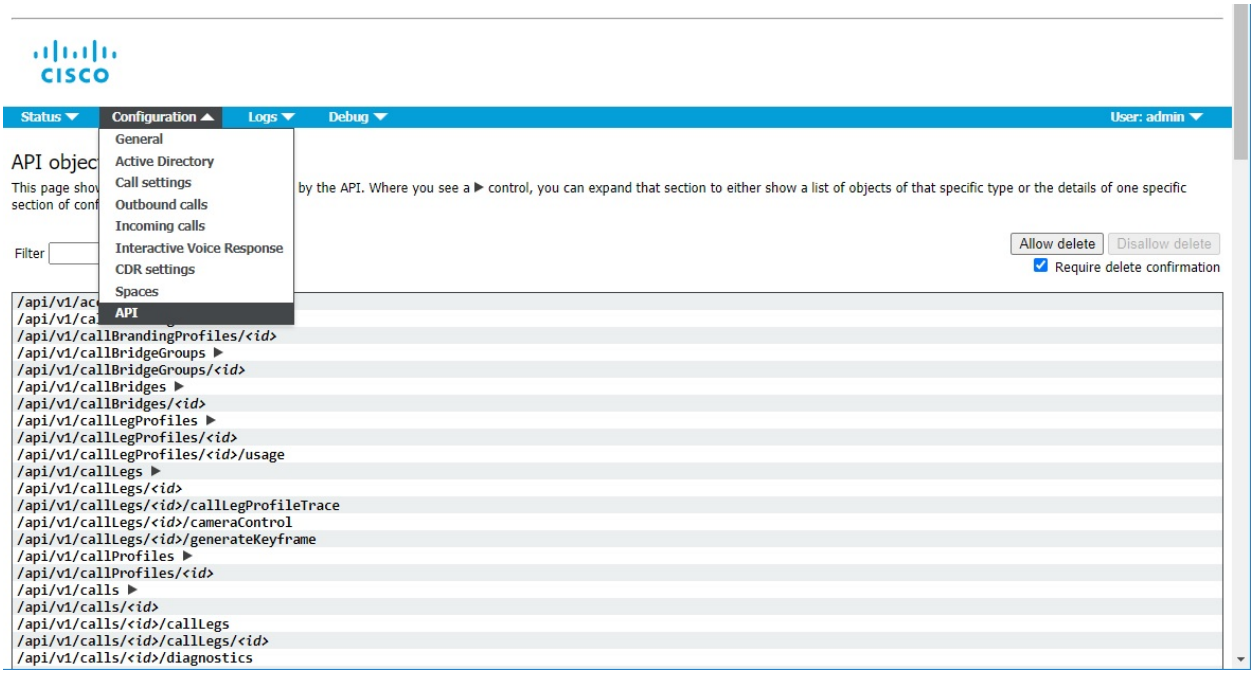
You have a choice whether to setup the Call Bridges in the cluster to link peer-to-peer, or for calls to route via call control devices between the clustered Call Bridges.

For more information, see *Clustering Call Bridges*.

I.8 API

From version 2.9, the API can be accessed using the Meeting Server Web Admin Interface rather than using API Methods and third-party applications. After logging in to the Web Admin interface, navigate to the **Configuration** tab and select **API** from the pull-down list. See Figure 59.

Figure 59: Accessing the API via the Meeting Server web admin interface



Note: To access the API via the web interface you still need to do the initial Meeting Server configuration settings and authentication using the MMP as you would if you were using a third party application.

Refer to Appendix J for examples of using the API tool through the Web Admin interface.

Appendix J API Examples

The examples in this appendix show how to use the API through the Web Admin Interface, but it is still possible to use API tools such as POSTMAN.The examples do not use all the parameters that are possible for these API methods; see the API Reference guide for additional details.

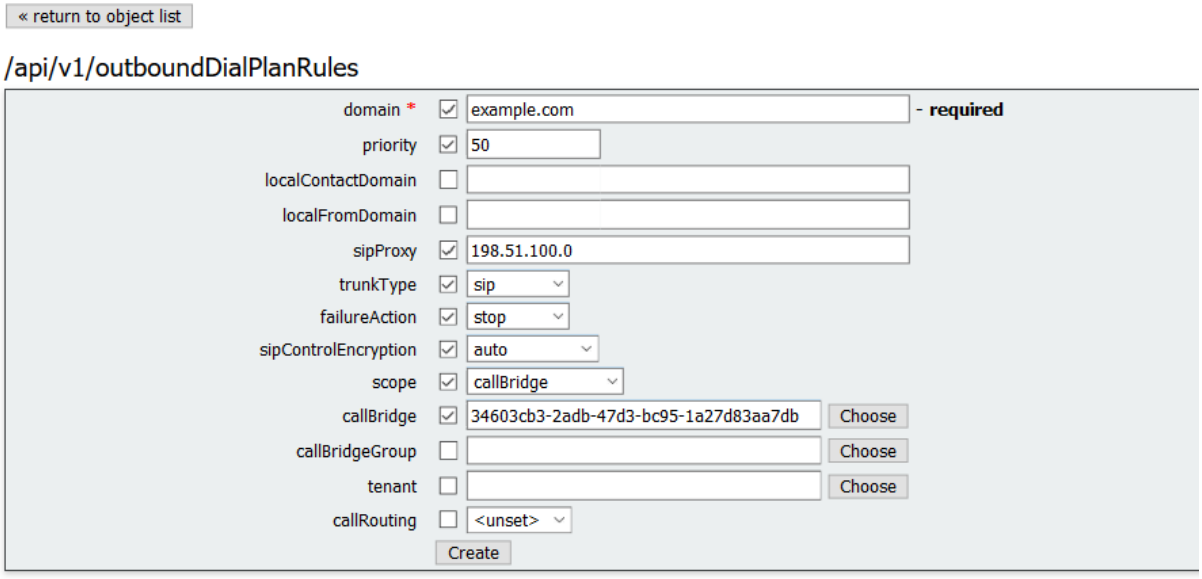
The following examples are provided:

- Creating an Outbound Dial Plan Rule for a Specific Call Bridge in a Cluster
- Setting up a Web Bridge on the Meeting Server
- Creating Web Bridge Customization on a Call Bridge
- Setting up the Turn Server and connecting to the Call Bridge
- Creating a space and adding members
- Creating Call Leg Profiles for host and guest access
- Applying Access Methods to a space

J.1 Creating an Outbound Dial Plan Rule for a Specific Call Bridge in a Cluster

Using the Web Admin interface of a Meeting Server in the cluster, select **Configuration**>**API**:

1. From the list of API objects, tap the ► after **/outboundDialPlanRules**
2. Click the **Create new** button
3. Enter the **domain** to be matched for the dial plan rule to be applied, the example below uses **example.com**
4. Set **scope** = **callBridge**, the outbound dial plan rule will only be valid for the Call Bridge selected in step 4
5. Click the **Choose** button beside the **callBridge** parameter, and click **Select** next to the Call Bridge for which the rule is valid
6. Click **Create**



The example above also sets **priority** = 50, dial plans with a priority greater than 50 will be applied before this dial plan, **sipProxy** set to the IP address of the proxy device through which to make calls, **trunkType** = sip, **failureAction** = stop to prevent the next outbound dial plan rule from being tried if this one fails to connect a call, and **sipControlEncryption** = auto to allow fall back to unencrypted control traffic in the event of encrypted control connections failing.

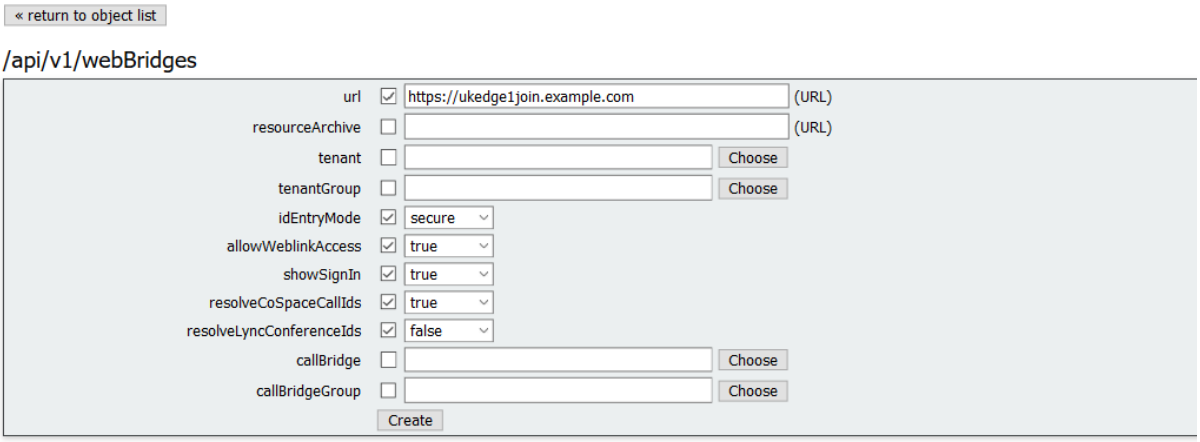
7. Verify the parameters of the outbound dial plan rule before clicking **on return to object list** button at top of page.

J.2 Setting up a Web Bridge on the Meeting Server

Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:

1. From the list of API objects, tap the ▶ after **/webBridges**
2. Click the **Create new** button
3. Enter the **url** of the Web Bridge, where ukedge1join.example.com is the FQDN used by the Call Bridge to reach the Web Bridge.

   The example below also has **idEntryMode** = secure to require both the call id and passcode to be entered before joining a space, **allowWebLinkAccess** = true so that the Web Bridge allows guests to access spaces by following a weblink, **showSignIn** = true so that the Web Bridge displays the Sign In button, **resolveCoSpaceCallIds** = true to allow the Web Bridge to accept call ids of a space and space access method for the purpose of allowing a visitor to join a space, and **resolve LyncConferenceIds** = false to prevent call ids being resolved to scheduled Lync/Skype for Business call ids.

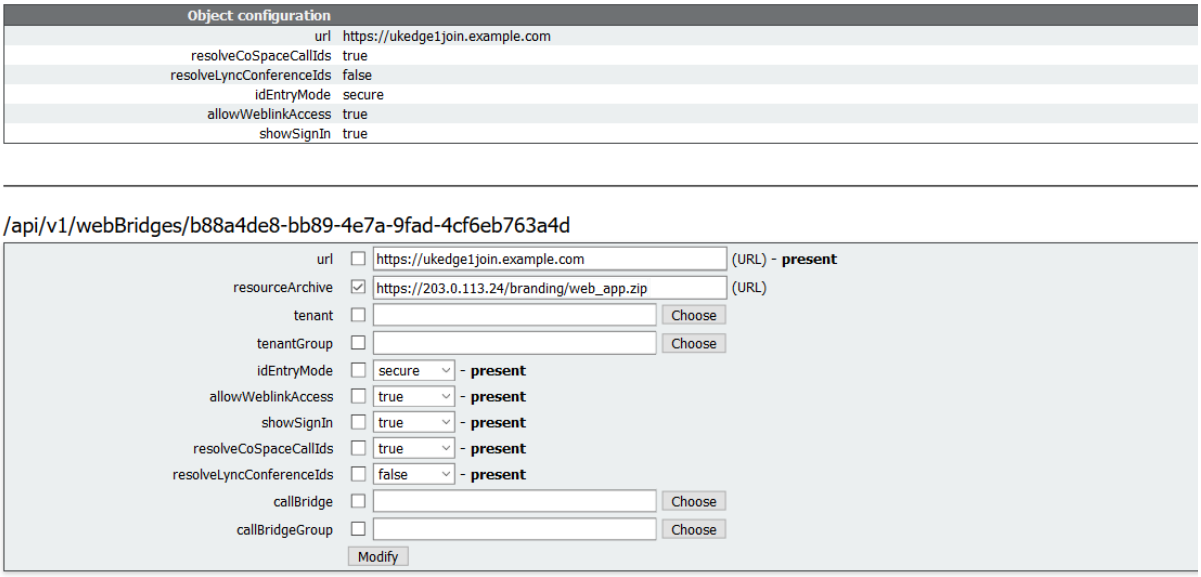


4. Click **Create**.

J.3 Creating Web Bridge Customization on a Call Bridge

Using the the WebBridge configured in the previous example:

1. Click **Modify** and enter the url for the **resourceArchive**, the url is where you have archived the customization file which the Call Bridge will use to customize elements of the web app (sign-in background image, icon displayed, text below icon and text on browser tab).



Note: For more information on customizing the web app, refer to the Cisco Meeting Server Customization Guidelines.

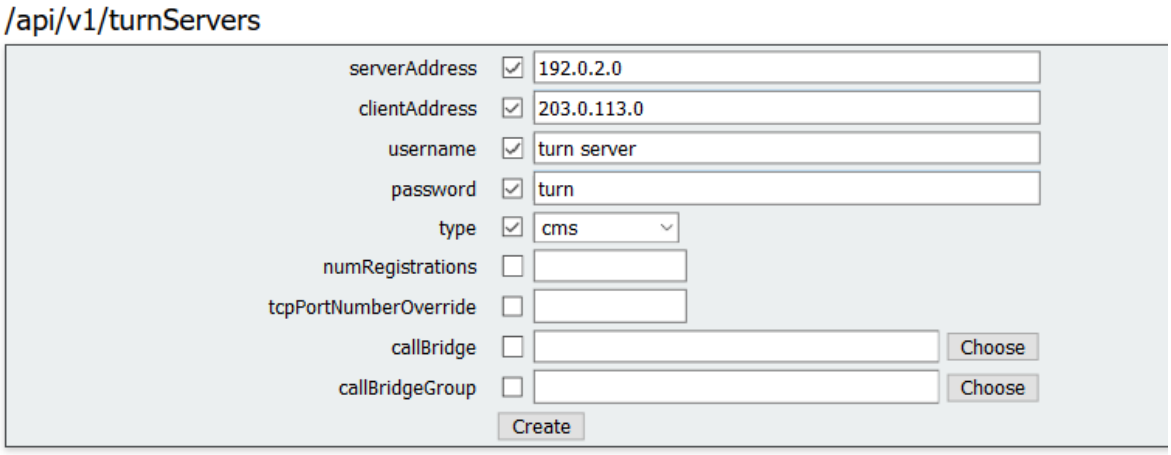
J.4 Setting up the TURN Server and connecting to the Call Bridge

Note: The TURN Server is not available on the Cisco Meeting Server 2000. It is more suited to the lower capacity Cisco Meeting Server 1000/ Small and specification-based VM servers.

Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:

1. From the list of API objects, tap the ▶ after **/turnServers**
2. Click the **Create new** button
3. Enter the **server address**, **username**, **password**, **type**, and **client address** if using the web app, and any other parameters as appropriate, where:

   server address = the address that the Call Bridge will use to reach this TURN server,
   username = the username to use when making allocations on this TURN server,
   password = the password to use when making allocations on this TURN server,
   type = **cms**, selects UDP/TCP port 3478 to connect to the server,
   client address = the address that web apps should use to reach this TURN server.



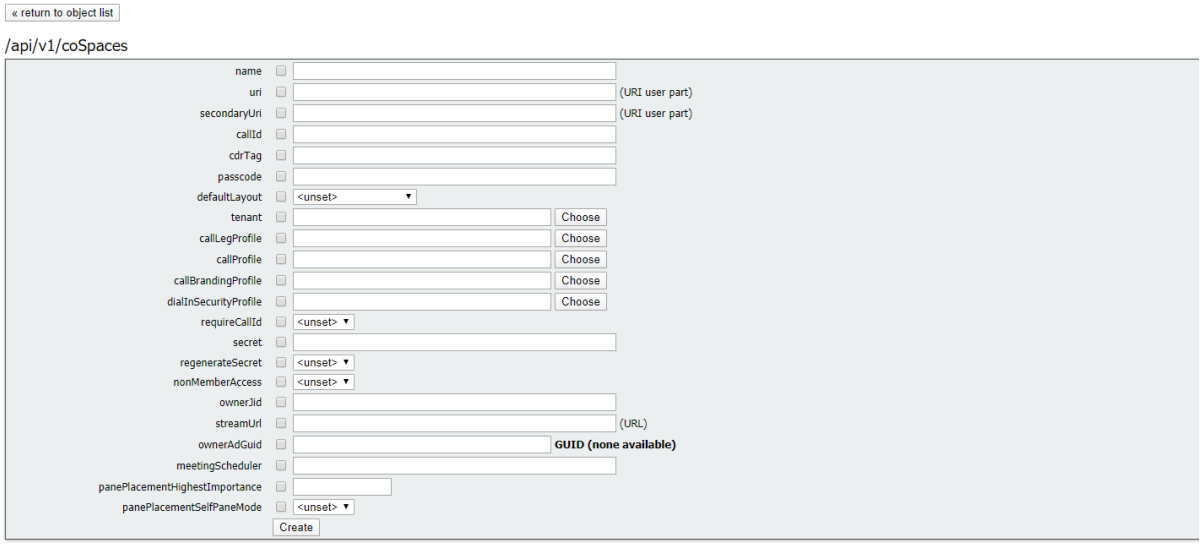
4. Click **Create**
5. Verify the parameters of the TURN server before clicking **on return to object list** button at top of page.

J.5 Creating a space and adding members

Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:

1. From the list of API objects, tap the ▶ after **/coSpaces**
2. Click the **Create new** button

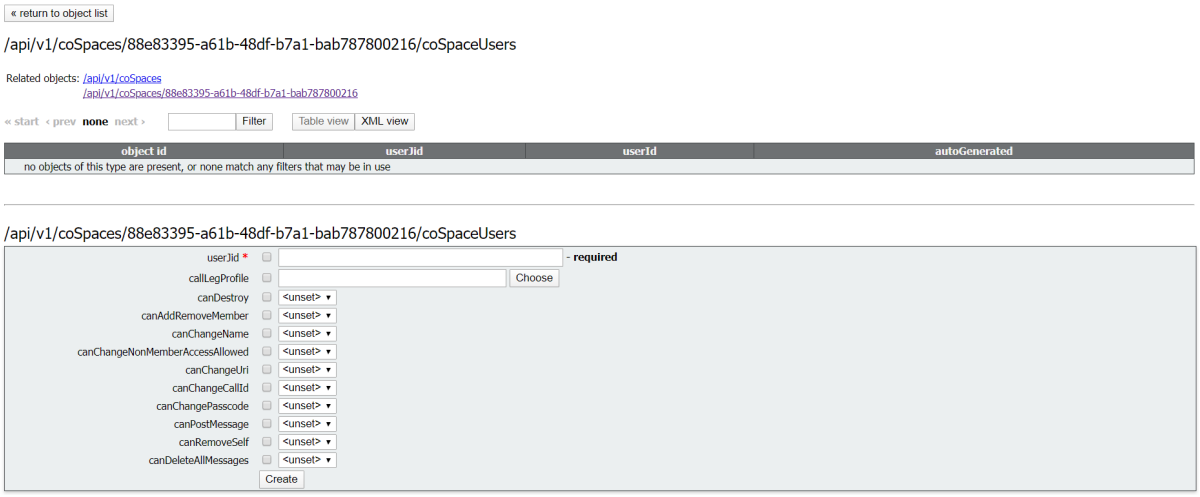
3. Enter the **name**, **uri**, **secondary uri**, **call id**, **passcode** and any other parameters as appropriate



4. Click **Create**.

J.5.1 Adding Members to the space

1. From the list of **Related objects** at the top of the page click **/api/v1/coSpaces/...../coSpaceUsers**
2. Enter the **userJid** and any other parameters as appropriate. The userJid is the identifier for the user, for example **first.last@example.com**.



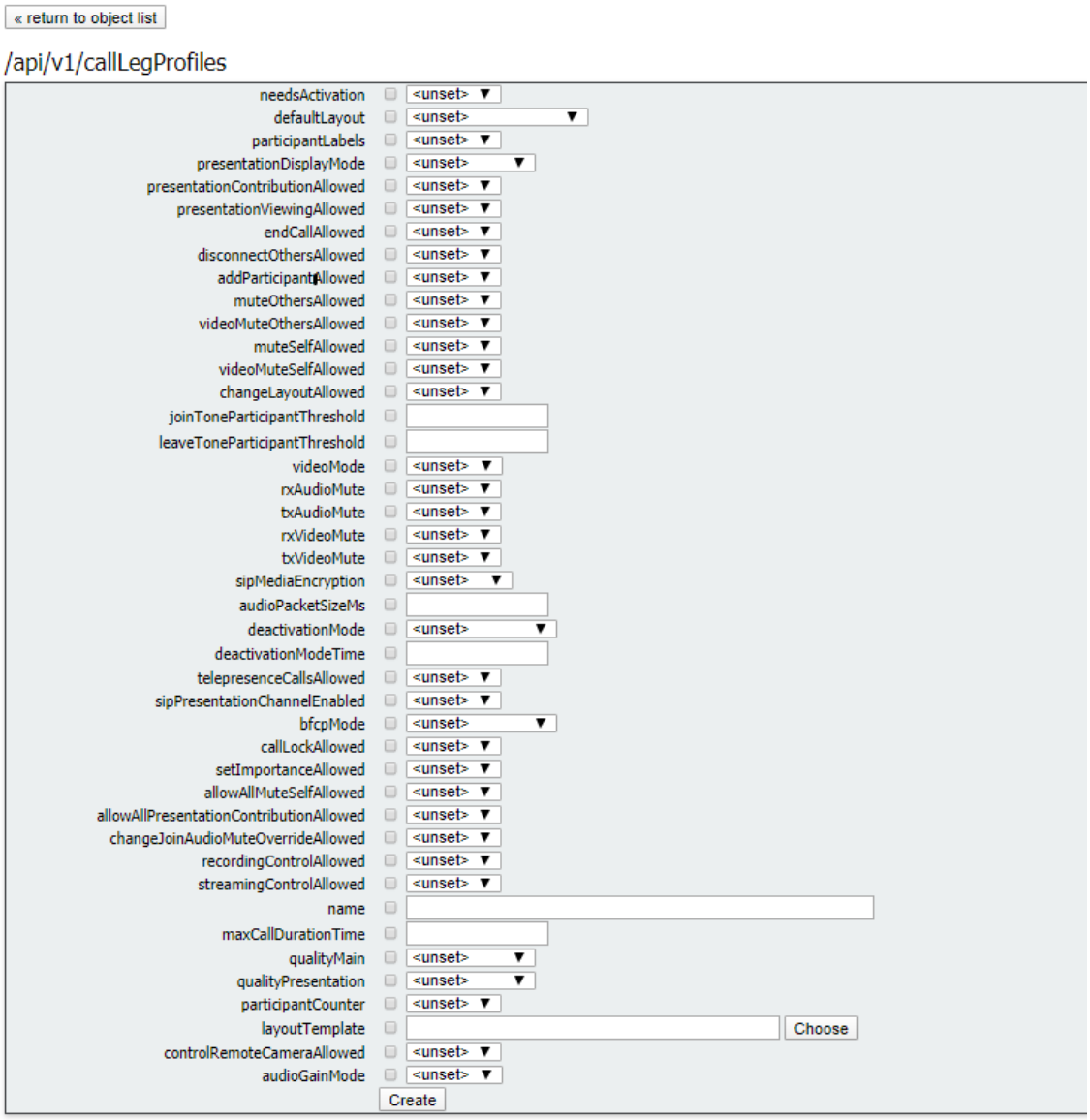
3. Click **Create**
4. Repeat for all other Members that need adding to the space.

J.6 Creating Call Leg Profiles

This example creates two callLegProfiles, one for hosts and the other for guests.

Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:

1. From the list of API objects, tap the ▶ after **/callLegProfiles**
2. Create a callLegProfile for a host
   a. Click the **Create new** button
   b. Set the parameter **needsActivation** = **false**, and **defaultLayout**=**allEqual**



   c. Click **Create**

3. Create a callLegProfile for a guest

a. Click the **Create new** button

b. Set the parameter **needsActivation** = **true**, **defaultLayout**=**speakerOnly**, **deactivationMode**=**disconnect** and **deactivationModeTime**=**10**

Note: Guests will be disconnected automatically 10 seconds after the host leaves the meeting.

These host and guest call leg profiles can be applied when you create access methods for a space, see next example.

J.7 Applying Access Methods to a space

This example explains how to apply different access methods to a space for host access and guest access.

Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:

1. From the list of API objects, tap the ▶ after **/coSpaces**
2. Either click on the **object id** of an existing space or create a new one
3. From the list of **Related objects** at the top of the page click **/api/v1/coSpaces/...../accessMethods**
4. Create an accessMethod for hosts using this space
   a. Set parameter **name** = **host**, **callID**=**12345678**, click on the **Choose** button beside the **callLegProfile** parameter and select the call leg profile created in step 2 in the previous example
   b. Enter other parameters as appropriate
   c. Click **Create**
5. Create an accessMethod for guests using this space
   a. Set parameter **name** = **guest**, **callID**=**87654321**, click on the **Choose** button beside the **callLegProfile** parameter and select the call leg profile created in step 3 in the previous example
   b. Enter other parameters as appropriate
   c. Click **Create**
6. Test this configuration; you should see the following behavior.
   - All guests join the space by dialing 87654321
   - Hosts can join the spaceby dialing 12345678.

- All guests cannot see any video steams or hear any audio until a host joins.
- All guests are disconnected in 10 seconds after the last host leaves the meeting.

Cisco Legal Information

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

© 2025 Cisco Systems, Inc. All rights reserved.

Cisco Trademark

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)