Cisco Meeting Server

Cisco Meeting Server Release 2.9

Scalability & Resilience Server Deployment Guide

December 02, 2020

Contents

What's new

Version	Change
December 01, 2020	Added section: Ports open on loopback
September 09, 2020	Addition of using MMP command "webbridge url-redirect" as an alternative in deploying multiple Web Bridge 2s.
August 20, 2020	Graphics added to API Examples appendix.
June 17, 2020	Cisco Meeting Server web app call capacity figures added. Important note for Expressway users deploying Web Bridge 3 updated.
Jun 1, 2020	Some examples changed to reflect using Web Admin interface to access API.
April 27, 2020	Web Bridge information edits.
April 08, 2020	New version for 2.9. Introduction of Web Bridge 3, support for third-party SIP recorder.
January 21, 2020	Removed duplication in text
November 22, 2019	Moved Dial Transform overview from API guide to this guide.
November 13, 2019	New version for 2.8. Call capacity information updated.
September 30, 2019	Minor correction.
August 15, 2019	Minor correction.
August 12, 2019	New version for 2.7.
July 19, 2019	Minor correction.
June 24, 2019	Minor corrections.
June 03, 2019	Minor corrections.
April 26, 2019	New version for 2.6. Added information on license changes and support for Skype for Business 2019.

Version	Change
March 19, 2019	Added note on need for each participant in dual homed conference to have unique "from:" SIP address.
February 18, 2019	Minor corrections to WebRTC app and Web Bridge section.
January 31, 2019	Clarification added to Streamer component support information.
January 29, 2019	Recording indicator for dual homed confererences section updated.
January 08, 2019	Minor corrections to appendix on scaling deployments
January 07, 2019	Miscellaneous correction
January 02, 2019	New version for 2.5. Added information on new browser support for WebRTC app, hosting branding files on a local server and new appendix on the growth in scaling deployments.
November 21, 2018	Miscellaneous corrections
October 30, 2018	Miscellaneous corrections
October 02, 2018	Added sections on setting recorder resolution and recording indicator for dual homed conferences.
October 01, 2018	New version for 2.4. Announcing removal of H.323 Gateway, SIP Edge, TURN Server, XMPP Server and Load Balancer components in a future version of the Meeting Server software.
Version	Change
Jun 1, 2020	Some examples changed to reflect using Web Admin interface to access API.
April 27, 2020	Web Bridge information edits.
April 08, 2020	New version for 2.9. Introduction of Web Bridge 3, support for third-party SIP recorder.
January 21, 2020	Removed duplication in text
November 22, 2019	Moved Dial Transform overview from API guide to this guide.

Version	Change
November 13, 2019	New version for 2.8. Call capacity information updated.
September 30, 2019	Minor correction.
August 15, 2019	Minor correction.
August 12, 2019	New version for 2.7.
July 19, 2019	Minor correction.
June 24, 2019	Minor corrections.
June 03, 2019	Minor corrections.
April 26, 2019	New version for 2.6. Added information on license changes and support for Skype for Business 2019.
March 19, 2019	Added note on need for each participant in dual homed conference to have unique "from:" SIP address.
February 18, 2019	Minor corrections to WebRTC app and Web Bridge section.
January 31, 2019	Clarification added to Streamer component support information.
January 29, 2019	Recording indicator for dual homed confererences section updated.
January 08, 2019	Minor corrections to appendix on scaling deployments
January 07, 2019	Miscellaneous correction
January 02, 2019	New version for 2.5. Added information on new browser support for WebRTC app, hosting branding files on a local server and new appendix on the growth in scaling deployments.
November 21, 2018	Miscellaneous corrections
October 30, 2018	Miscellaneous corrections
October 02, 2018	Added sections on setting recorder resolution and recording indicator for dual homed conferences.

Version	Change
October 01, 2018	New version for 2.4. Announcing removal of H.323 Gateway, SIP Edge, TURN Server, XMPP Server and Load Balancer components in a future version of the Meeting Server software.

1 Introduction

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

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 one Call Bridge goes down, meetings can be hosted on the other(s).
- Scalability (increased capacity); for example, one meeting can be hosted across 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 .

When deploying for scalability and resilience, use the API rather than the Web Admin Interface when possible: in some cases it is a necessity, and instructions are provided in this document.

- Install the Meeting Server following the appropriate Installation Guide for the deployment.
- Each Meeting Server can host all the components ("combined deployment") or be an Edge or Core server (part of a “split deployment”).

> **Note:** Acano X1 servers are suitable as Edge servers or standalone database servers.

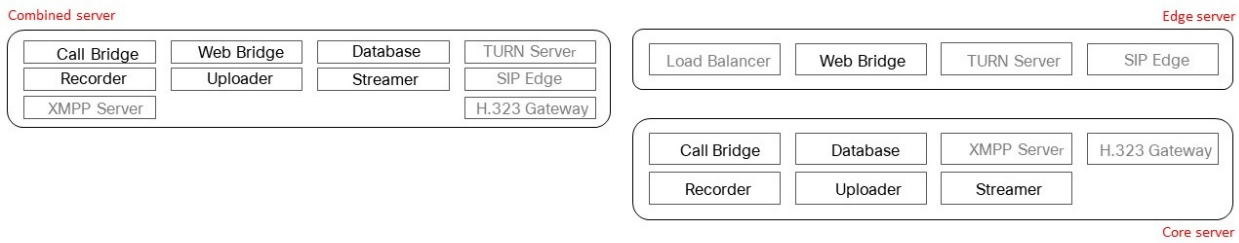
Note: Although this guides covers the SIP edge and TURN server components within the Meeting Server, customers are encouraged to start planning their transition to using Cisco Expressway at the edge of their network and the Meeting Server in the core of their network. The SIP edge, TURN server, internal Firewall and H.323 gateway components will be removed from the Meeting Server software at some point in the future.

In addition, in the future, the Cisco Meeting WebRTC App and Cisco Jabber will be the supported apps to join Meeting Server hosted conferences, in addition to SIP endpoints, and Lync/Skype for Business clients in dual homed conferences. On withdrawal of the native Cisco Meeting Apps, the XMPP server and Load Balancer components will be removed from the Cisco Meeting Server software.

Note: The components greyed out in the figure below will be removed from the Meeting Server software at some point in the future.

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

Figure 1: Combined vs. split Core & Edge deployments



1.1 Using the Cisco Expressway-E as the edge device in Meeting Server deployments

Cisco Expressway software edge features have been developed to enable the Cisco Expressway-E to be used as the edge device in Meeting Server deployments with small to medium web app scale requirements. Use the TURN server capabilities in Cisco Expressway-E to enable:

- participants using the browser based Meeting Server web app to join conferences hosted on the Meeting Server,
- remote Lync and Skype for Business clients to join conferences hosted on the Meeting Server.

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
If you are deploying Web Bridge 3 and web app you must use Expressway version X12.6 or later, earlier Expressway versions are not supported by Web Bridge 3. If you are deploying solely Web Bridge 2 and Meeting App for WebRTC you can continue to use Expressway versions earlier than X12.6.

If you wish to deploy Web Bridge 3 and use Cisco Meeting Server web app, see Appendix K.

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.

Note: Cisco Expressway-E can not be used to connect remote Cisco Meeting App thick clients (Windows/Mac desktop or iOS) to conferences hosted on the Meeting Server. Nor can the Cisco Expressway-E 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: 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 (X12.6)

Table 2: Expressway edge support for the Meeting Server

Cisco Expressway-E version	Edge feature	Meeting Server version
X12.6	Supports Cisco Meeting Server web app. See Cisco Expressway Web Proxy for Cisco Meeting Server (X12.6)	2.9 and later

Cisco Expressway-E version	Edge feature	Meeting Server version
X8.11	Supported: - load balancing of clustered Meeting Servers, - Microsoft clients on Lync or Skype for Business infrastructure in other organizations, or Skype for Business clients on Office 365 (not "consumer" versions of Skype). - interoperability between on-premise Microsoft infrastructure and on-premise Meeting Server, **where no Microsoft calls traverse into or out of the organization.** - standards based SIP endpoints. - standards based H.323 endpoints. - Cisco Meeting App thin client (Web RTC app) using TCP port 443. Not supported: - off premise Cisco Meeting App thick clients (Windows/Mac desktop or iOS). - interoperability between on-premise Microsoft infrastructure and on-premise Meeting Server **where Microsoft calls traverse into or out of the organization**, in this scenario, the Meeting Server must use the Microsoft Edge server to traverse Microsoft calls into and out of the organization. See Cisco Meeting Server with Cisco Expressway Deployment Guide (2.4/X8.11.4).	2.4 to 2.8
X8.10	Supported: - Microsoft clients on Lync or Skype for Business infrastructure in other organizations, or Skype for Business clients on Office 365 (not "consumer" versions of Skype), - standards based SIP endpoints, - Cisco Meeting App thin client (Web RTC app) using UDP port 3478 to connect to the Meeting Server via the Expressway reverse web proxy. Not supported: - load balancing of clustered Meeting Servers, - off premise Cisco Meeting App thick clients (Windows/Mac desktop or iOS) or Cisco Meeting App thin client (Web RTC app) using TCP port 443, - interoperability between on premises Microsoft infrastructure and Meeting Server; in this scenario, the Meeting Server must use the Microsoft Edge server to traverse Microsoft calls into and out of the organization. See Cisco Expressway Web Proxy for Cisco Meeting Server (X8.10)	2.3

Cisco Expressway-E version	Edge feature	Meeting Server version
X8.9	Supported: - Microsoft clients on Lync or Skype for Business infrastructure in other organizations, or Skype for Business clients on Office 365 (not "consumer" versions of Skype), - standards based SIP endpoints. Not supported: - load balancing of clustered Meeting Servers,, - off-premise Cisco Meeting App thick clients (Windows/Mac desktop or iOS) and Cisco Meeting App thin client (WebRTC app), - interoperability between on premises Microsoft infrastructure and Meeting Server; in this scenario, the Meeting Server must use the Microsoft Edge server to traverse Microsoft calls into and out of the organization See Cisco Expressway Options with Meeting Server and/or Microsoft Infrastructure	2.2

You are encouraged to migrate your Meeting Server deployments from using the Meeting Server edge components to using the Expressway X8.11 (or later) TURN server. The SIP edge, TURN server, internal Firewall and H.323 gateway components will be removed from the Meeting Server software at some point in the future

1.2 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.

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 2.9, Load Balancing Calls Across Cisco Meeting Servers
SIP Registrar	Cisco Expressway-E and Expressway-C Basic Configuration (X12.6)

Feature	Configuration covered in this guide
H.323 Gatekeeper	Cisco Expressway-E and Expressway-C Basic Configuration (X12.6)

1.2.1 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.

1.2.2 Supported deployments

Figure 2 and Figure 3 illustrate recommended Meeting Server deployments.

Both deployments show an Expressway pair (Expressway-C and Expressway-E) being used as the edge device for the Meeting Server. The Expressway-E is located in the DMZ, while the Expressway-C is located in the internal network between the Meeting Server and Cisco Unified Communications Manager.

The Cisco Meeting WebRTC App can connect via the TURN server on the Expressway-E , or via the TURN server on the Meeting Server if already configured for native Cisco Meeting Apps (Windows, Mac and iOS). Native Cisco Meeting Apps need to connect via the XMPP/Load Balancer components of the Meeting Server, as the Expressway does not support XMPP.

Figure 3 illustrates Microsoft infrastructure added to the deployment to support dual homed conferencing.

Figure 2: Cisco Unified Communications Manager-centric deployment example

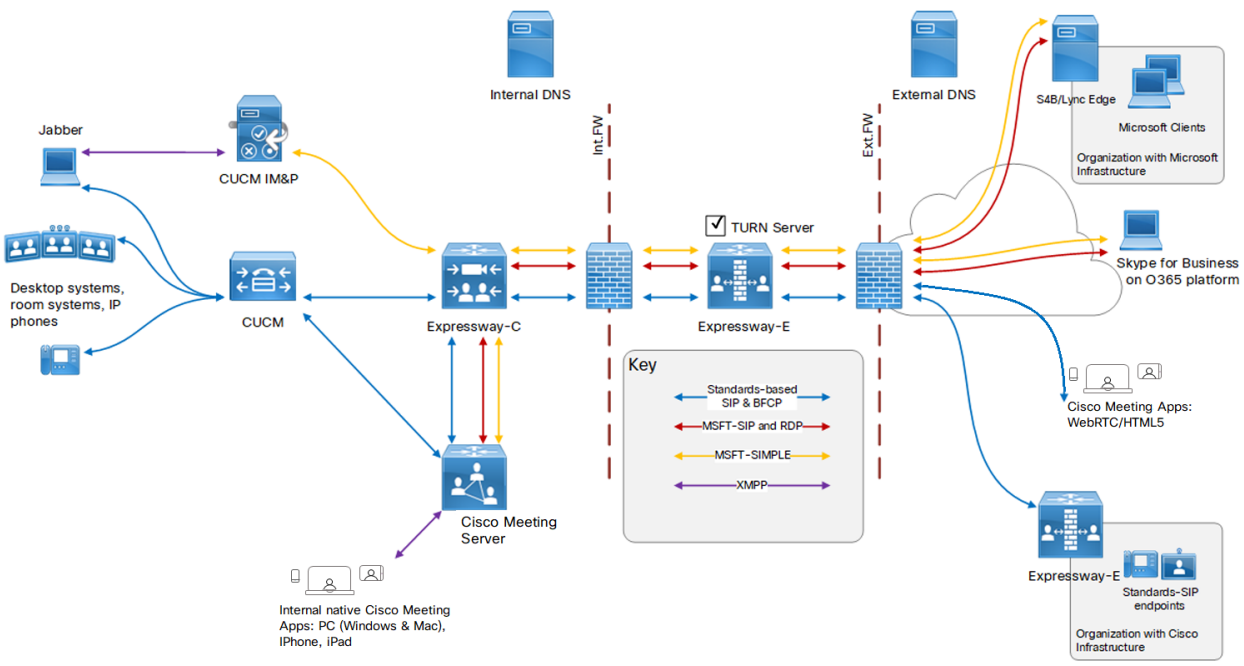
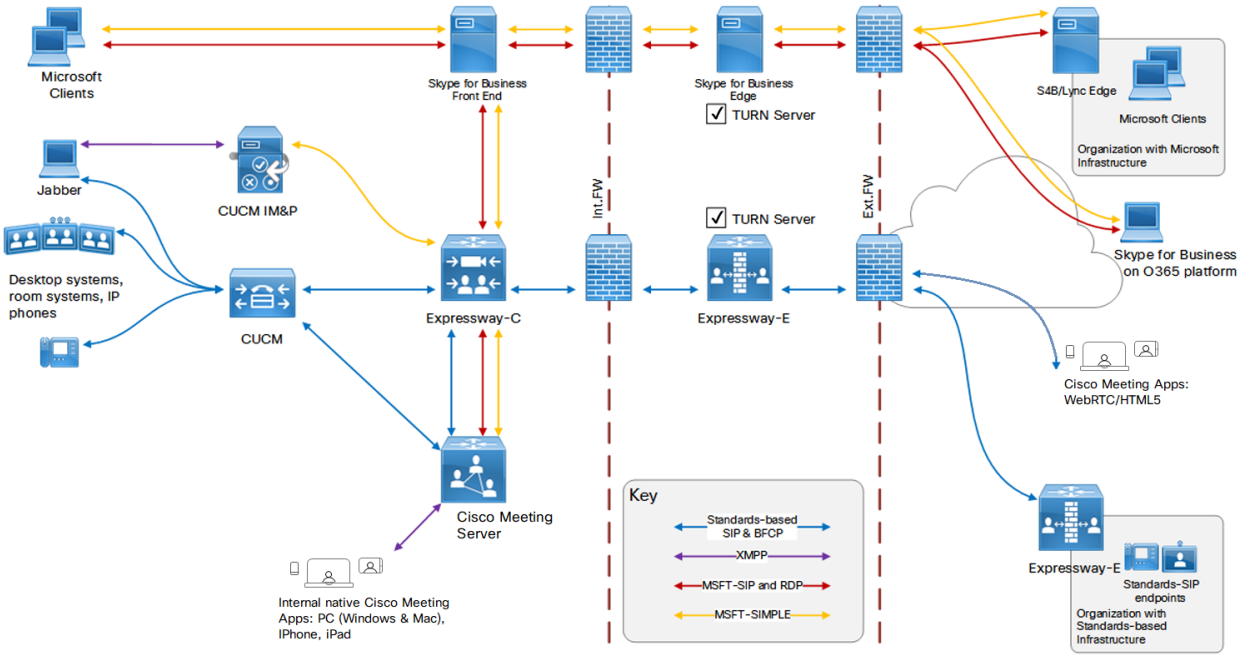


Figure 3: Cisco and Microsoft Infrastructure on-premises deployment example



Note: When planning the dial plan on Expressway, each Meeting Server in a cluster requires its own neighbor zone on the Cisco Expressway. For more information see Appendix A in the white paper Load Balancing Calls Across Cisco Meeting Servers.

1.2.3 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.

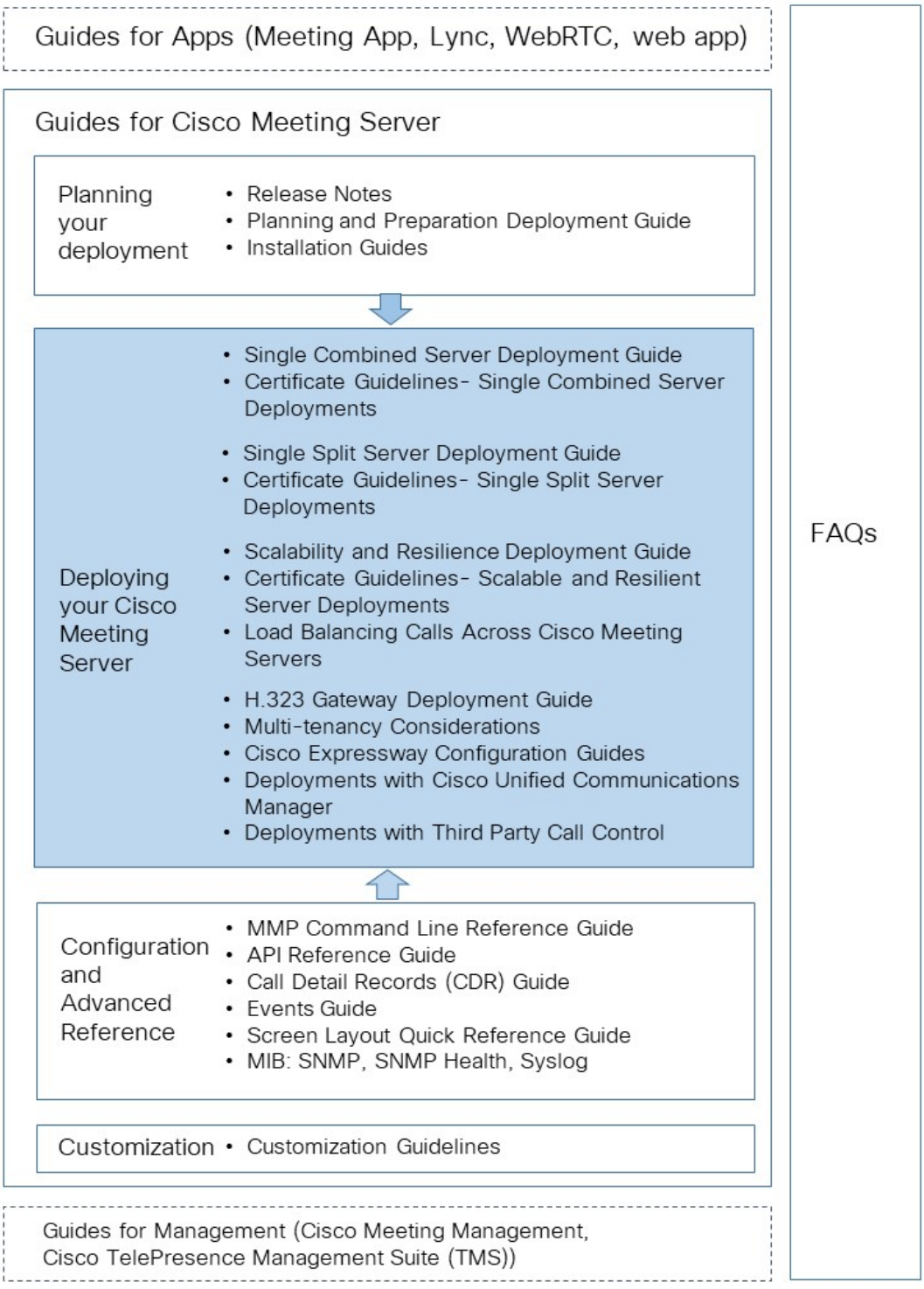
1.3 How to use this guide

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 has been renamed space.

Figure 4: 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.

1.3.1 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.

1.4 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 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.

  Note: Prior to version 2.9 software you need to configure multiple Call Bridges using the API and third party API tools, such as POSTMAN; only use the Web Admin interface for configuring a single Call Bridge.

1.4.1 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 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.
X-series servers	Serial console port or using SSH on the ethernet interface labeled Admin	Five physical ethernet interfaces labeled Admin, A, B C and D. There is no physical separation between the media interfaces A-D on an X-series server, but the Admin interface is physically separate. Each interface is configured independently at the IP level. IP forwarding is not enabled in either the Admin or host IP stack. **CAUTION:** Do not set the Web Admin to listen on the Admin interface. If you do, it may cause out of memory problems if there are a lot of web/API requests, resulting in various processes such as syslog or web proxy being killed to maintain core functionality. If this happens you will see a loss of Syslog messages and Web Admin access. See this FAQ.

1.4.2 New tools to ease configuring Meeting Server

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

- Installation Assistant (from version 2.8). Simplifies the creation of a simple Cisco Meeting Server installation for demonstrations, lab environments, or as the starting point for basic installations.
- 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 X.X Single Server Simplified Deployment guide. It is a standalone tool that uses a browser interface 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 must be run on a computer separate from Meeting Server. Refer to the Installation and Configuration guide for Installation Assistant 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.

Note: For Cisco Meeting App, guest access alone or optionally imported LDAP users can be enabled.

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 X-series products or the Cisco Meeting Server 2000 platform.

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

From version 2.9, Cisco Meeting Management connected to a Meeting Server or Meeting Server cluster, provides the facility to provision LDAP authenticated Cisco Meeting App users and 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.

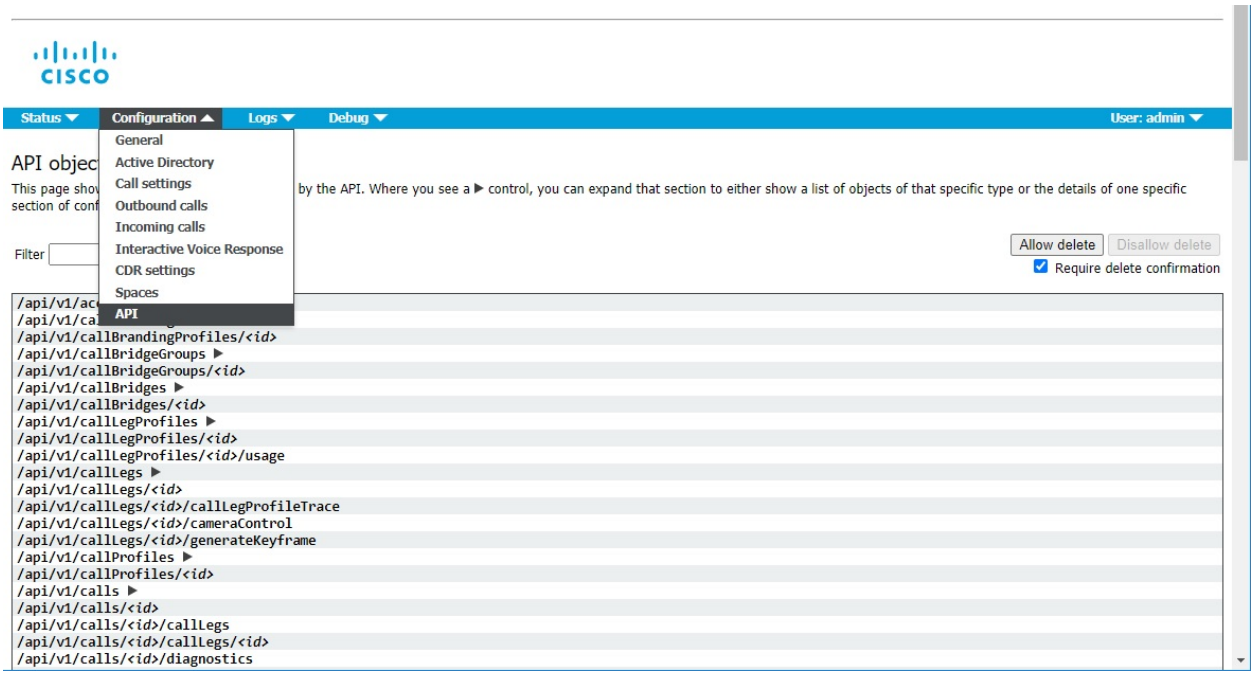
Refer to the Cisco Meeting Management User Guide for Administrators for information on connecting LDAP servers to Meeting Server clusters, how to add one or more user imports, how to create a space template, reviewing and committing the changes and finally running the LDAP sync.

API access on the web interface

To simplify using the API without the need for third-party applications, version 2.9 a user interface for the API that can be accessed via the **Configuration** tab of the Meeting Server web interface, as shown in Figure 5.

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 5: Accessing the 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.

1.5 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.

1.5.1 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.

1.5.2 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.

1.6 Cisco licensing

You need licenses for the Cisco Meeting Server. For information on purchasing and assigning licenses, see Section 1.6.3 and Section 1.6.4.

1.6.1 Cisco Meeting Server licensing

The following features require a license installed on the Meeting Server before they can be used:

- Call Bridge
- Call

- Recording
- Streaming

From version 2.4, you no longer need to purchase a branding license to apply single or multiple branding to the WebRTC app login page, IVR messages, SIP or Lync call messages or invitation text.

The XMPP activation key is included in the Cisco Meeting Server software.

In addition to feature licenses, user licenses also need to be purchased, there are 3 different types of user licenses:

- PMP Plus,
- SMP Plus,
- ACU

For information on user licensing, see Section 1.6.2.

Note: From version 2.4, 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, Cisco Meeting Server and the VM software image. For more information on the unencrypted SIP media mode and activation key see Appendix F.

Note: You need a license file for each individual Call Bridge, however, you can share licenses 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, ACU, Recording/Streaming.

If a license is only installed on one Call Bridge in the cluster then the feature(s) will only work for calls on that Call Bridge. The feature will fail for calls on other Call Bridges in the cluster, unless you share the license with the other Call Bridges. Note that each Call Bridge in the cluster requires its own license file.

For more information, see the Appendix "Sharing Call Bridge licenses within a cluster".

1.6.2 Cisco user licensing

Cisco Multiparty licensing is the primary licensing model used for Cisco Meeting Server; Acano Capacity Units (ACUs) can still be purchased, but cannot be used on the same Call Bridge as Multiparty licenses. Contact your Cisco sales representative if you need to migrate ACUs to Multiparty licenses.

Multiparty licensing is available in two variations: Personal Multiparty Plus (PMP Plus) licensing, which offers a named host license, and Shared Multiparty Plus (SMP Plus) licensing, which offers a shared host license. Both Personal Multiparty Plus and Shared Multiparty Plus licenses can be used on the same server.

Note: You can determine the number of distinct calls across a cluster of Call Bridges using the parameter **weightedCallsActive** on API object **/system/multipartyLicensing**. The sum of weighted calls across a cluster matches the number of distinct calls on the cluster. For example, if CMS1 shows 3 **callsActive** and 2 **weightedCallsActive**, and CMS2 shows 2 **callsActive** and 1 **weightedCallsActive**, then there are 3 conferences in total on the cluster and 3 PMP Plus/SMP Plus licenses are required.

1.6.2.1 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 (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 1.6.5.

1.6.2.2 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.

1.6.2.3 Cisco Meeting Server Capacity Units

Acano Capacity Units (ACUs) have been renamed Cisco Meeting Server Capacity Units. Each Capacity Unit (CU) supports 12 audio ports or the quantity of concurrent media streams to the Cisco Meeting Server software shown in Table 5.

Table 5: Capacity Unit Licensing

Media Stream	Number of licenses per Capacity Unit	Number of licenses required per call leg
1080p30	0.5	2
720p30	1	1
480p30	2	0.5

Each CU also entitles the Licensee to content sharing in each meeting containing at least one video participant. For more information refer to the terms and conditions of the CU license.

1.6.3 Obtaining Cisco user licenses

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, ACU, 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.

1.6.4 Assigning Personal Multiparty licenses to users

Follow these steps to apply Multiparty licensing to the Meeting Server. This procedure requires that users imported from a single LDAP source are either all licensed or all not licensed.

Using the Web Admin interface of a Meeting Server in the cluster, select **Configuration**>**API**:

1. Create a userProfile or update an existing one.
   a. From the list of API objects, tap the ► after **/userProfiles**
   b. Click the **Create new** button or select the **object id** of an existing user profile
   c. Set **hasLicence** = **true** to indicate users associated with this userProfile have a Cisco Multiparty user license, or

      set **hasLicence** = **false** to indicate users associated with this userProfile do NOT have a Cisco Multiparty user license. Alternatively, leaving the **hasLicense** field unset will select the default setting of **false**.
   d. Click **Create** or **Modify** to save your change.
2. Create an ldapSource or update an existing one.
   a. From the list of API objects, tap the ► after **/ldapSources**,
   b. Click the **Create new** button or select the **object id** of an existing ldap source,
   c. Set **userProfile** = object id of the user profile created in step 1 above,
   d. Click **Create** or **Modify** to save your change. This associates the userProfile created in step 1 with the appropriate LDAP source.
3. Sync the ldap source.
   a. From the list of API objects, tap the ► after **/ldapSyncs**
   b. Click the **Create new** button,
   c. Set **ldapSource** = object id of the ldapSource created in step 2 above,
   d. Click **Create** to sync the LDAP source. All imported users will be associated with the given userProfile.

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.

1.6.5 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.

1.6.6 Determining Cisco Multiparty licensing usage

Table 6 below lists the API objects and parameters that can be used to determine the consumption of Multiparty licenses.

Table 6: 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.

1.6.7 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 Cisco Meeting App and a SIP endpoint, or two Cisco Meeting 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 Cisco Meeting App to Cisco Meeting App, Cisco Meeting App to SIP and SIP to SIP.

1.6.8 Retrieving license usage snapshots from a Meeting Server

From version 2.6, two new API objects were added to enable an administrator to retrieve license usage from the Meeting Server. These cannot be accessed though the Web Admin Interface, instead use an API tool like 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.

1.7 License reporting

1.7.1 License reporting

Meeting Server records license usage for each license type and reports the usage over a 90 day window to Cisco Meeting Management.The usage of recording licenses indicates the number of conferences recording concurrently, similarly the streaming license usage indicates the number of conferences streaming concurrently.

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 6 and Figure 7 illustrate typical deployments.

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

Figure 6: Example of a Meeting Server deployment using an Acano X-series server in a single combined server deployment

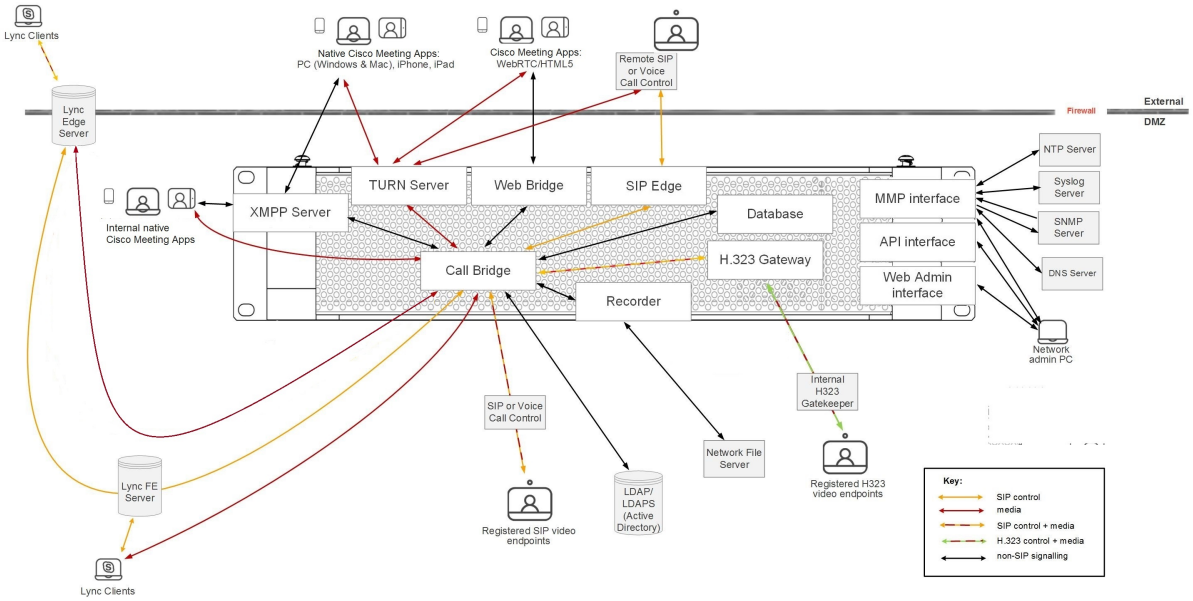
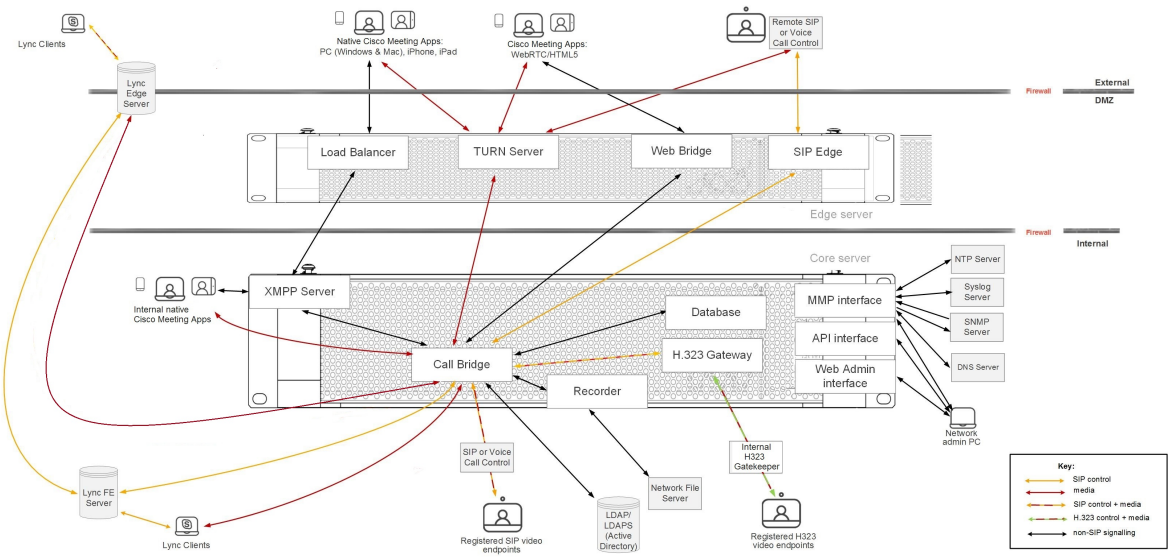


Figure 7: Example of a Meeting Server deployment using an Acano X-series servers in a split server deployment



Note:

- The Meeting Server includes a Recording facility and a Streaming facility. Only enable the Recorder/Streamer on the same server as the Call Bridge if you are simply evaluating the features. 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. See Chapter 1 for more information on recording and Chapter 17.1 for more information on streaming.
- The SIP Edge component is still a beta feature in version 2.4, and should not be deployed in a production network. It will be removed in a future version of the Cisco Meeting Server software. You are advised to start migrating over to the SIP Edge component in Cisco Expressway X8.11.
- The Meeting Server includes an H.323 Gateway. The gateway is designed to be used only with the Call Bridge. Other than a brief summary in Section 2.7 this guide does not cover the H.323 Gateway, instead see the H.323 Gateway Deployment Guide for more information. Note that Cisco plans to end of life the Cisco Meeting Server H.323 Gateway component in November 2018, after which there will be no further development or feature releases related to the H.323 Gateway. Customers are encouraged to start evaluation of the more mature H.323 Gateway component in the Cisco Expressway, and plan their migration over.

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.

An alternative to using the Web Admin Interface, is to use a REST API tool for example Postman or Chrome Poster, to access the Meeting Server's API. The Meeting Server API is routed through the Web Admin Interface, so an HTTPS connection is setup between the browser and the Meeting Server. 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 Server or Lync AVMCUs. The Call Bridge exchanges audio and video streams so that participants can see and hear each other.

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.

2.2.1 Call Bridge license

The Call Bridge license allows the Call Bridge to be used for media calls. The license needs to be installed on:

- the Cisco Meeting Server 1000,
- VM servers with Cisco Meeting Server software installed and configured as a combined server deployment (all components are on the same server),
- VM servers with Cisco Meeting Server software installed and configured as a Core server in a split server deployment.

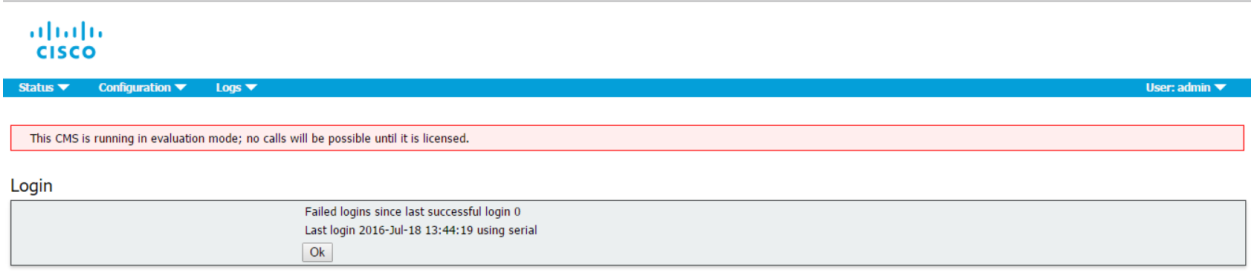
You need to have the Call Bridge activated to create any calls, if you require demo licenses to evaluate the product then contact your Cisco sales representative or Cisco partner.

Acano X-Series Servers do not require a license. VMs configured as Edge servers do not require a license for the Call Bridge.

Note: If you are deploying a cluster of Call Bridges you require a license for the cluster, all purchased features are shared across the cluster. You need to purchase a separate license for each cluster. When you purchase a license you will be asked for the MAC address of each server in the cluster that is hosting a Call Bridge which requires activation. This is the MAC address of interface A of your VM, not the MAC address of the server platform that the VM is installed on. Cisco will generate a license file for each server in the cluster, the name of each generated file will include the MAC address so that you can identify the appropriate license file to load on a server. Before uploading the license file rename it as **cms.lic**

To apply the license after uploading the license file, you need to restart the Call Bridge. However, you must configure the Call Bridge certificates and a port on which the Call Bridge listens before you can do this. These steps are part of the Meeting Server configuration and described in Section 4 and the Certificate Guidelines for Scalable and Resilient Server Deployments.

The banner “This CMS is running in evaluation mode; no calls will be possible until it is licensed.” is displayed in the Web Admin interface until a valid cms.lic file is uploaded. After you upload the license file, the banner is removed.



2.3 Database

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

In a scalable 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 Web Bridges

In version 2.9, Meeting Server introduces the new Cisco Meeting Server web app which is a browser-based client for Cisco Meeting Server that lets users join meetings (audio and video). To use this feature you need to deploy the new Web Bridge 3. In addition, Meeting Server version 2.9 still offers the original Cisco Meeting App WebRTC (also referred to here as Web Bridge 2).

In this release Cisco Meeting Server web app is fully supported for internal calls, but not recommended for external calls (see Appendix K), and it is not yet fully featured. It is intended that in due course it will support virtually the same feature set and supersede Cisco Meeting App WebRTC.

For information on enabling and configuring the original Web Bridge 2 and WebRTC app, refer to the sections Configuring the MMP and Deploying Web Bridge 2 . For further information on the WebRTC app (including browser support), see the Cisco Meeting App WebRTC Important information guide.

For information on deploying the new Web Bridge 3 and Cisco Meeting Server web app, refer to Appendix K and Cisco Meeting Server web app Important Information.

For all customization information, see Cisco Meeting Server 2.9 Customization Guidelines.

2.5 Hosting branding files locally

Note: Hosting branding files locally on Acano X Series servers is beta quality in version 2.5.x.

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 2 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 2. 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 2 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.

2.6 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.

2.7 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 Cisco Meeting Apps or SIP endpoints you need to enable the TURN server, refer to the section on Deploying the

TURN Servers . If you are using Cisco Meeting 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. This is the normal port used by the Call Bridge to connect to it, and is also available for remote connections.

The TURN server can also listen on a second port for TCP and/or TLS, typically 443.

Although the configuration option for this is named "tls", TURN actually accepts UDP, TCP and TLS on this additional port.

If you need to use TCP connections to the TURN server from a Call Bridge then either:

- on the Call Bridge set the **`tcpPortNumberOverride`** for a TURN server to the port configured (see next section)

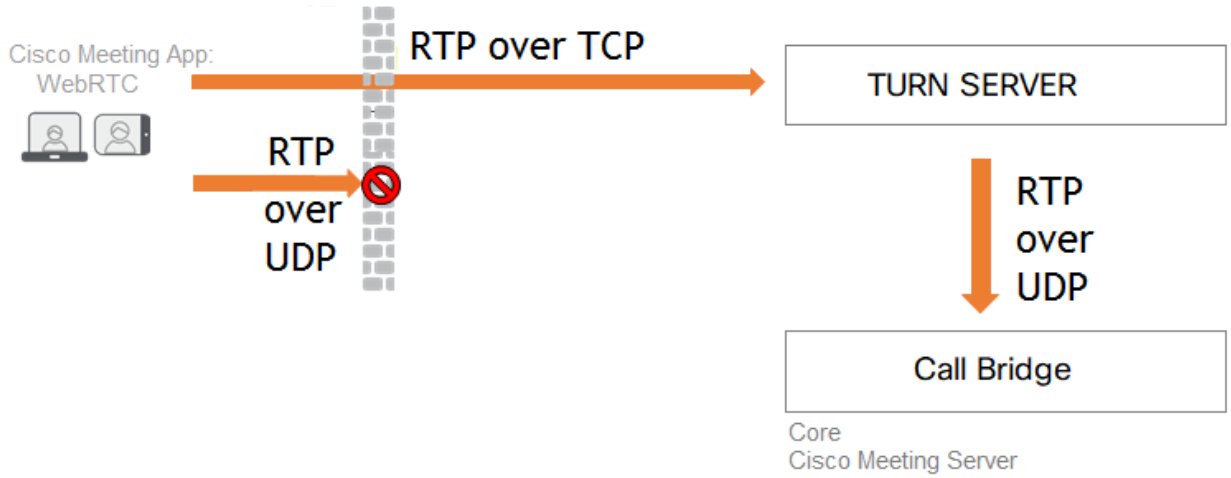
  or

- or change your firewall rules to open TCP port 3478 from the Call Bridge to the TURN server

Note: In a single combined server deployment, the TURN server will never listen on port 443 on the loopback interface.

Media sent over TCP is encrypted using TLS. The TURN server supports TCP to UDP interworking (see Figure 8). 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 8: TURN server supporting TCP and UDP



From version 2.0.4, 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.

Note: The Web Bridge sends STUN traffic to the TURN server in order to determine round trip time. For scalable deployments with multiple Web Bridges and TURN servers, the round trip time enables the Web Bridge to select the best TURN server for the session. From a network and firewall perspective, this will appear as though the Meeting Server is sending STUN traffic to it's own public IP address, network tools may flag this as an attack. This traffic can either be allowed or blocked, if this traffic is blocked the Web Bridge will choose one of the TURN servers for the WebRTC client but it might not be the best one for the WebRTC client in question. However, it should not have any other impact on the Meeting Server.

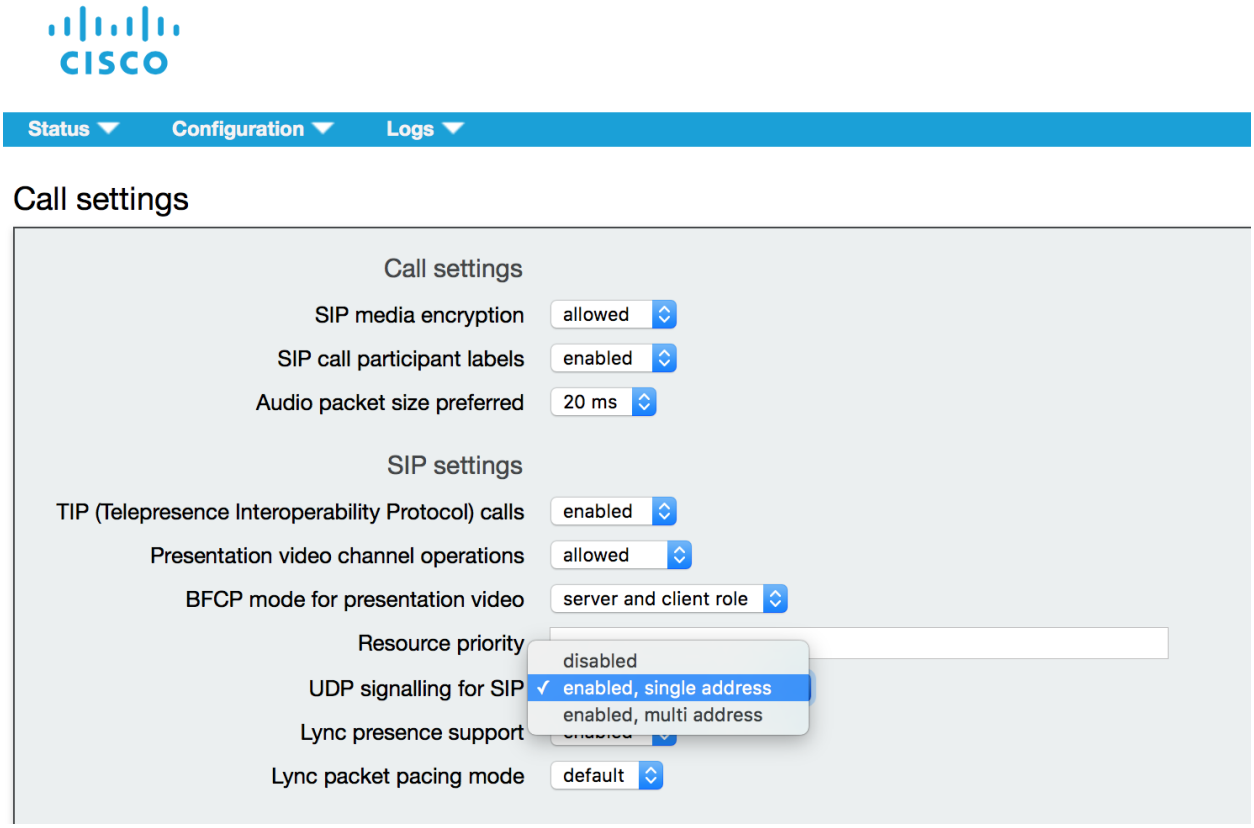
2.7.1 Enabling and disabling UDP signaling for SIP

The "UDP signaling for SIP" setting allows you to completely disable SIP over UDP, or to enable "single address" or "multi address" mode. Single address mode corresponds to the SIP over UDP behavior in versions prior to 2.2 and is the default, multi address mode allows SIP over UDP on multiple interfaces.

Use multi address mode if the Call Bridge is configured to listen on more than one interface for SIP over UDP traffic. Disable "UDP signaling for SIP" if you use SIP over TCP, or require that all of your network traffic is encrypted .

The "UDP signaling for SIP" mode is set through the Web Admin interface of the Call Bridge. Log into the Web Admin interface and select **Configuration>Call settings**, see Figure 9.

Figure 9: Settings for UDP signaling for SIP



2.8 XMPP server

Note: Cisco is simplifying the Cisco Meeting Server and Cisco Meeting App interaction, and as a result the app dependence on XMPP will be removed. Once this development is complete, Cisco will remove XMPP from the Cisco Meeting Server product line. Customers are encouraged to start planning the migration to the Cisco Meeting WebRTC app rather than using the Cisco Meeting App thick clients (Windows, Mac and iOS).

Customers who are using Cisco Meeting Apps require an XMPP license installed on the server(s) running the XMPP server application. The XMPP license is included in the Cisco Meeting Server software. You will also need a Call Bridge activated on the same Cisco Meeting Server as the XMPP server.

The XMPP server handles the signaling to and from Cisco Meeting Apps, including the WebRTC app. If you are NOT planning to use the Cisco Meeting Apps for PC, iOS (iPhone and iPad), Mac or WebRTC Client you do not need to enable the XMPP server, disregard all sections referring to the XMPP server.

2.8.1 Deploying Cisco Meeting Apps

Note: Acano clients are now referred to as Cisco Meeting Apps in the Meeting Server documentation.

If you are using any of the Cisco Meeting Apps you need to enable the XMPP server (combined deployments) or the Load Balancer, trunk and XMPP server (split deployments). Refer to the sections on Configuring the MMP, Deploying the XMPP Server and Deploying the Trunk and the Load Balancer.

CAUTION: The maximum number of concurrent XMPP clients supported by the current Meeting Server software is 500.This maximum is a total number of all different clients (Cisco Meeting App, WebRTC Sign-in and WebRTC Guest clients) registered at the same time to clustered Meeting Servers. If the number of concurrent XMPP registrations exceeds 500 sessions, some unexpected problems with sign in may occur or it may lead to a situation where all currently registered users need to re-sign in, this can cause a denial of service when all users try to sign in at the same time.

2.8.2 XMPP resiliency

The Cisco Meeting Server supports XMPP resiliency in multi-server deployments. XMPP resiliency provides fail-over protection for a client being unable to reach a specific XMPP server.

When setup in resilient mode, the XMPP servers within a deployment are loaded with the same configuration. Each knows the location of the others and they establish links between them.

Note: There is a network latency limitation (or Round Trip Time) of 200 ms or less between the XMPP servers in a cluster.

For more information on XMPP resiliency see Section 7.4.

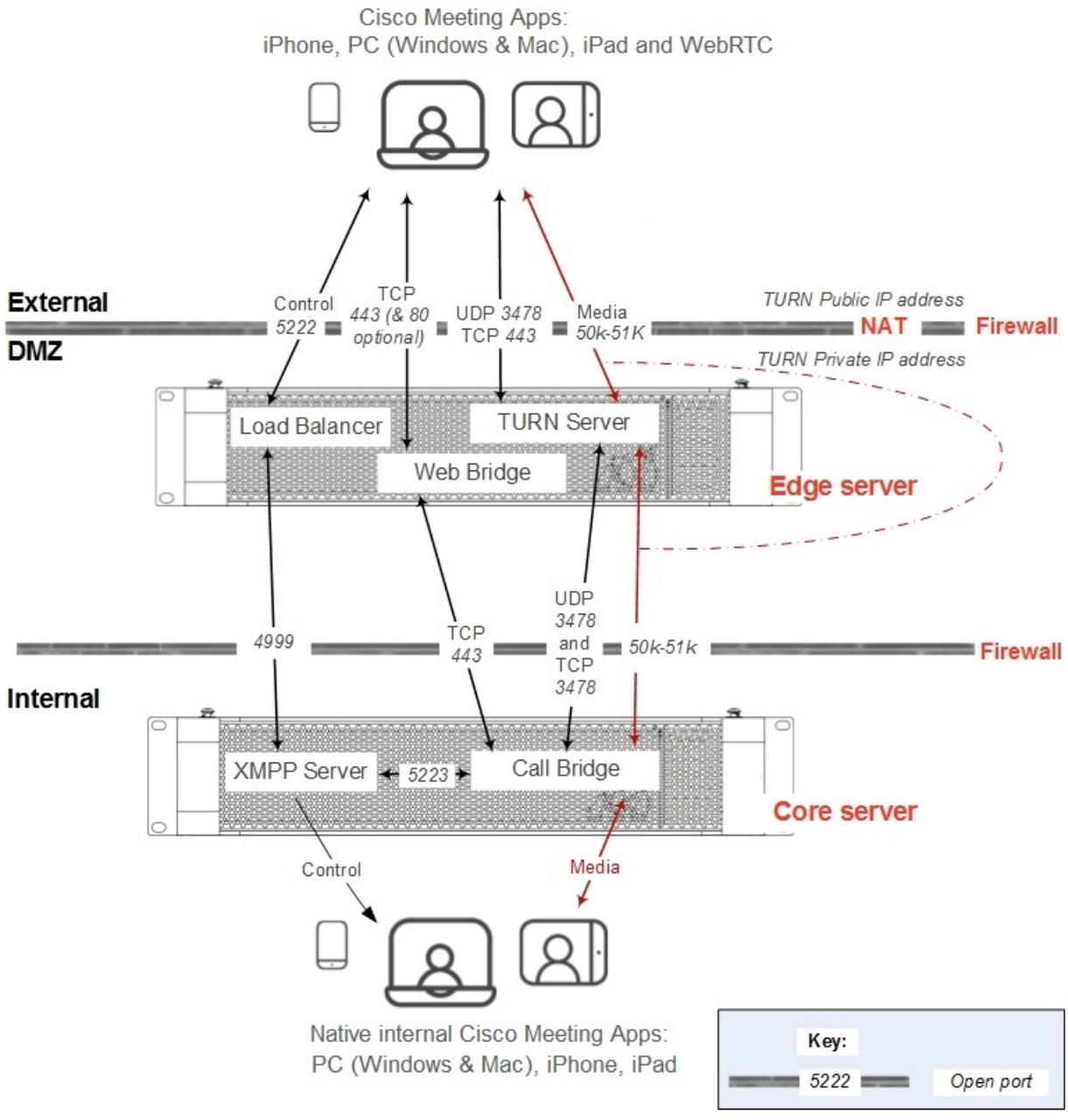
2.9 Load Balancer

The Load Balancer provides a single point of contact for Cisco Meeting Apps in split deployments. It listens on an external interface and port (as described in the XMPP client SRV record) for incoming connections. Equally, the Load Balancer accepts incoming TLS connections from the XMPP server over which it can multiplex TCP connections from external clients. This creates a TLS trunk between the Core and the Edge.

The Load Balancer does not require a license, however it does require an enabled Call Bridge.

The following diagram shows example control and media flows during a Cisco Meeting App call in a split deployment.

Figure 10: Example call flow diagram



Points to note on the figure above:

- The following ports must be open:
  - UDP Port 3478 between the TURN server and remote Cisco Meeting Apps for PC, Mac, iOS and the WebRTC client. Note: if Port 3478 is blocked, the WebRTC client will fallback to Port 443. The remote Cisco Meeting Apps for PC, Mac and iOS will not fallback to Port 443.

Note: If you use Port 443 for both the Web Bridge and the TURN server then they must be on different interfaces of the Meeting Server. Alternatively chose a different port for the TURN server.

  - UDP Port 3478 and TCP Port 3478 from Call Bridge to TURN server for split server deployments. The default configuration for the TURN server changed from version 2.0.4. TURN server now listens on TCP port 3478 for communication from the Call Bridge, rather than TCP port 443 as in previous releases. TCP port 3478 and UDP port 3478 are used in the event that ports 32768-65535 are blocked by the internal firewall.
  - UDP Port 50000-51000 from Call Bridge to TURN server (for media). Although the range between the TURN server and the external Cisco Meeting Apps is shown as 50000-51000, future releases may require a wider range of 32768-65535.
  - TCP Port 443 (HTTPS) from Call Bridge to Web Bridge (for guest login). The Web Bridge is required for Cisco Meeting Apps to look up guest login when using a web browser that does not support WebRTC (for example Internet Explorer).

- The TURN server listens on TCP port 3478 for communication from the Call Bridge. You do not need to open UDP port 3478 or TCP port 3478 as they are internal to the Meeting Server.
- Internal clients connect directly to the XMPP server on port 5222 and media connects directly between the Cisco Meeting App and the Call Bridge.
- External Cisco Meeting Apps establish a control connection to the Load Balancer (black line). Media can go directly from the Cisco Meeting App to the Call Bridge (dashed red line) or be relayed via the TURN server if required (red line).
- Both internal and external Cisco Meeting Apps use ICE/TURN to find suitable candidates for connectivity and choose the best: in the case of internal clients this will always be the local host candidates on the internal network.
- The necessary ports need to be open on the firewall between Core and Edge components to allow the media UDP traffic to pass (UDP ports 32768 - 65535) and the trunk between the Load Balancer and the XMPP server. The Web Bridge uses port 443 (and optionally port 80).
- Separate internal and external SRV records for the XMPP service need to be configured, directed to the two interfaces on the Core server/virtualized server that the XMPP server is listening on. The Call Bridge to XMPP connection should also use the XMPP server's internal address.

2.10 H.323 Gateway

Note: Cisco plans to remove the H.323 Gateway component from the Cisco Meeting Server software in a future version. Customers are encouraged to start evaluation of the more mature H.323 Gateway component in the Cisco Expressway, and plan their migration over.

The H.323 Gateway enables an H.323 call to connect to the Call Bridge. The H.323 Gateway does not provide firewall traversal or call control, you are recommended to deploy an H.323 Gatekeeper to perform these functions. With the H.323 Gateway enabled, you can make the following calls:

H.323 call > H.323 GW > space

H.323 call > H.323 GW > Call Bridge->Lync

H.323 call > H.323 GW > Call Bridge->SIP device

H.323 call > H.323 GW > Call Bridge->Cisco Meeting App

The H323 Gateway can be enabled on the same server as the Call Bridge or on a separate one. By default the H.323 Gateway uses port 6061.

Refer to the H.323 Gateway deployment guide for more information.

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 XMPP service or 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 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 13.

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 Recording meetings

Note: From 2.9, the Meeting Server allows configuration of an external third-party SIP recorder so that when recording is started an administrator-configured SIP URI is called instead of using the Meeting Server internal recorder component. For more information, see Section 16

2.13.1 License keys for recording

Recording is controlled by license keys, where one license allows one simultaneous recording. The license is applied to the server hosting the Call Bridge (core server) which connects to the Recorder, not the server hosting the Recorder.

Note: The recommended deployment for production usage of the Recorder is to run it on a dedicated VM with a minimum of 4 physical cores and 4GB of RAM. In such a deployment, the Recorder should support 2 simultaneous recordings per physical core, so a maximum of 8 simultaneous recordings.

To purchase recording license keys, you will need the following information:

- number of simultaneous recordings,
- MAC address of interface A on the servers hosting the Call Bridges.

You can purchase recording license keys through Cisco's ecommerce tool.

2.14 Streaming meetings

The Streamer component adds the capability of streaming meetings held in a space to the URI configured on the space.

An external streaming server needs to be configured to be listening on this URI. 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. However, we have only tested against Vbrick as an external streaming server.

2.14.1 License keys for streaming

You will need one or more licenses for streaming which is loaded on the Meeting Server hosting the Call Bridge, not the server hosting the Streamer. 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.

2.15 Diagnostics and troubleshooting

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
- retrieving diagnostics on Recorder/Sreamer/Web Bridge

2.15.1 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.

2.15.2 Log bundle

Meeting Server can produce a log bundle containing the configuration and state of various components in the Meeting Server. This log bundle will help Cisco Support speed up their analysis of your issue.

If you need to contact Cisco support with an issue, follow these steps to download the log bundle from the Meeting Server.

1. Connect your SFTP client to the IP address of the MMP.
2. Log in using the credentials of an MMP admin user.
3. Copy the file logbundle.tar.gz to a local folder.
4. 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.
5. Send the renamed file to your Cisco Support contact for analysis.

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.

2.15.3 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

2.15.4 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)
```

2.15.5 Retrieving diagnostics on a Recorder/Streamer/Web Bridge

There are API objects that enable the retrieval of:

- the number of **`activeRecordings`** on **`/recorders/<recorder id>`**
- the number of **`activeStreams`** on **`/streamers/<streamer id>`**:

and to retrieve the **`status`** on **`/recorders/<recorder id>`**, **`/streamers/<streamer id>`**, **`/webBridges/<web bridge id>`**. The table below shows the status settings for the components.

Status	Component	Recorder	Streamer	Web Bridge
unused	component is unused	✓	✓	✓
success	connected to the queried Call Bridge	✓	✓	✓
connectionFailure	could not connect to the queried Call Bridge	✓	✓	✓
invalidAddress	the configured URL is invalid	✓	✓	
dnsFailure	the configured URLcannot be resolved by the DNS server	✓	✓	

Status	Component	Recorder	Streamer	Web Bridge
remoteFailure	a connection was established with the component but the Call Bridge received a failure response	✓	✓	
unknownFailure	an unknown failure occured	✓	✓	
lowDiskSpace	has limited disk space available	✓		

2.16 General points about scalability and resilience

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.

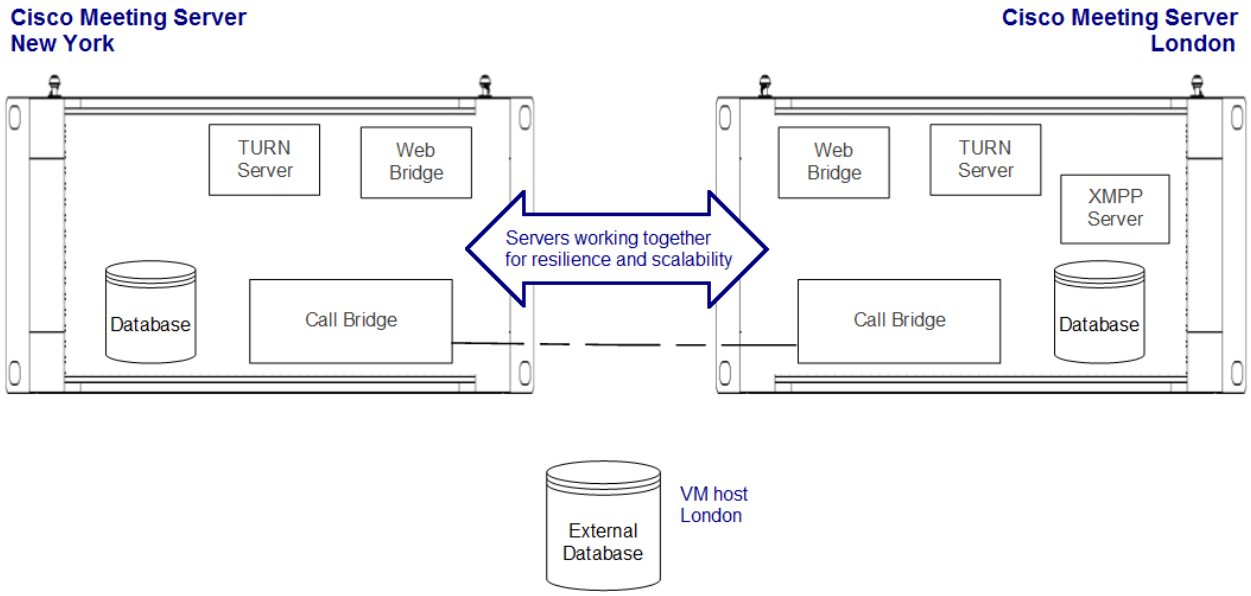
- Multiple Edge and/or Core servers (Edge servers will have edge components enabled, Core servers will have core components enabled, Call Bridge and the database are always core components). The location of these Edge and Core servers will depend on your requirements, possibly co-locating Core and Edge 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.

2.16.1 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 apart from the XMPP server on the New York server, 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 the TURN server and Web Bridge on the other host server, and the XMPP server on the London host. 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.

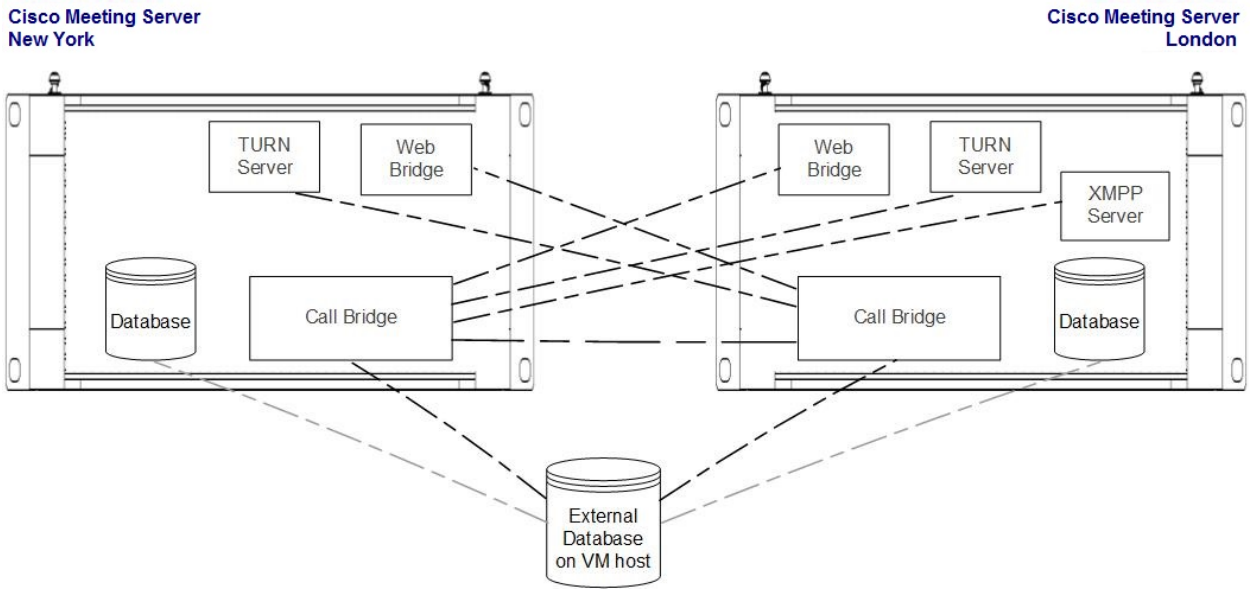
- The two Web Bridges are configured using the API as described in Appendix 1.2
  - The Web Bridge services can be configured to have a single DNS A record externally. However, when configuring the Web Bridges on the Call Bridge(s) there must be a unique hostname or IP address for each Web Bridge configured on the Call Bridge(s). This enables, each Web Bridge to be uniquely identifiable by every Call Bridge
- The TURN servers are configured via the API as described in Appendix 1.4
  - The TURN servers can be configured either by hostname (with one hostname resolving to potentially multiple servers via DNS) or by IP address. This configuration is stored in the shared database.
  - The Cisco Meeting Apps are always monitoring TURN servers in the background via their connections to an XMPP server. When a call starts, the client is sent a list of available TURN servers but will have already chosen the best TURN server for each interface: therefore, when joining a call there is no additional delay in choosing a TURN server. Cisco reserves the right to change and enhance the algorithms used

- The XMPP server is configured using the MMP as described in Configuring the MMP.

Note: You may have multiple XMPP servers in your deployment, providing each XMPP server is in a different domain to the other XMPP servers. A single XMPP server can host multiple XMPP domains. For example, both example.com and example.org can exist on the same Meeting Server. For information on setting up XMPP multi-domains, see Section 7.

Figure 11 transforms into Figure 12 when connections are shown.

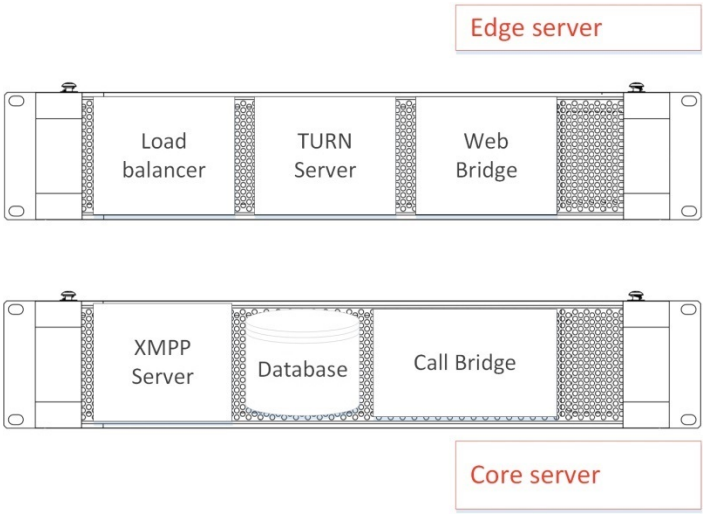
Figure 12: Simple deployment showing connections



2.16.2 Example using split deployments and geo-distribution

Cisco Meeting Apps require a persistent connection to the XMPP server. If you have security concerns, the XMPP server should be run on a Core server and a Load Balancer run on an Edge server to provide the persistent connection to the clients. Figure 13 is a schematic of the single split deployment with components rearranged for scalability and resilience: the XMPP server moved to the Core server with the Load Balancer on the Edge server.

Figure 13: Split Core and Edge servers showing location of XMPP server on Core server

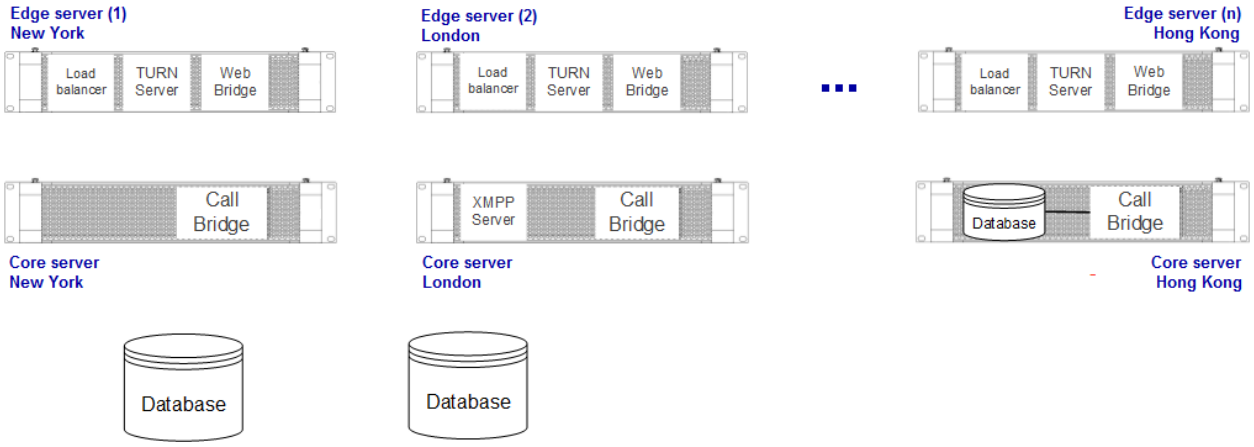


- The Load Balancer provides a single point of contact for Cisco Meeting Apps and the XMPP server. It listens on an external interface and port (as described in the XMPP client SRV record) for incoming connections. Equally, the Load Balancer accepts incoming TLS connections from the XMPP server over which it can multiplex TCP connections from external clients. This creates a TLS trunk between the Core and the Edge. More than one Edge server can trunk to the XMPP server.

  The Load Balancer does not require a license
- The Web Bridge and Call Bridge are configured using a DNS A record, and the XMPP server has associated XMPP SRV records
- Connections between the Call Bridge on the Core server and the Web Bridge and TURN servers on the Edge servers use same ports as in previous releases: that is, 443 and 3478 respectively

This deployment can be scaled in a number of ways, for example see Figure 14 below. The Hong Kong Core server currently hosts the database; New York and London use external databases.

Figure 14: Multi Core & Edge server deployment



2.16.3 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.
- When using certificates, all inter-database communication between database cluster peers is handled through SSL for security. When no certificates are used then no security is present.
- 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 (reading from a replica database), but will not be able to perform any database writes. This includes modification operations via the API, modification of spaces via a Cisco Meeting App, login via XMPP and LDAP sync. SIP calls will operate as normal.
- 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 database nodes forming the 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.

3 Prerequisites

3.1 Prerequisites

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 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 Load Balancer (if used) and the XMPP server.

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 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 change. 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 Cisco Meeting 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 Cisco Meeting 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 Cisco Meeting App calls. Instructions for deploying this dial plan are given in Chapter 11. 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 Cisco Meeting Apps as appropriate.
- Access to a SIP Call Control platform if you intend to make SIP calls. Chapter 12 and Chapter 13 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.

3.1.10 Specific prerequisites for Acano X-series server

- A suitable environment: refer to the Acano Hardware/Environmental Data Sheet for details on the required power and cooling
- The Acano X- series server has two power modules, and country-specific power cables are supplied for the AC power supplies. The server will work with just a single power unit connected. To implement power supply redundancy you must connect both modules to power supplies. Connecting the modules to independent power supplies allows for the greatest resiliency.
- 2U of rack space if using the rack mounting kit; 3U of rack space if installing on a shelf
- A minimum of two Ethernet links:
  - One for the MMP (labeled Admin on the back of the Acano X-series server). The speed can be 100M or 1G.
  - One for a media interface (there are four labeled A to D). The speed can be 1G or10G.

IP addresses can be configured statically or automatically via DHCP or SLAAC/DHCPv6. Ethernet links will operate at the speed of the network switch; the switch port should be set to auto negotiate speed. If you are using a speed of 10G be sure to use the appropriate cable.

See the Installation Guide for the Acano X-series server for full details.

4 Configuring the MMP

The Meeting Server components are configured using the MMP.

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

The Cisco Meeting Server 2000, Cisco Meeting Server 1000 and Acano X-series servers ship with the latest software release available at the time of shipment, but may not be up-to-date. Equally, 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.

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, 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.

7. Run the **database cluster upgrade_schema** command on the primary database server. Use the **database cluster status** command to check that the database schema has upgraded successfully.

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,
- XMPP service is connected (if used),
- and no fault conditions are reported on the Web Admin interface and log files.

Check that you are able to connect using SIP and Cisco Meeting App (as well as Web Bridge 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. In the case of XMPP clustering, you will have to re-cluster XMPP (by reinitializing XMPP on the node and joining the others to it) as that will not successfully restore from the **`backup rollback <name>`** command. 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 needs a key and certificate pair that is used to establish TLS connections with SIP Call Control devices and with the Lync Front End (FE) server. If you are using Lync, this certificate will need to be trusted by the Lync FE server.

Note: SIP and Lync calls can traverse local firewalls using the SIP Edge component, this is a beta feature and should not be used in production environments. If you plan to evaluate this feature, note that you need to configure trust between the Call Bridge and the SIP Edge, for more information see Chapter 15.

Note: SIP and Lync 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 (chosen from A, B, C or D). By default the Call Bridge listens on no interfaces.

Configure listening interfaces on each Call Bridge as follows:

1. Create and upload the certificate 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 be listening on a network interface that is not NAT'd to another IP address. This is because the Call Bridge is required to convey the same IP that is configured on the interface in SIP messages when talking to a remote site.

3. Configure the Call Bridge to use the certificates by using the following command so that a TLS connection can be established between the Lync FE server and the Call Bridge, for example:

   ```
   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.
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 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 Configuring the XMPP server

If you are using the Recorder or Streamer components or any of the Cisco Meeting Apps including the WebRTC Client you now need to configure the XMPP server and then enable it. Otherwise, skip this section.

Note: The Recorder and Streamer components behave as XMPP clients, so the XMPP server needs to be enabled on the Meeting Server hosting the Call Bridge.

From Cisco Meeting Server 2.0, the XMPP license is included in the Cisco Meeting Server software.

Note: You will need a Call Bridge activated on the same Meeting Server as the XMPP server.

---

1. Follow the instructions in Section 7.1 to set the DNS records for the XMPP server
2. Sign into the MMP and generate the private key and certificate using the information in the Certificate Guidelines. Upload the certificates to the server hosting the XMPP server.

The XMPP server can be configured to listen on any subset of the four media interfaces and ignore connections from any interface in the complement.

3. Establish an SSH connection to the MMP and log in.
4. To configure the XMPP server to use one or more interfaces enter the following command:

```
xmpp listen <interface allowed list>
```

The following is an example where interface is set to interface A and B.

```
xmpp listen a b
```

5. Assign the certificate and private key files that were uploaded earlier, using the command:

   xmpp certs <keyfile> <certificatefile> [<cert-bundle>]

   where keyfile and certificatefile are the filenames of the matching private key and certificate . If your CA provides a certificate bundle then also include the bundle as a separate file to the certificate. See the Certificate Guidelines for further information

6. Configure the XMPP server with the following command:

   xmpp domain <domain name>

   The following is an example where the domain name is example.com.

   xmpp domain example.com

7. Enable the XMPP service:

   xmpp enable

8. To allow a Call Bridge to access the XMPP server securely (after configuration), provide a component name for the Call Bridge to use to authenticate e.g. **cb_london**:

   xmpp callbridge add <component name>

   for example

   xmpp callbridge add cb_london

   A secret is generated; for example, you see:

   ```
   cms>xmpp callbridge add cb_london
   Added callbridge: Secret: aB45d98asdf9gabgAb1
   ```

   Note: Each Call Bridge requires a unique component name so that all the Call Bridges can connect to the XMPP server at the same time.

   Note: When using Call Bridge Groups, either all of the Call Bridges in a Call Bridge Group, or none of them, should be added in this step.

9. Make a note of the domain, component and secret generated in the previous steps because they are required when you use the Web Admin interface to configure the Call Bridge access to the XMPP server (so that the Call Bridge will present the authentication details to the XMPP server).

   Note: If you lose the details, use the MMP command **xmpp callbridge list** to display them.

4.5.1 Configuring XMPP multi-domains

A single XMPP server can host multiple XMPP domains. For example, both example.com and example.org can exist on the same Meeting Server. It is possible to configure multiple tenants

with the same XMPP domain (as in previous releases), or each tenant with their own domain, or mix these schemes.

Note: It is strongly recommended that multiple XMPP domains are not used for a single tenant, or in cases where tenants are not used.

To configure multiple domains for the XMPP server to listen to, use the MMP command:

```
xmpp multi_domain add <domain name> <keyfile> <certificatefile> [<crt-bundle>]
```

where:

<keyfile> is the private key that you created for the XMPP server

<certificatefile> is the signed certificate file for the XMPP server

[<crt-bundle>] is the optional certificate bundle as provided by the CA

Note: You also need to add a DNS SRV record for each additional XMPP domain, and to add the domain to the Incoming Calls page on the Web Admin interface (**Configuration>Incoming calls**).

Note: Restart the XMPP server for the configured multiple domains to take effect.

Note: The XMPP server will not start if the private key or certificate files are missing or invalid.

To list the domains that the XMPP server is listening to, use the command:

```
xmpp multi_domain list
```

To delete a domain that the XMPP server is listening to, use the command:

```
xmpp multi_domain del <domain name>
```

4.6 Configuring Web Bridge 2

Web Bridge 2 is used by the WebRTC app. If you are deploying the WebRTC app you need to set the network interface for the Web Bridge 2 and then enable it. Otherwise, skip this section.

Note: If you wish to deploy Web Bridge 3 and the new Cisco Meeting Server web app introduced in 2.9, skip this section and refer to Appendix K.

1. SSH into the MMP.
2. Configure the Web Bridge 2 to listen on the interface(s) of your choice with the following command:

   ```
   webbridge listen <interface[:port] allowed list>
   ```

The Web Bridge can listen on multiple interfaces, e.g. one on public IP and one on the internal network. (However, it cannot listen on more than one port on the same interface.)

The following is an example where interfaces are set to interface A and B, both using port 443.

```
webbridge listen a:443 b:443
```

3. Create DNS A record for the Web Bridge and set it to resolve to the IP address of the Ethernet interface you want the Web Bridge to listen on.
4. Create a certificate and private key for the Web Bridge 2 as described in the Certificates Guidelines. Upload the certificate file to the MMP via SFTP.
5. Add the Call Bridge certificate to the Web Bridge 2 trust store as described in the Certificates Guidelines document.
6. The Web Bridge 2 supports HTTPS. It will forward HTTP to HTTPS if configured to use "http-redirect". To do so:

   a. Enable HTTP redirect with the following command:

   ```
   webbridge http-redirect enable
   ```

   b. If required (see the note below), set the Windows MSI, Mac OSX DMG and iOS installers that are presented to WebRTC users:

   ```
   webbridge clickonce <url>
   webbridge msi <url>
   webbridge dmg <url>
   webbridge ios <url>
   ```

   Note: If you only use browsers that support WebRTC (e.g. Chrome) you do not need to set these download locations because browser functionality will be used for guest access to space. However, if you use browsers that do not (e.g. IE, Safari) then configure these locations so that when the Meeting Server detects the device being used (iOS device, Mac, or PC), it can redirect the user to the configured client download link for that device, and prompt the user to install the correct Cisco Meeting App so that they can join the meeting. After installation, the user is connected to the space as a Guest.

7. Enable the Web Bridge 2 with the following command:

   ```
   webbridge enable
   ```

8. Use the Web Admin interface to configure the settings through which the Call Bridge communicates with the Web Bridge 2.

4.7 Configuring the TURN server

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

1. SSH into the MMP.
2. Configure the TURN server with the following command:

   `turn credentials <username> <password> <realm>`

   The following is an example where username is **`myusername`**, the password is **`mypassword`** and it uses the realm **`example.com`**.

   `turn credentials myTurnUsername myTurnPassword example.com`

3. If the TURN server has a public IP address rather than being NAT'ed (see Figure 16), this step is not required, go on to step 4. If the TURN server is located behind a NAT, set the public IP Address that the TURN Server will advertise using:

   `turn public-ip <ip address>`

   The following is an example where a public IP address is set to 5.10.20.99

   `turn public-ip 5.10.20.99`

   CAUTION: Locating the TURN server behing a NAT requires careful configuration of the NAT, to ensure connectivity always works. This is due to how Interactive Connectivity Establishment (ICE) works, and is not a problem specific to the TURN deployment within the Meeting Server. For information on deploying TURN servers behind NAT, see Appendix H.

   Note: The IP address set here should not be confused with the IP addresses set in the Web Admin Interface **Configuration > General** page. The MMP commands configure the TURN server itself, while the **Configuration > General** page settings allow the Call Bridge and external clients to access the TURN server, and are explained in *Web Admin interface settings for the TURN server*.

Figure 15: TURN server public IP address (not NAT'ed) – Combined server deployment

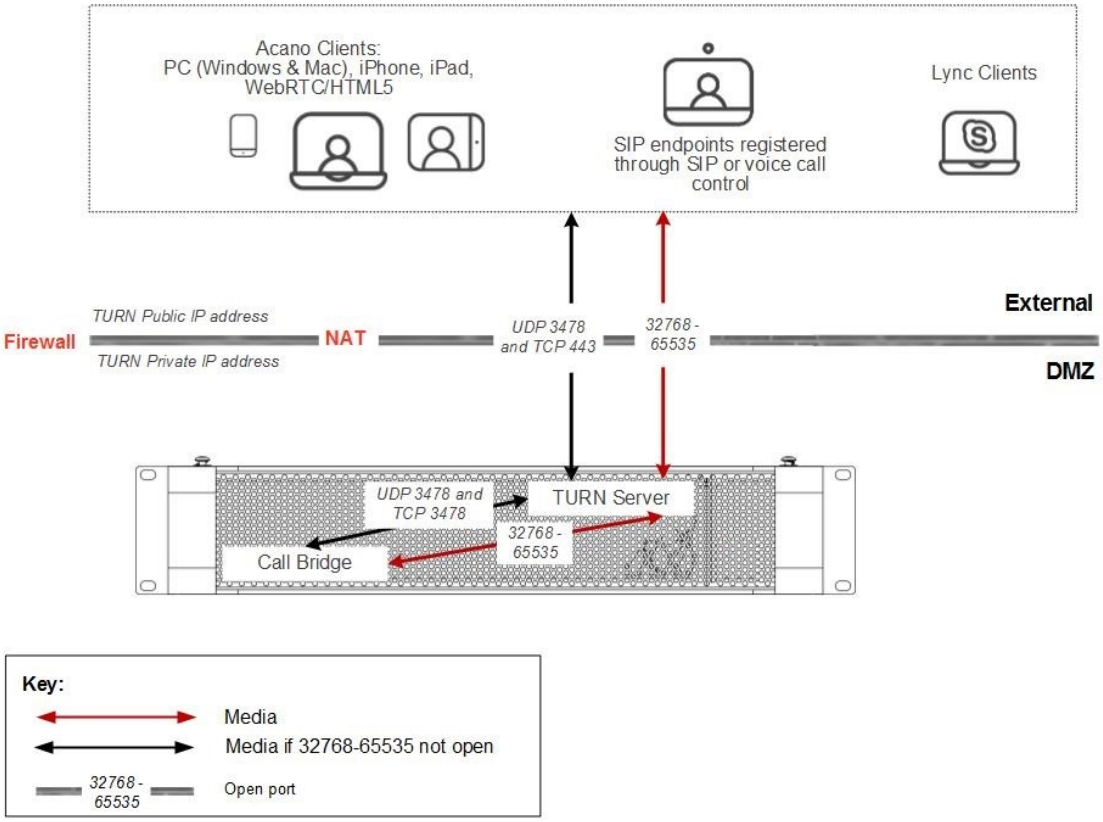
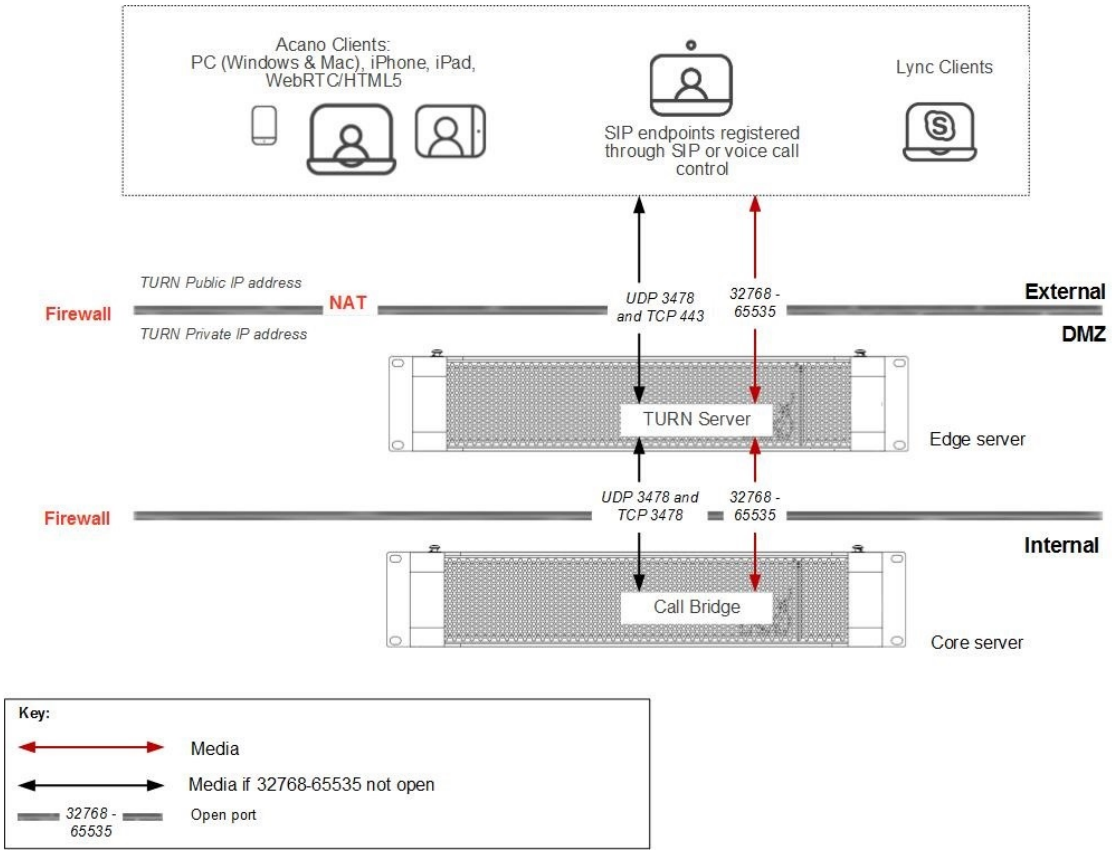


Figure 16: TURN server public IP address (not NAT'ed) - Split server deployment



Note: Although the port range between the TURN server and the external clients is shown as 32768-65535, currently only 50000-51000 is used. The required range is likely to be larger in future releases.

4. Configure the TURN Server to listen on a specific interface using:

   `turn listen <interface allowed list>`

   In a single combined server deployment, the TURN server must be configured to listen on the loopback interface. Ensure that the allowed list of interfaces to listen on contains at least one interface, and specify the loopback interface. The loopback interface must not be the first interface in the allowed list.

   For example:

   `turn listen c lo` when a Call Bridge or Web Bridge is colocated on the same server as this TURN Server

   `turn listen c` for split server deployments

Note: You can specify more than one interface for the TURN server to listen on. If specifying multiple interfaces for the TURN server, the first one must be the public interface, i.e. the one on the public network, or the one that a NAT forwards to. For example, `turn listen b a` where `b` is the NAT'd interface and `a` is the private internal interface.

5. Select the port for the TURN server to listen on using:

   ```
   turn tls <port|none>
   ```

   for example:

   ```
   turn tls 443
   ```

   Note: For maximum connectivity from external locations, Cisco recommends that port 443 is used for both the TURN Server and the Web Bridge. However, to set up TCP to UDP interworking on a TURN server, the Web Bridge and TURN Server must listen on different interface:port combinations.

   To run both the TURN server and the Web Bridge on port 443 requires them to be run on separate servers/VMs, or if on the same server/VM they need to be on different interfaces and different subnets.

   If this is not possible then select a non-standard port for the TURN server, for example: `turn tls 447` and use the `tcpPortNumberOverride` parameter to configure the port on the Call Bridge (see step 7).

6. Since media sent over TCP is encrypted using TLS, a certificate is required on each TURN server that carries out TCP to UDP interworking. The certificate should be signed by the same CA as that used for the Web Bridge.

   a. Generate a private key and the Certificate Signing Request (.csr) file for the TURN server. For information on how to generate a private key and .csr file, refer to the Certificate Guidelines.

      Note: The public key is created and held within the .csr file.

   b. Submit the .csr file to the CA for signing.

   c. SSH into the MMP

   d. Disable the TURN server interface before assigning the certificate

      ```
      turn disable
      ```

   e. Upload the signed certificate and intermediate CA bundle (if any) to the Meeting Server using SFTP.

   f. Check that the certificate (and certificate bundle) and the private key match

```
pki match <certicatefile> <cert bundle/CA cert> [<CA cert>]
```

g. Check that the specified certificate is signed by the root CA using the certificate bundle to determine the chain of trust

```
pki verify <certicatefile> <cert bundle/CA cert> [<CA cert>]
```

h. Assign the certificate (and certificate bundle) and private key pair to the TURN server

```
turn certs <keyfile> <certificatefile> [<cert-bundle>]
```

i. Re-enable the TURN server

```
turn enable
```

7. If in step 5 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.

---

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 either one database node in the cluster or 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.

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 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.

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.

1. On the server with the database that will start as primary, sign in to the MMP.
2. 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`
3. 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:

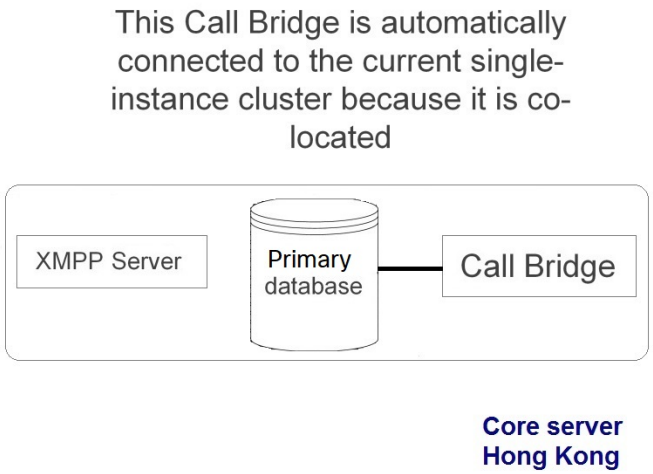
   - [a|b|c|d|e] – the name of the interface (the first IPv6 address is preferred, otherwise the first IPv4 address is chosen) e.g.
     `database cluster localnode a`
   - ipv4:[a|b|c|d|e] – the name of the interface restricted to IPv4 (the first IPv4 address is chosen) e.g. **`database cluster localnode 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.

3. Enter the MMP command: **database cluster initialize** and press **Y** in response to the prompt to initialize this as the primary database for 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 17: Co-located Call Bridges are automatically connected



4. 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>
      ```

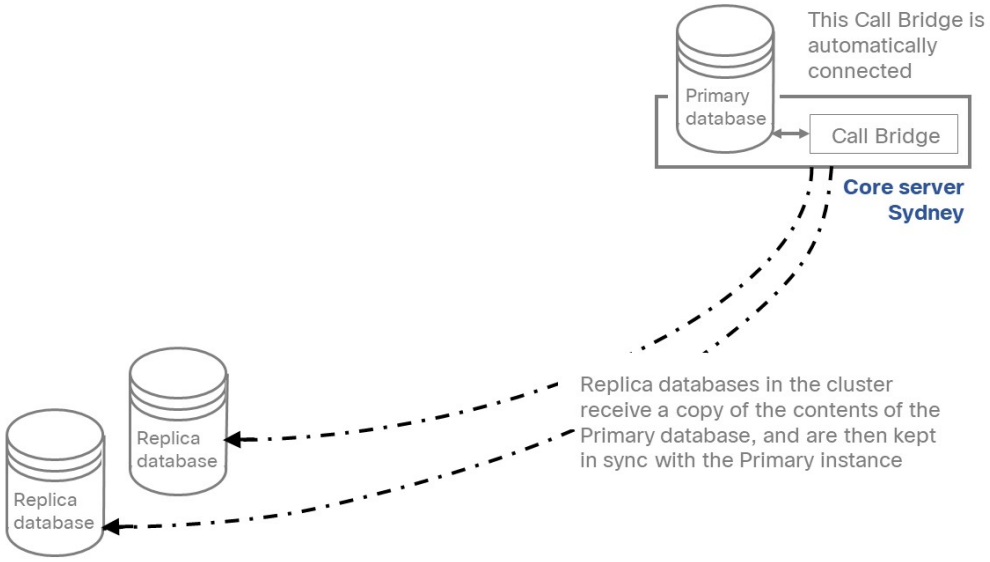
      Note: <interface> can be in any of the formats listed previously in this section

      For example:

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

   b. "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 18: 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).

c. 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).

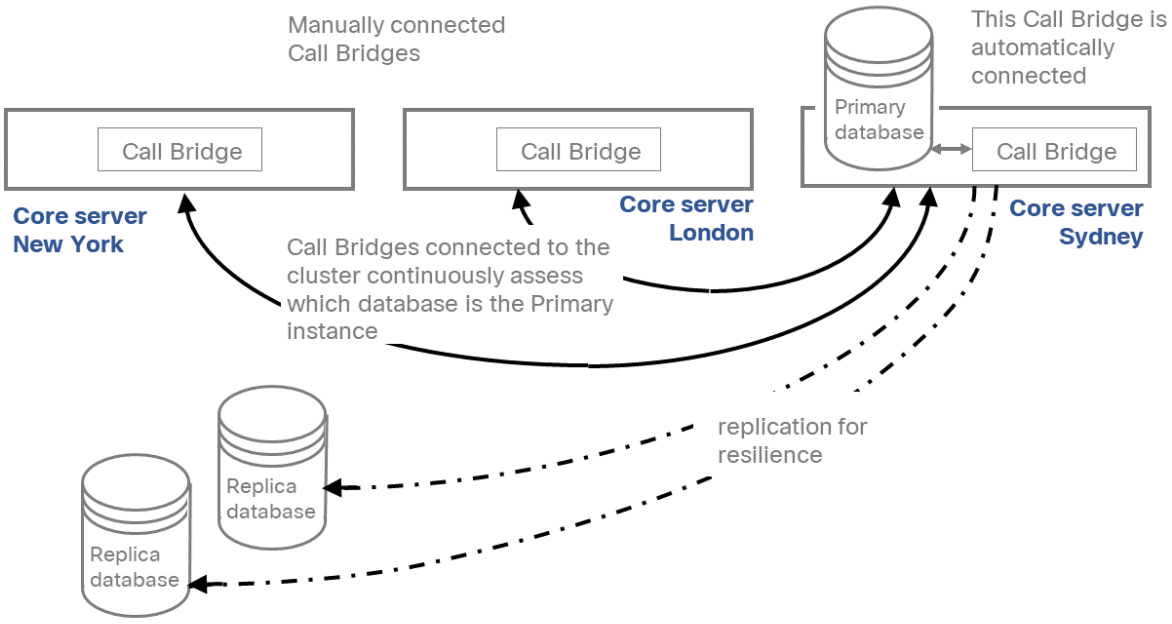
1. Sign in to the MMP of the Core server with an unconnected Call Bridge and issue the command **database cluster connect <hostname/IP>**. The hostname or IP address can be for any database in the cluster.

   ```
   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 19: Example of database clustering and Call Bridge connections



5.6 Upgrading the database schema

Note: This section does not apply when you first set up database clustering, but these steps must be run after every subsequent Core server software update; otherwise the Call Bridges and databases may be out-of-step with regards to the database schema.

The procedure for upgrading a clustered system is:

1. Before upgrading your Meeting Servers, take a backup of the current configuration on each of the servers. Use the **`backup snapshot <filename> command`**. 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.
2. Save the cms.lic and certificate files to the local drive as well.
3. Now check the **`database cluster status`**. Check that all calls (SIP and clients) are working and no fault conditions are listed on the Web Admin pages. Identify which server has the primary database.
4. Upgrade each database, one by one, starting with the replica databases and finishing with the primary database. Wait until each server has fully booted, and for the database to have

connected to the database cluster BEFORE upgrading the next server. Use the **`database cluster status`** command to check for full connection to the cluster.

At this stage **`database cluster status`** should report that all nodes are healthy and in sync, but the Call Bridges will not be operating correctly and will show database errors.

Do not go onto the next step until the database servers are in sync.

5. Log into the MMP of the server hosting the primary database and issue the command **`database cluster upgrade_schema`**
6. Verify that the operation was successful by using **`database cluster status`**

   A status of Enabled means success, whereas Error indicates an issue.
7. Check that the Web Admin interface on each Call Bridge can display the list of spaces.

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 guidelines for information on creating, uploading and assigning certificates to Call Bridges.

Note: SIP and Lync call traversal of local firewalls using the SIP Edge component, is a beta feature and should not be used in production environments. If you plan to evaluate this feature, note that you need to configure trust between the Call Bridge and the SIP Edge, for more information see Chapter 15.

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).

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.

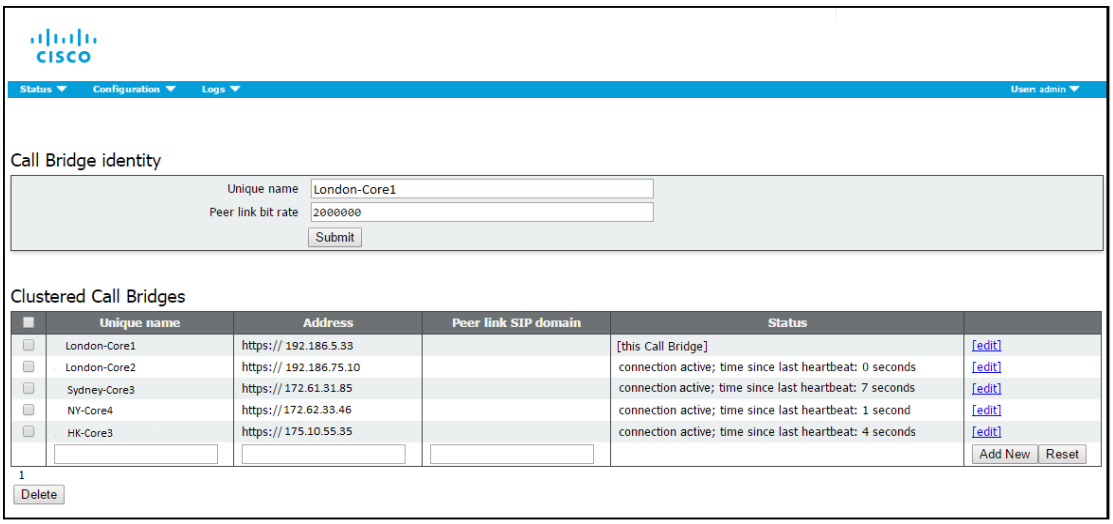
   c. Leave the Peer Link SIP Domain blank unless you have call control devices between your Call Bridges. If you leave the field blank then the address of the outbound SIP call will be in the form of a random URI generated by the Call Bridge followed by the IP address of the Call Bridge that it is linking to (for example randomURI@10.10.10.10) . 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 matches the Peer Link SIP Domain. If routing through an existing call control, it is recommended to use a unique domain or each servers' FQDN for the Peer link SIP domain. This ensures no accidental call loops that 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.

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.

Prior to version 2.4, DTMF sequences could only be configured for call legs on a local Call Bridge. DTMF sequences could not be configured via the participant API, which meant that DTMF could not be sent to participants on a remote Call Bridge, or when calling out from a cluster of Call Bridges.

From version 2.4, 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 11 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.

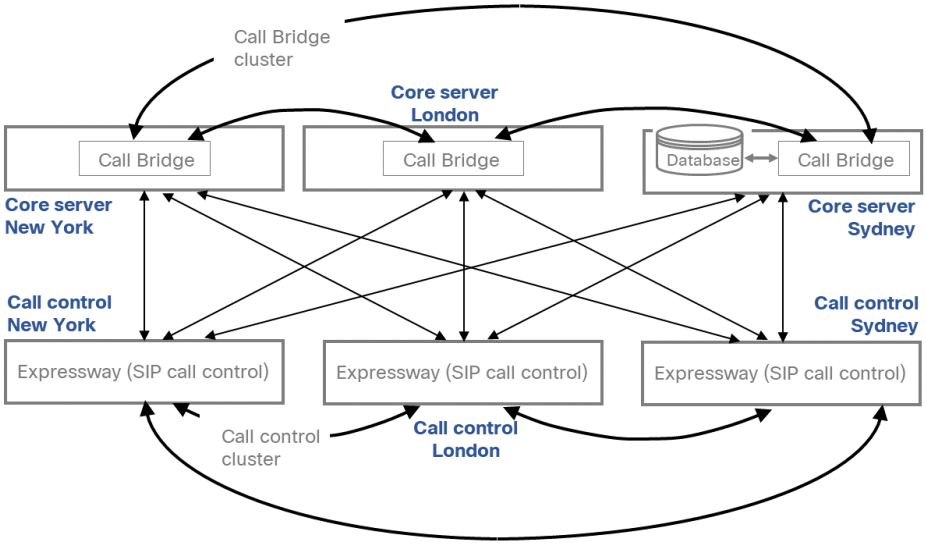
6.4.2 Examples

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.

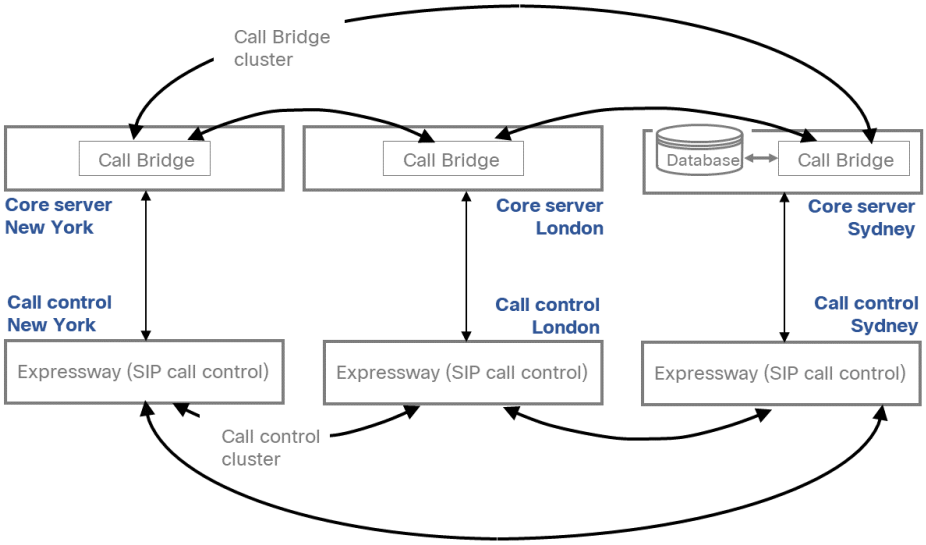
Example #1: One option is to configure a cluster of VCSs to forward calls to the cluster of Call Bridges. This provides resiliency because any VCS can route a call to any Call Bridge, and calls from any Call Bridge can be routed to any VCS (see the figure below).

Figure 20: 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 21: 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 7 for load limits.

Table 7: Load limits for server platforms

System	Load Limit
Meeting Server 2000	700000 (see note below)
Meeting Server 1000	96000
X3	250000
X2	125000
X1	25000
VM	1250 per vCPU

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.

Note: From version 2.6, the call capacity for Cisco Meeting Server 2000 with Call Bridge Groups enabled, has increased to 700 HD calls, and the loadlimit has increased from 500000 to 700000. The load calculation for the different call resolutions has been updated to match the new 700000 limit. Load limits for other Meeting Server platforms stay as they were previously; these changes only apply to the Cisco Meeting Server 2000.

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 exisiting 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 Cisco Meeting 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 Cisco Meeting App participants (including the WebRTC app users):

- a Cisco Meeting App user joining as a member of the space,
- a Cisco Meeting App user joining as a non-member of the space, with and without a passcode
- a guest user joining the space,
- a participant added to a space from Cisco Meeting App.

By default, Cisco Meeting App participants are also load balanced if the **`loadBalancingEnabled`** parameter is set to true on the **`/callBridgeGroups`** API object (by default it is set to false). The decision on where to place the call is no longer restricted to the first Call Bridge which the Cisco Meeting App connects to.

The load balancing algorithm includes:

- Cisco Meeting App participants added via the API with a Call Bridge Group specified. The media will come from a Call Bridge in the specified Call Bridge Group, the Call Bridge chosen will be based on the existing algorithms
- Cisco Meeting App participants added via the API with a Call Bridge specified. The media will come from that Call Bridge.
- Cisco Meeting App participants simply joining a space without having been added to the space via the API. If this occurs, the Call Bridge that the Cisco Meeting App first connects to is determined, if that Call Bridge is part of a Call Bridge Group then the call is load balanced.

To load balance Cisco Meeting App calls, ensure that each Call Bridge in the Call Bridge Group has a connection to the XMPP cluster or single XMPP server, see the appropriate deployment guide for details on how to configure the connection.

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

Disabling load balancing Cisco Meeting App participants

To disable load balancing Cisco Meeting 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:

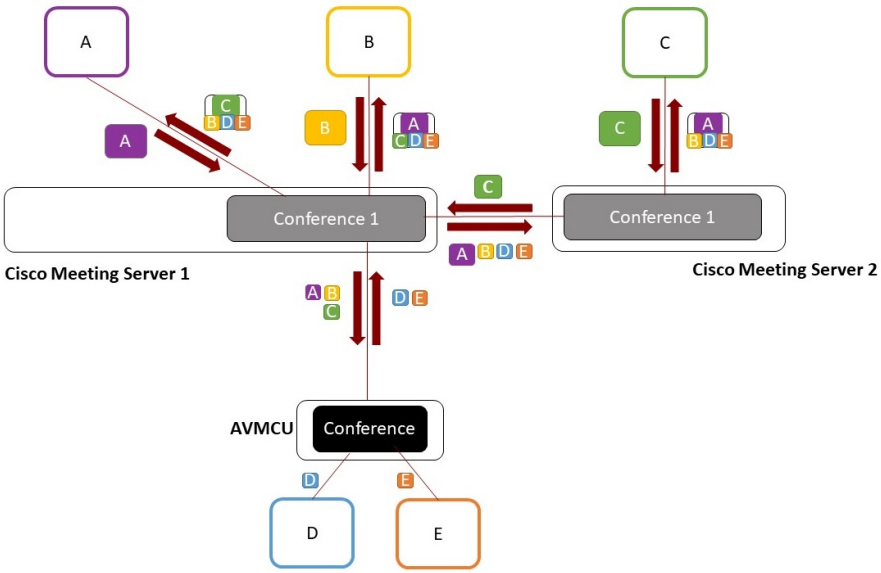
- **`dualHomeCluster`**: media streams are sent between the clustered Meeting Servers, see Figure 22. All calls from the SIP participants will be combined into one conference spanning both Call Bridges; one Call Bridge will call out to the AVMCU. **`dualHomeCluster`** uses one Multiparty Plus license 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 22: Lync AVMCU/Meeting Server deployment using dualHomeCluster mode



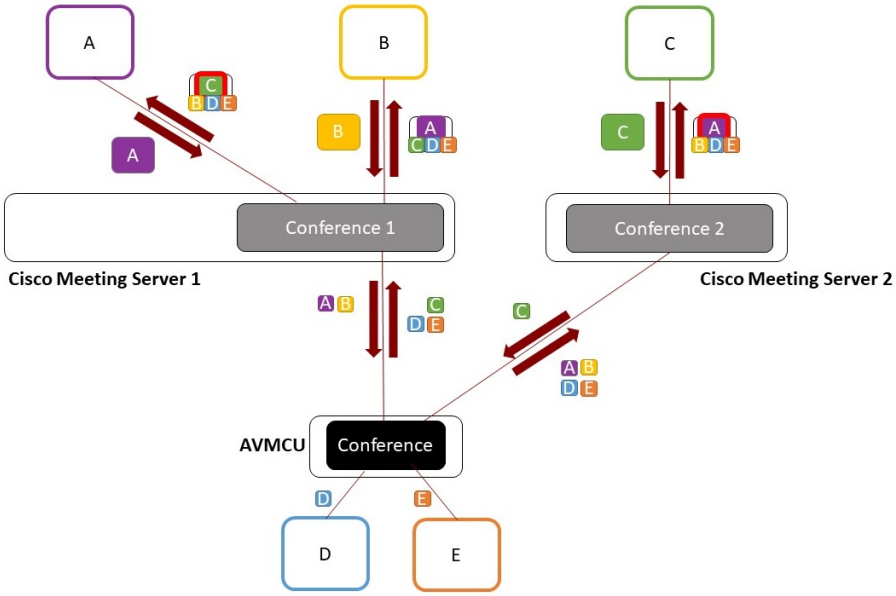
- **dualHomeCallBridge**: will result in the two SIP participants on the same Call Bridge being combined into one conference, see Figure 23. Streams seen by endpoint C come via the AVMCU, the stream of endpoint A seen by endpoint B does not come via the AVMCU. **dualHomeCallBridge** mode involves multiple conferences on the Meeting Servers and will consume multiple Multiparty Plus licenses; two Multiparty Plus licenses are consumed in the example given in Figure 23.

  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 23 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 23: 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 24,

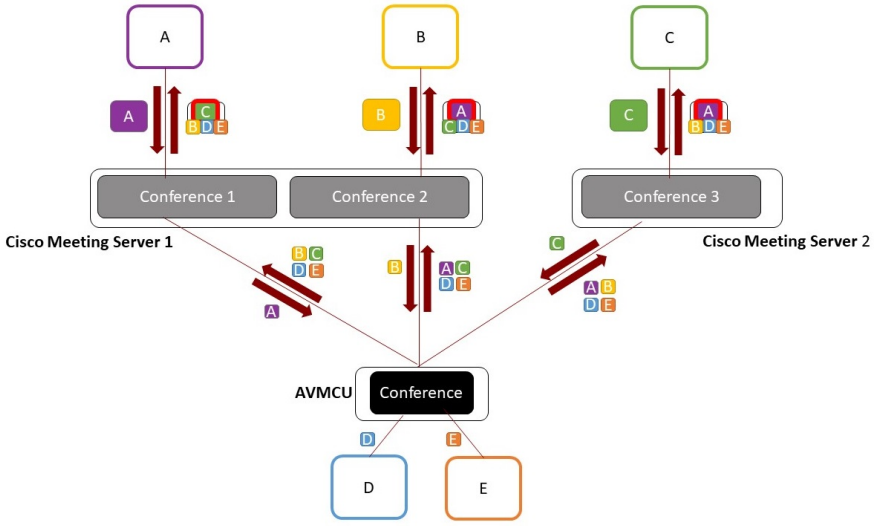
and can potentially be at lower resolution, indicated by a red outline around the main panes potentially affected.

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 24, 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 24: Lync AVMCU/Meeting Server deployment using gateway mode



6.8 More video streams over distribution links between clustered Call Bridges (preview feature)

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 the XMPP Server

Note: Cisco is simplifying the Cisco Meeting Server and Cisco Meeting App interaction, and as a result the app dependence on XMPP will be removed. Once this development is complete, Cisco will remove XMPP from the Cisco Meeting Server product line. Customers are encouraged to start planning the migration to the Cisco Meeting WebRTC app rather than using the Cisco Meeting App thick clients (Windows, Mac and iOS).

Unless otherwise noted, these instructions apply equally to combined or split deployments. These deployment instructions assume that you have already followed the appropriate Installation Guide for your Meeting Server.

CAUTION: This caution does not apply if you have not enabled XMPP servers or if you are only using combined host servers.

If you have enabled an XMPP server on an Edge server and you now intend to move it to a Core server, then you need to do the following steps in this order:

1. Disable the XMPP server(s) on the Edge server using the **`xmpp disable`** MMP command
2. Enable XMPP server(s) on the Core server (see the first two sections below)
3. Enable the Load Balancer on the Edge server (see Deploying the Trunk and the Load Balancer).

7.1 Configuring DNS Records for the XMPP Server

Note: The remainder of this section assumes that you have already followed the steps in Section 4 and configured the XMPP server.

DNS SRV records are required for any location in which it is desired to access the XMPP service. Depending on the configuration of your deployment, these can include internal DNS records within the LAN which resolve directly to the XMPP server, or public DNS records which resolve to the Load Balancer in the DMZ.

Note: You can configure the DNS resolver(s) to return values which are not configured in DNS servers or which need to be overridden; custom Resource Records (RRs) can be configured using the **`dns rr`** command which will be returned instead of querying DNS servers. However, this has the disadvantage that when you start adding additional XMPP servers you must add these records on each server. See the MMP Command Reference for details.

1. Create DNS A records for the fully qualified domain name (FQDN) of the XMPP service's host server and set it to the IP Address of the interface that the XMPP server is configured to listen on.

   In order to enable client connections, including use of the WebRTC Client, an _xmpp-client._tcp record is required. On a typical deployment, this will resolve to port 5222. Inside the LAN, if the core server is directly routable, it can resolve to the XMPP service running on the core server.

   For example:

   _xmpp-client._tcp.example.com could have the following SRV records:

   _xmpp-client._tcp. example.com 86400 IN SRV 10 50 5222 core1. example.com

   In locations where it is not possible to route to the core server, XMPP traffic should instead be handled by the Load Balancer, running on an Edge server. From the example above:

   _xmpp-client._tcp. example.com 86400 IN SRV 10 60 5222 edge1. example.com

   _xmpp-client._tcp. example.com 86400 IN SRV 10 20 5222 edge2. example.com

   _xmpp-client._tcp. example.com 86400 IN SRV 10 20 5222 edge3. example.com

   where **edge1. example.com** is in the DMZ, and therefore accessible from the Internet.

   If multiple geographically distributed edge servers are available, GeoDNS can be used in order to tell clients to favor Edge servers close to them.

   The XMPP service can federate with any other standards-compliant XMPP service. In order to enable this, create an _xmpp-server._tcp SRV record, usually resolving to port 5269. Because XMPP federation will usually be across the Internet, typically these records are only required to point to servers available in the DMZ. For example:

   _xmpp-server._tcp. example.com 86400 IN SRV 10 60 5269 edge1. example.com

   _xmpp-server._tcp. example.com 86400 IN SRV 10 20 5269 edge2. example.com

   _xmpp-server._tcp. example.com 86400 IN SRV 10 20 5269 edge3. example.com

   ---

   Note: The 60 and 20 in the above examples are priorities. DNS SRV records have a priority field and clients try those servers with the lowest priority first. In addition there is a weight field within a given priority. For a weight of "2, 1, 1", the client will choose the server with weight "2" 50% of the time, and the servers with weight "1" will each be chosen 25% of the time).

   ---

2. Test the above with the following commands:

   nslookup -q=srv _xmpp-server._tcp.example.com

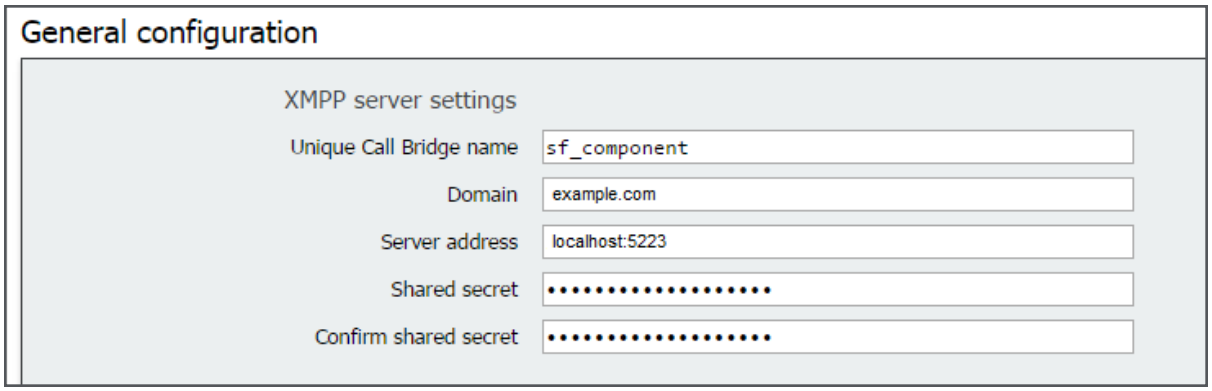
```
nslookup -q=srv _xmpp-client._tcp.example.com
```

7.2 Connecting Call Bridges to the XMPP Server

Note: When using Call Bridge Groups, either all of the Call Bridges in a Call Bridge Group, or none of them, should be added in this step.

To allow a Call Bridge to access the XMPP server:

1. Log in to the Web Admin Interface of the Call Bridge
2. Go to **Configuration > General**



3. Complete the fields in the XMPP Server Settings section.
   - Unique Call Bridge name (this is the component name set up previously, no domain part is required, as shown):

     cb_london
   - Domain (this is the XMPP server domain set up previously:

     example.com
   - Server Address is the IP address or hostname of the XMPP server, with an optional <port> (default is 5223):

     localhost:5223
   - Shared secret: as generated during the XMPP server configuration (see step 9 in Section 4).
4. Save your configuration by selecting **Submit** at the bottom of this page.
5. Go to **Status > General** and verify the server connection. You should see details similar to the following:

System status

Uptime	8 hours, 4 minutes, 18 seconds
Build version	1.8.3
XMPP connection	connected to localhost (secure) for 8 hours, 4 minutes, 16 seconds
Authentication service	registered for 8 hours, 4 minutes, 16 seconds

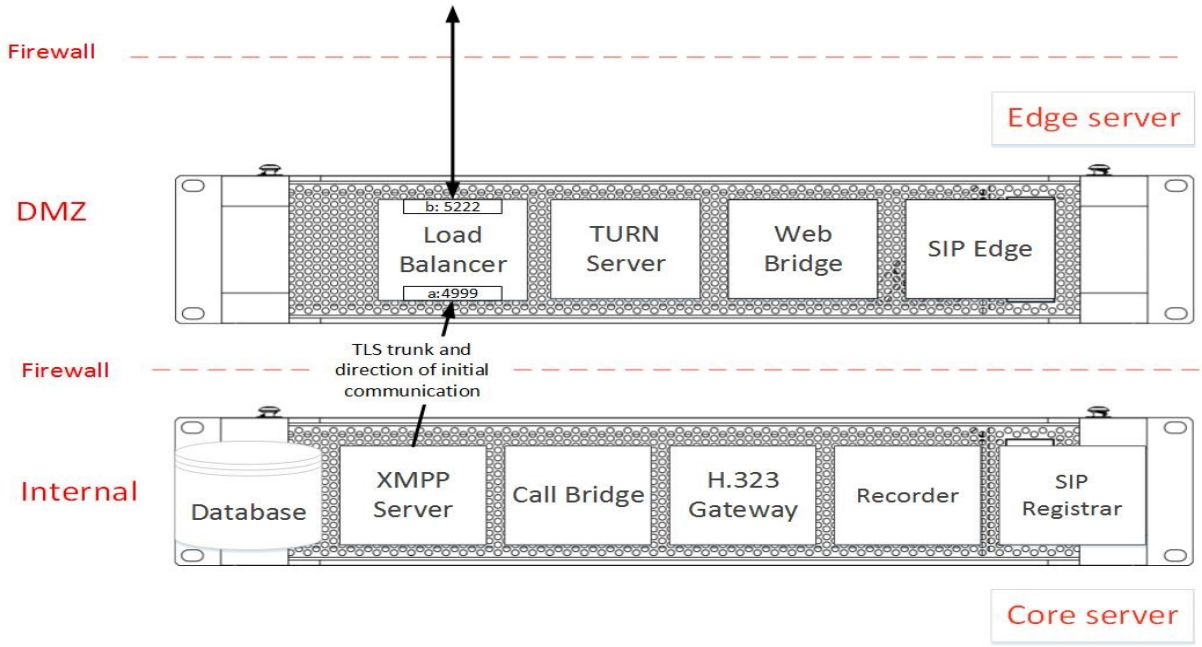
See the MMP Command Reference for details of the **`xmpp callbridge`** commands.

7.3 Deploying the Trunk and the Load Balancer

In a split deployment the XMPP server is located on the Core server for security, and connects via a Load Balancer on an Edge server. The Core server initiates a TLS connection to the Edge server. The Core server and Edge server mutually authenticate, and the Edge server starts to listen on port 5222 for incoming client XMPP connections. A client XMPP connection is serviced by the Load Balancer and relayed to the Core server using the TLS trunk.

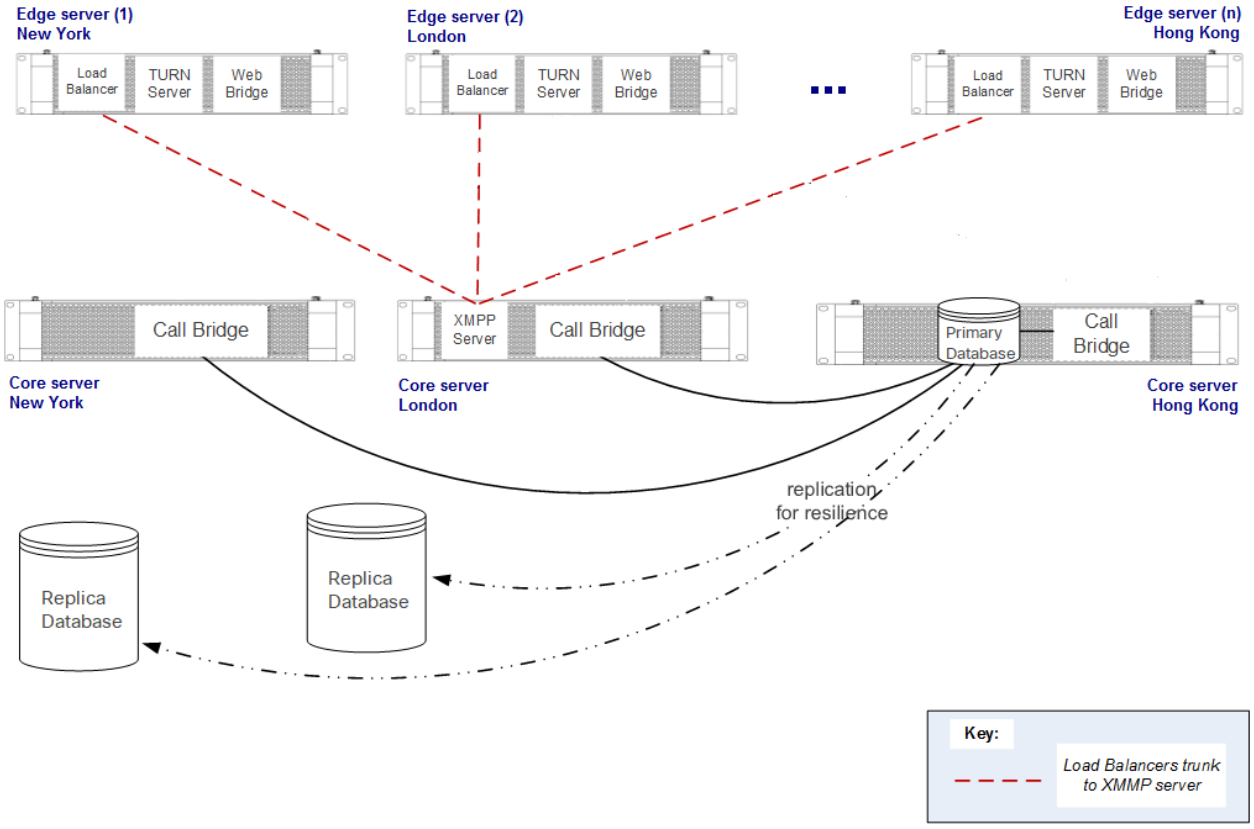
Note: the direction of traffic: the Core server connects out from the internal network to the Edge server in the DMZ–which is the natural direction of traffic as far as firewalls are concerned and convenient if the internal network uses NAT.

Figure 25: Trunk and Load Balancer listening ports in a split deployment



A Core server hosting the XMPP service can create multiple trunks to multiple Edge servers in different geographic locations.

Figure 26: Load balancers configured in a split deployment



To configure the Load Balancer you need:

- Network interfaces and ports to use for public connections
- Network interface and port to use for trunk connections
- Simple name (tag) for the Edge server
- Key and the corresponding certificate (and perhaps a bundle of CA-signed and intermediate certificates) for the trunk TLS connection
- Bundle of trusted certificates for authentication of the trunk connections

To configure a trunk you need:

- Domain name or IP address of the Edge server to trunk to
- Edge port to trunk to
- Service which will use the trunk (i.e. XMPP) or the port used by the service
- Simple name (tag) for the trunk
- Key and the corresponding certificate for the trunk TLS connection
- Bundle of trusted certificates for authentication of the trunk connections

If you are simply reconfiguring the Load Balancer and Trunk then follow the steps in Section 7.3.1 . However if you are setting up the Load Balancer and Trunk for the first time and need to create certificates, then follow the instructions in the Certificate guidelines. The Certificate guidelines details the same steps as in Section 7.3.1 but also covers information on creating, uploading and assigning certificates to the Load Balancer and the Trunk.

After following the steps you will have deployed a Trunk between the Core server and the Edge server.

7.3.1 Reconfiguring the Load Balancer and trunk

Note: This section assumes that you have already installed certificates for the Load Balancer and trunk on the Meeting Server.

1. SSH into the MMP of the Edge server
2. Create an Edge instance, using the command:

   `loadbalancer create <tag>`

   If the tag for the Edge server is "Edge1to LB", type:

   `loadbalancer create Edge1toLB`
3. Assign the private key/certificate pair to the Load Balancer and the trunk's certificate using the command:

   `loadbalancer auth <tag> <keyfile> <certificatefile> <trust-bundle>`

   where **`keyfile`** and **`certificatefile`** are the filenames of the matching private key/certificate pair for the Load Balancer, and <trust-bundle> is the certificate for the trunk.

   For example:

   `loadbalancer auth Edge1toLB edge1.key edge1.crt core1.crt`

   Note: The trunk certificate **`core1.crt`** is added as a 'trust bundle' to the Edge server
4. Configure the trunk interface and port, using:

   `loadbalancer trunk <tag> <iface>:<port>`

   for example, if the trunk connection will be allowed on interface A, port 4999, then type:

   `loadbalancer trunk Edge1toLB a:4999`
5. Configure the public interface and port (for accepting client connections), using

   `loadbalancer public <tag> <iface:port allowed list>`

   for example, if client connections are to be allowed on B, port 5222, then type:
   `loadbalancer public Edge1toLB b:5222`

6. In a common Edge server deployment, the Web Bridge is also enabled and needs to make use of the trunk. To allow this, configure the loopback as a public interface, e.g.
   loadbalancer public Edge1toLB b:5222 lo:5222
7. Enable the trunk, using:

   loadbalancer enable <tag>

   for example
   loadbalancer enable Edge1toLB

   Note: The public port is not opened until there is a trunk to service the connection.

8. SSH into the MMP of the Core server
9. Create a trunk between the Core and Edge server for xmpp traffic

   trunk create <tag> <port/service name>

   For example:
   trunk create trunktoEdge1 xmpp
10. Assign the private key/certificate pair to the trunk and the Load Balancer's certificate using the command:

    trunk auth <tag> <key-file> <cert-file> <trust-bundle>

    where **keyfile** and **certificatefile** are the filenames of the matching private key/certificate pair for the trunk, **<trust-bundle>** is the certificate for the Load Balancer.

    For example:

    trunk auth trunktoEdge1 core1.key core1.crt edge1.crt

    Note: The Load Balancer certificate **edge1.crt** is added as a 'trust bundle' to the Core server

11. Configure the Edge server that this trunk will connect to, using:

    trunk edge <tag> <edge name/ip address> [<default port>]

    For example, if the Edge server name is edge1.example.com using port 4999, then type:

    trunk edge trunktoEdge1 edge1.example.com 4999

    Note: If the domain name resolves to multiple IP addresses, a connection will be attempted to all.

12. Enable the trunk interface

    trunk enable <tag>

    For example:

    trunk enable trunktoEdge1

Note: To see the full list of Load Balancer and trunk commands, refer to the Cisco Meeting Server MMP Command Reference.

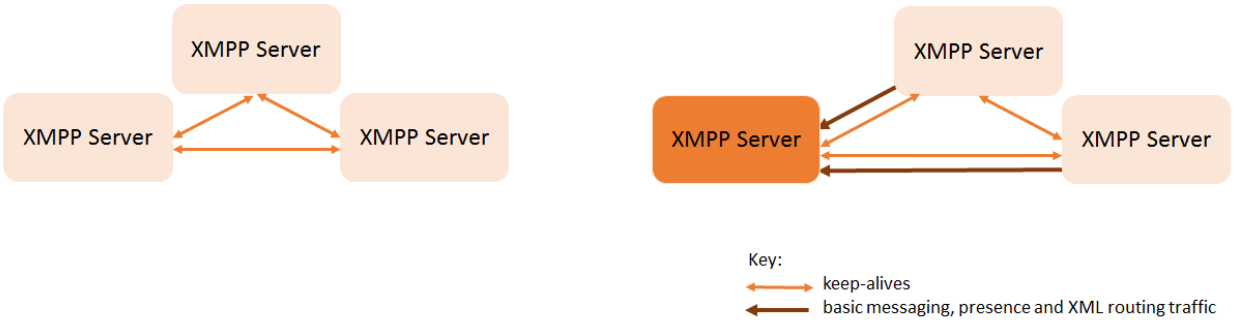
7.4 Support for XMPP resiliency

XMPP resiliency provides fail-over protection for a client being unable to reach a specific XMPP server. XMPP resiliency can be configured in multi-server deployments where there are three XMPP servers in the deployment.

Note: Deployments with only two XMPP servers will not benefit from resiliency, and if one fails it will cause an outage, effectively doubling the risk of failure versus stand-alone mode. This is due to the failover algorithm requiring more than half of the nodes to be available in order for the system to make good decisions about which XMPP server is the primary node.

When setup in resilient mode, the XMPP servers within a deployment are loaded with the same configuration. Each knows the location of the others and they establish links between them. They use keep-alive messages to monitor each other and elect a primary node. XMPP messages can be sent to any server. Messages will be forwarded to the primary XMPP server, see Figure 27. The XMPP servers continue to monitor each other, if the primary node fails then a new node is elected as the primary node and the other XMPP servers forward traffic to the new primary node.

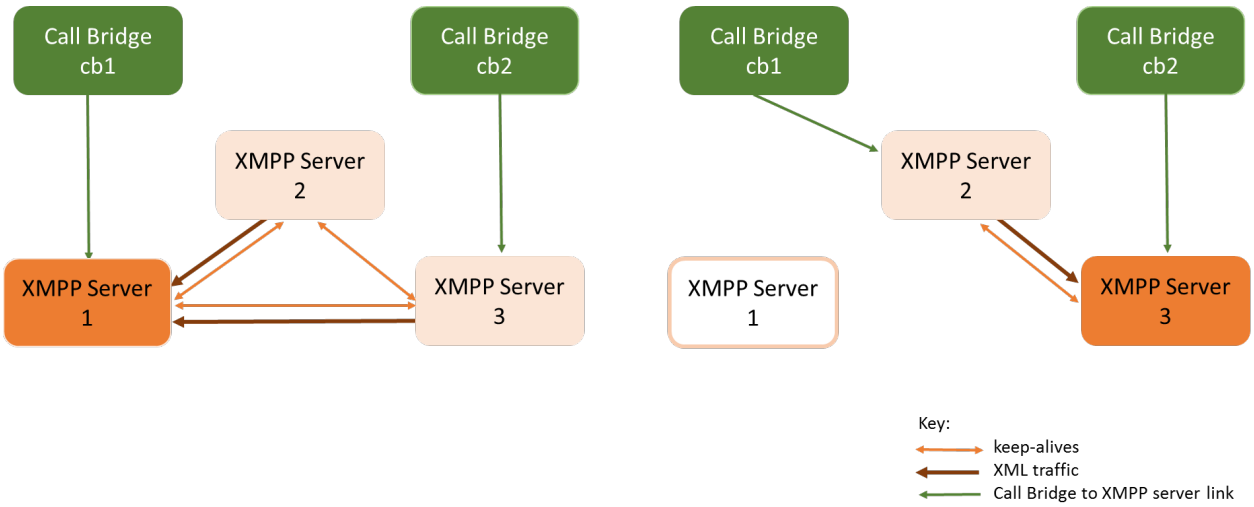
Figure 27: XMPP servers electing the primary node



Note: There is a network latency limitation (or Round Trip Time) of 200 ms or less between the XMPP servers in a cluster.

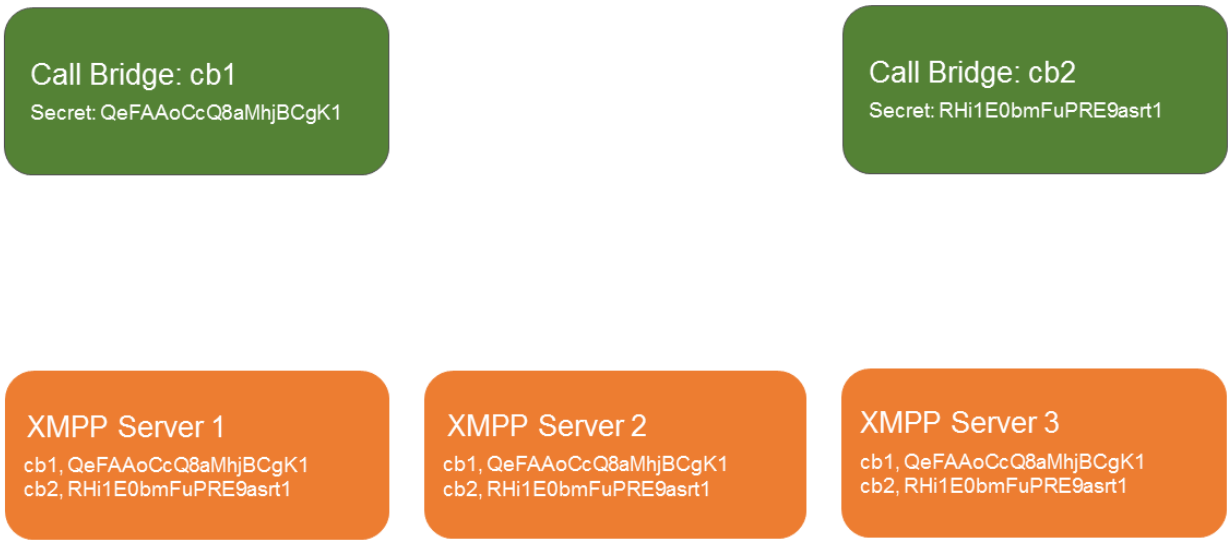
In a resilient deployment, the XMPP server that a Call Bridge connects to is controlled via DNS. This choice is based on the DNS priority and weight given. A Call Bridge only connects to one XMPP server at a time. There is no requirement for all Call Bridges to connect to the same XMPP server since all traffic is forwarded to the primary node. If a network problem results in the Call Bridge losing connection to the XMPP server, the Call Bridge will attempt to reconnect to another XMPP server, see Figure 28.

Figure 28: Call Bridges connecting to XMPP servers



The Call Bridge must be configured on any XMPP Server that it can connect to. This involves adding the name and secret of each Call Bridge to each XMPP Server, see Figure 29.

Figure 29: Call Bridge names and secrets added to XMPP servers



To summarize:

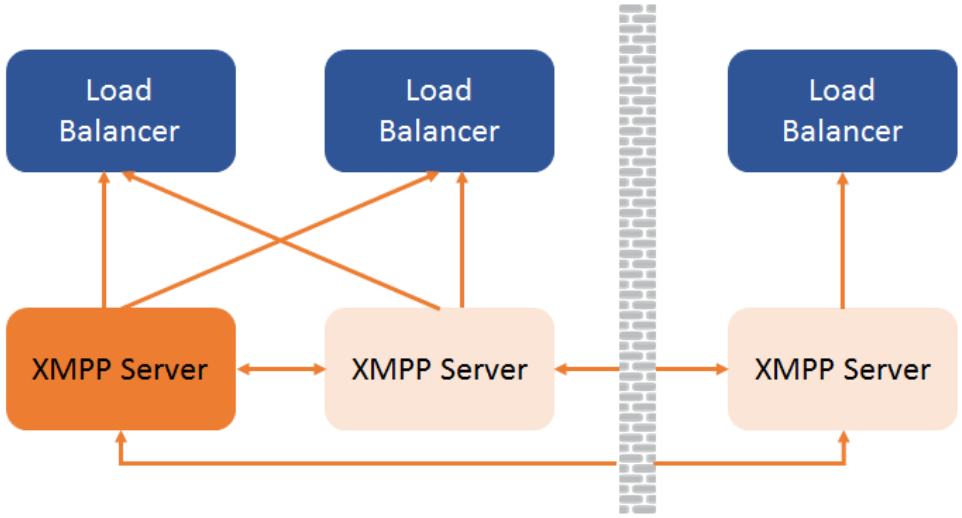
- Each Call Bridge is set up with one account consisting of its Call Bridge name and shared secret that the XMPP servers know about.

- Each XMPP server in the cluster needs a list of these Call Bridge accounts. This list needs to be the same on each XMPP server so that all Call Bridges can connect, it is important that the secret is set correctly when adding the Call Bridges to the 2nd and 3rd XMPP servers.

The example in section Section 7.4.1 explains the process for configuring the Call Bridges and XMPP servers to setup XMPP resiliency.

Clients connect via the load balancer. Each load balancer can be configured to have trunks from one or more XMPP servers. There is no need for a full mesh of connections, see Figure 30. In cases where the load balancer doesn't have a trunk from the primary XMPP server, the traffic is routed internally via another XMPP server.

Figure 30: Load Balancers connecting to XMPP servers



7.4.1 Example of deploying XMPP resiliency

This example of deploying XMPP resiliency uses three XMPP servers, and assumes that you are configuring the XMPP servers for the first time. If you have previously deployed XMPP onto an existing server, then steps 1-4 only need to be done on the new servers. Similarly, steps 8-10 and steps 23-25 may have previously been carried out. There is no need to repeat these steps.

If XMPP resiliency has previously been deployed then it is recommended to reset all servers apart from the first when following these instructions.

CAUTION: We recommend that you remove nodes from the cluster before resetting them. If you do an XMPP reset on a node while it is still in the cluster and then re-join the node to the existing XMPP cluster it will create a duplicate entry of that node when you check the status using **`xmpp cluster status`**. This can cause issues in a resilient setup.

For each XMPP server node, establish an SSH connection to the MMP, log in and at the **`cms>`** prompt, follow steps 1 to 4

1. If this node has previously been used for XMPP, reset the XMPP configuration on this node.
   xmpp reset
2. Create a component secret for this XMPP server
   xmpp domain <domain-name>
3. Setup the interfaces for the XMPP server to listen on
   xmpp listen <interface allowed list>
4. Assign the private key, and the certificate file and certificate bundle as provided by your Certificate Authority and required for connection to Cisco Meeting Apps.
   xmpp certs <keyfile> <certificatefile> [<cert-bundle]

   Where keyfile and certificatefile are the filenames of the matching private key and certificate. If your CA provides a certificate bundle then also include the bundle as a separate file to the certificate.

   For example:
   xmpp certs xmppserver1.key xmppserver1.crt xmppserverbundle.crt

If you used the same certificate on all boxes then skip steps 5 and 6, and use the certificate itself as the allowed list (in step 7).

5. Using SFTP, download and combine all of the XMPP server certificates into one bundle of allowed listed certificates, in one of the following ways:
   - Linux or UNIX-like Operating Systems:

     cat xmppserver1.crt xmppserver2.crt xmppserver3.crt > xmpp-cluster-allowed-list.crt
   - Windows or DOS:

     copy xmppserver1.crt + xmppserver2.crt + xmppserver3.crt xmpp-cluster-allowed-list.crt
   - Manually combine the certificates using Notepad or Notepad++. There must be no spaces on the first certificate's "END CERTIFICATE" line and the second (and further certificate's) "BEGIN CERTIFICATE" line, but there MUST be a carriage return at the end of the file. They MUST also be in Base64 encoded format.
6. Using SFTP, upload **xmpp-cluster-allowed-list.crt** to all of the Meeting Server.
7. On each node, assign the bundle of allowed listed certificates from step 6. For example:

   xmpp cluster trust xmpp-cluster-allowed-list.crt

   or if you have used a single certificate for multiple servers:

   xmpp cluster trust xmppserver.crt

Set up clustering on one of the nodes with an XMPP server. Follow steps 8 to 14 on this node:

8. Enable the XMPP server node
   xmpp enable

9. Configure one of the Call Bridges that will connect to the XMPP cluster:
   xmpp callbridge add <callbridge name>

   A secret is automatically generated; for example:

   ```
   cms>xmpp callbridge add cb1
   Added callbridge: Secret: QeFAAoCcQ8aMhjBCgK1
   ```

   this configures the XMPP server to allow connections with the Call Bridge named cb1.

   Note the domain, Call Bridge name and secret generated, you will need this information later when you configure the Call Bridge access to the XMPP server (so that the Call Bridge will present the authentication details to the XMPP server)

10. Repeat step 9 for all of the Call Bridges that will connect to the XMPP cluster. **Each Call Bridge will need to have a unique name.**

    Note: If you have not already noted the details for the Call Bridges that you have added to the XMPP server, then use the command:
    xmpp callbridge list
    and make a note of the domain, name and secret generated for each Call Bridge that will connect to the XMPP cluster. These are required in step 17.

    Note: When using Call Bridge Groups, either all of the Call Bridges in a Call Bridge Group, or none of them, should be added in this step.

    ---

11. Disable this XMPP server node
    xmpp disable
12. Enable the XMPP cluster on this node
    xmpp cluster enable
13. Initialize the XMPP cluster on this node. This command will create a 1 node xmpp cluster, the other nodes (xmpp servers) will be joined to this cluster in step 22.
    xmpp cluster initialize
14. Re-enable this node
    xmpp enable
15. Make a note of the IP address of this node, this is required in step 22.
    xmpp cluster status

Follow steps 16 to 22 for each of the remaining XMPP server nodes in the cluster (nodes 2 and 3):

16. Enable the XMPP server node
    xmpp enable
17. Add each Call Bridge to this node. This requires the Call Bridge to be added using the same Call Bridge name and secret from the first XMPP server node. This is achieved using the command:
    xmpp callbridge add-secret <callbridge name>
    Enter

```
callbridge secret>
```

18. Repeat step 17 until you have added all of the Call Bridge secrets to this node.
19. Disable this node
    ```
    xmpp disable
    ```
20. Enable the XMPP cluster on this node
    ```
    xmpp cluster enable
    ```
21. Re-enable this node
    ```
    xmpp enable
    ```
22. Join this node to the cluster, where **<cluster>** is the IP address or domain name of the node from step 15.
    ```
    xmpp cluster join <cluster>
    ```

Now configure each Call Bridge with the authentication details of the XMPP servers in the cluster. This enables the Call Bridges to access the XMPP servers.

23. Log in to the Web Admin interface on one of the Call Bridges
24. Navigate to **Configuration>General** and enter the following:
    - unique Call Bridge name as used in steps 9 and 10, no domain part is required for example
      ```
      cb1
      ```
    - domain: this is the XMPP server domain
      ```
      example.com
      ```
    - server address of the XMPP server. Set this field if you want this Call Bridge to only use a co-located XMPP server, or you don't have DNS configured. Using the co-located XMPP server reduces latency. Leave this field empty to allow this Call Bridge to failover between XMPP servers, this requires the DNS entries to be setup, see the Note below.

Note: Refer to Section 7.1 for advice on setting up DNS records for the XMPP server nodes. If you plan to use DNS to connect between Call Bridges and XMPP servers you will also need to set up a DNS SRV record for the xmpp cluster to resolve to the DNS A record of each of the XMPP servers in the cluster. The format of the DNS SRV record is: _ **_xmpp-component._tcp.<domainNameofXMPPserver>**

For example:

```
_xmpp-component._tcp.example.com. 86400 IN SRV 0 0 5223
xmppserver1.example.com
_xmpp-component._tcp.example.com. 86400 IN SRV 0 0 5223
xmppserver2.example.com
_xmpp-component._tcp.example.com. 86400 IN SRV 0 0 5223
xmppserver3.example.com
```

The example above specifies port 5223 (use another port if 5223 is already used).

- the shared secret used for this Call Bridge as used in steps 9 and 10
- and confirm the shared secret

25. Repeat steps 23 and 24 for each Call Bridge that will connect to the cluster.

Your cluster should now be running. You can check with:

```
xmpp cluster status
```

you will get a report on the live state of the xmpp cluster. If the cluster has failed, then this command will return the statistics of the xmpp server running on this Meeting Server only. Use this command to try and help diagnose connectivity problems.

7.4.2 Identifying issues within an XMPP cluster

Table 8 below indicates possible causes to symptoms that might be seen in your syslog sever.

Table 8: Troubleshooting issues with XMPP clustering

Symptom	Possible cause
Many TCP failures	Trust between your XMPP servers may not have been set up correctly, or there are connectivity issues between nodes. Test that each node can reach port 5222.
Client logged me out	Failover detection is set at 10 seconds which occasionally is enough time to cause a client to log out. Cisco Meeting Apps will re-login automatically, unless Call Bridges are not re-connecting. However, WebRTC clients will need to re-login.
Frequent elections	Frequent elections of the primary node in the cluster may indicate problems on your network. If the primary node does not hear from the other, secondary nodes, within 10 seconds then it steps down from being the primary node. Similarly, secondary nodes will stand for election if they do not hear from the primary node in a random time between 10 and 20 seconds (the random time prevents multiple secondarye nodes assuming the primary role).

7.4.3 Maximum number of concurrent XMPP clients supported by the Meeting Server

CAUTION: The maximum number of concurrent XMPP clients supported by the current Meeting Server software is 500. This maximum is a total number of all different clients (Cisco Meeting App, WebRTC Sign-in and WebRTC Guest clients) registered at the same time to clustered Meeting Servers. If the number of concurrent XMPP registrations exceeds 500 sessions, some unexpected problems with sign in may occur or it may lead to a situation where all currently registered users need to re-sign in, this can cause a denial of service when all users try to sign in at the same time.

7.4.4 XMPP server certificate validation

From version 2.4.0, the Call Bridge and Web Bridge have trust stores to hold the certificates for the XMPP servers in the deployment. If configured, these trust stores enable the Call Bridge and Web Bridge to check the identity of the XMPP servers when making connections to them. Validating the certificate files of the XMPP servers ensures that XMPP servers are legitimate, and removes the risk that an attacker could redirect traffic to an insecure XMPP server. In addition, validation can be used to prevent the WebRTC app from being used to connect to meetings hosted by other Meeting Server deployments.

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

8 Deploying Web Bridge 2

Note: If you are not using the WebRTC Client, skip this chapter.

In version 2.9, both Web Bridge 3 and the original Web Bridge can co-exist on Meeting Server. However, the original Web Bridge (also referred to as Web Bridge 2) will be removed from a future version of Meeting Server.

Note: This chapter only applies to Web Bridge 2 and the WebRTC Client. If you wish to deploy Web Bridge 3 and use Cisco Meeting Server web app, see Appendix K.

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

8.1 Deploying multiple Web Bridge 2s

It is possible to deploy multiple Web Bridges in a Meeting Server cluster. Due to how browsers interact with the Web Bridge, care must be taken when deploying multiple Web Bridges to avoid problems.

To find a Web Bridge, the browser performs a DNS A record lookup on the Web Bridge URI specified, for example `join.example.com`. In deployments with multiple Web Bridges, the result of this could be multiple IP addresses.

Unless a policy is in place, the browser can pick any of the DNS A records and switch between them during a session . This may result in the Web RTC clients rehoming across Web Bridges and causing connections to drop. The consequence of this will be that the user has a very poor user experience during the conference, with the call repeatedly dropping.

To avoid Web RTC clients rehoming, we recommend that you adopt one or more of the following practices:

1. Deploy the Expressway between the Meeting Server and the firewall, and use the Expressway CMS WebRTC proxy to load balance across the Web Bridges. See the guide Configure CMS WebRTC Proxy over Expressway.
2. Use a DNS solution that only returns a single IP address for each WebRTC client. There are two main ways of doing this:
   a. GeoDNS/GeoIP, where the source IP address is used to look up which server to use. This choice is based on where the user is, and requires a mapping of source IP address to location.

b. IP address hashing, where the source IP address is hashed to determine which record to return.

both use the source IP address to determine the result to return. Assuming the IP address of the WebRTC client doesn't change, then the client will always get the same DNS A record. Resiliency works by the DNS server switching the record it serves based on availability of the servers.

3. Use a load balancer (see note below) in front of the Meeting Servers. In this situation, point the DNS A record at the load balancer rather than the individual Web Bridges.

Note: HAproxy, a high performance TCP/HTTP load balancer, has been tested and has shown to work in 2 modes: source-ip load balancing, and cookie-insertion. Other load balancers are available, and Cisco is not recommending HAproxy.

---

4. Use the MMP command **`webbridge url-redirect`** as a workaround when deploying multiple Web Bridge 2s to configure the URL redirect location.

These methods can be combined for cases where the customer has multiple sites and multiple Web Bridges at a site, for example practices 1 and 2 would work, as would 2 and 3.

8.2 Setting up the Web Bridge 2s' certificates

Each Web Bridge needs a key and certificate pair that is used to establish TLS connections with Web RTC Clients. If you have not already done so, create, upload and assign certificates to the Web Bridges. Instructions are provided in the Certificate guidelines.

8.3 Setting up the Web Bridge 2s

If you have multiple Web Bridges we strongly suggest that you use the API to configure them , as described below. (An example is provided in Appendix 1.2.). However, there is one configuration setting that needs to be entered via the Web Admin Interface on each Web Bridge, this is the Web Bridge URI used to generate meeting invites and the cross launch URL for the Cisco Meeting App. Go to **Configuration > General**. In the **External access** section enter the **Web Bridge URI**.

8.3.1 Setting up the Web Bridge 2s via the API

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 only. Do not specify the URL of multiple Web Bridges, the Call Bridge needs to be able to open an https connection to a specific Web Bridge 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 will allow guests to access a space by 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 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 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, WebRTC client users can join scheduled Lync conferences by entering the Lync meeting id on the WebRTC signin page.
- tenants and tenant groups associated with the Web Bridge, if required,
- customization of the background image and logo on the landing page for the WebRTC Client, if required

See the section on Web Bridge Methods in the API Reference Guide for full details on configuring Web Bridges via the API.

Example:

1. Using the API on one Call Bridge, create a /webBridges/<web bridge id> node for each Web Bridge.
   a. For example, for each Web Bridge access the API as usual.
   b. POST to the Call Bridge's /webBridges node with the following values replaced by your values – each Web Bridge requires a unique URL:

      ```
      url = https://join.example.com

      resolveLyncConferenceIds= true
      ```

      Note: You do NOT need to sign into the Web Admin Interface **Configuration > General** to configure the Guest Account Jid Domain field and you should NOT fill in the Guest account client URI field. These fields only apply to single Web Bridges configured via the Web Admin Interface.

8.4 Web Bridge 2 call flow

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

1. PC web browser opens HTTPS connection to Web Bridge
2. User is prompted to **Join Call** (see step 3) or **Sign In** (see step 4)
3. If **Join Call** is selected, user is prompted to enter the Call ID and Passcode (if required)
    a. Web Bridge queries Call Bridge to validate Call ID and Passcode
    b. If successful, the User is prompted to enter a Name to be displayed in the call
    c. Upon completing these steps and clicking Join Call, the WebRTC app sends an http message to the Web Bridge on port 443, which requests temporary credentials from the Call Bridge over port 443
    d. Web Bridge then connects to the XMPP Server on port 5222, using the above temporary credentials, and the Call Bridge validates the credentials
    e. Call Bridge requests allocations from the TURN Server to use for this call on UDP 3478
    f. WebRTC app requests allocations from the TURN Server to use for this call on UDP 3478 (or TCP 443)
    g. If the UDP STUN packets sent by the WebRTC app to the TURN server are successful, the WebRTC app will send media from the TURN server, with a Media Port range of 32768-65535
    h. If the UDP STUN messages are un-successful, the WebRTC app will fall back and send messages to the TURN Server on TCP Port 443
    i. If the TCP connection is successful, Media will also be sent to the TURN Server on TCP Port 443
    j. The TURN Server will then relay this WebRTC Media to Call Bridge, converting to UDP if received as TCP from WebRTC app
4. If **Sign In** is selected, user is prompted to enter Username and Password
    a. Web Bridge will do DNS Lookup for the SRV record of _xmpp-client._tcp for the domain entered in the Username field
    b. Web Bridge connects to the XMPP Server returned in the DNS lookup and sends the Credentials as supplied for verification
    c. If Login is successful, the User is logged into the WebRTC app and is shown a Client view similar to the PC XMPP app
    d. Upon attempting a New Call or joining a Meeting, the app will connect as follows
    e. Web Bridge signals Call Bridge for the call request over XMPP

f. Call Bridge opens connections to the TURN Server to request allocations for ports to use for this call on UDP 3478

g. Once TURN allocations have succeeded, Call Bridge answers the call and sends the address and ports to use back to Web Bridge to be relayed to the WebRTC app

h. WebRTC app requests allocations from the TURN Server to use for this call on UDP 3478 (or TCP 443)

i. If the UDP messages are successful, the WebRTC app will send messages to the media port range of 32768-65535, using the specific ports relayed to it from Call Bridge

j. If the UDP messages are un-successful, the WebRTC app will fall back and send messages to the TURN Server on TCP Port 443

k. If the TCP connection is successful, media will also be sent to the TURN Server on TCP Port 443

l. The TURN Server will then relay this WebRTC media to Call Bridge, converting to UDP if received as TCP from WebRTC app

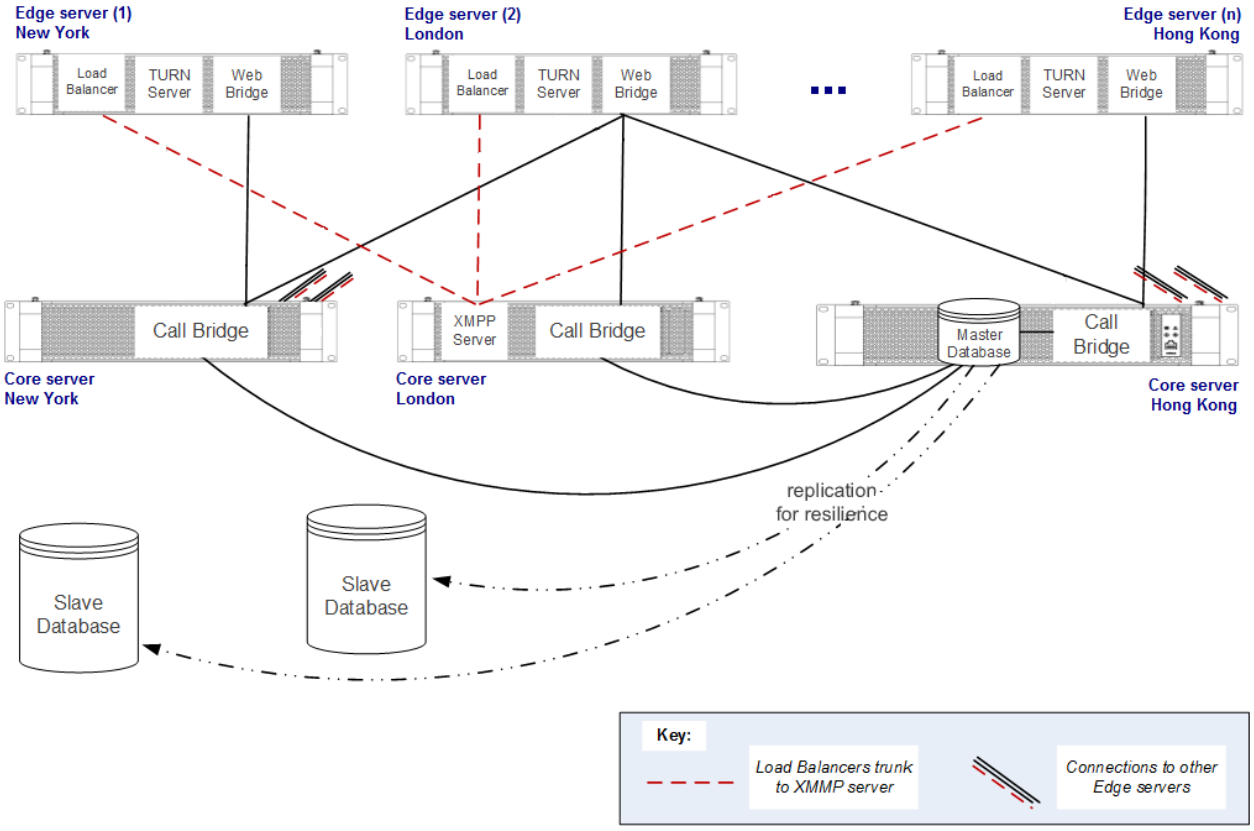
8.5 WebRTC Client Information

When a WebRTC Client user calls in:

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

The Web Bridge 2, Load Balancer and TURN server used for a call do not have to be co-located.

Figure 31: Web Bridges configured in a split deployment



8.6 Enabling HTTP redirect and the Web Bridge 2

The Web Bridge supports HTTPS. It will forward HTTP to HTTPS if configured to use "httpredirect". This is explained in Section 4

9 Deploying the TURN Servers

The TURN server listens on port 3478 for UDP. This is the normal port used by the Call Bridge to connect to it, and is also available for remote connections.

The TURN server can also listen on a second port for TCP and/or TLS, typically 443.

Although the configuration option for this is named "tls", TURN actually accepts UDP, TCP and TLS on this additional port.

If you need to use TCP connections to the TURN server from a Call Bridge then either:

- on the Call Bridge set the **`tcpPortNumberOverride`** for a TURN server to the port configured (see next section)

  or

- or change your firewall rules to open TCP port 3478 from the Call Bridge to the TURN server

---

Note: In a single combined server deployment, the TURN server will never listen on port 443 on the loopback interface.

The TURN server can also listen on a second port for TCP and/or TLS, typically 443.

Although the configuration option for this is named "tls", TURN actually accepts UDP, TCP and TLS on this additional port.

If you need to use TCP connections to the TURN server from a CallBridge then either:

- on the Call Bridge set the **`tcpPortNumberOverride`** for a TURN server to the port configured (see next section)

  or

- or change your firewall rules to open TCP port 3478 from the Call Bridge to the TURN server

---

Note: In a single combined server deployment, the TURN server will never listen on port 443 on the loopback interface.

9.1 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.

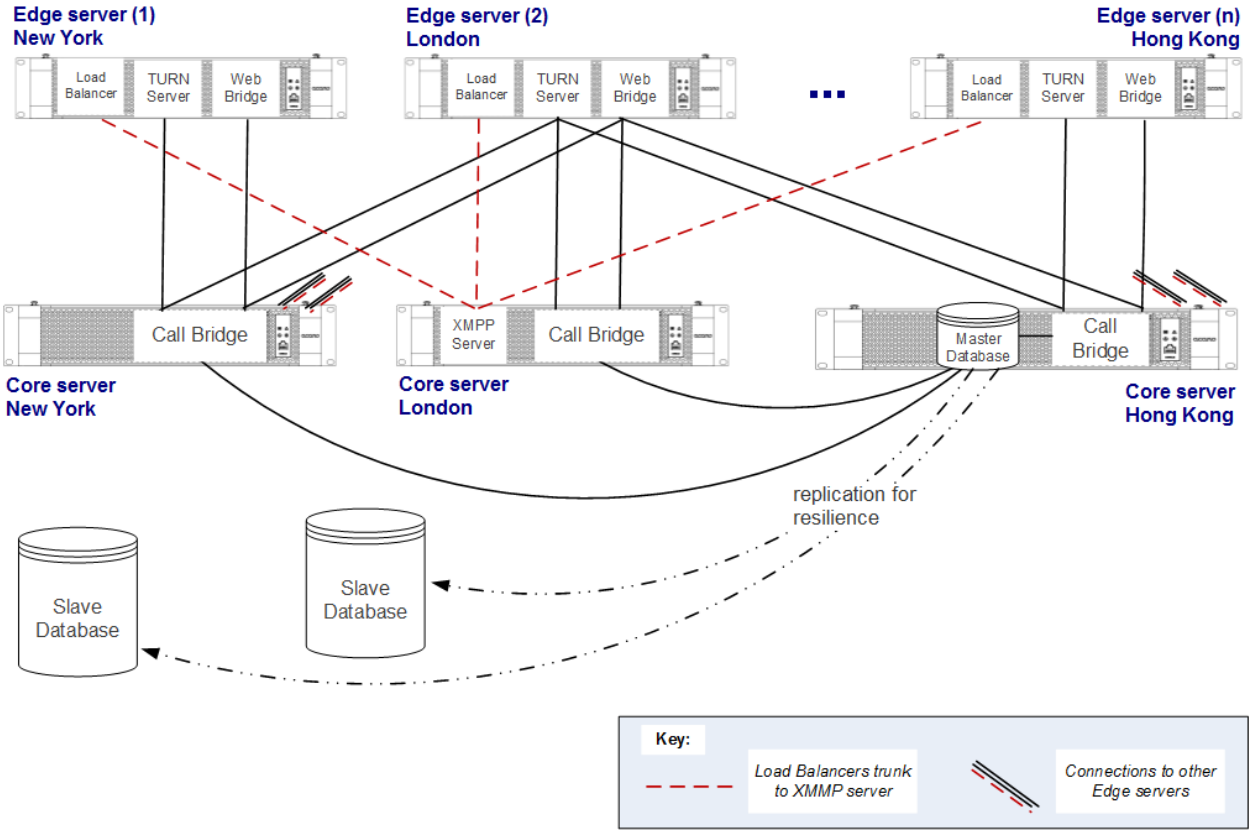
   For example, for the TURN server which will interwork the media, POST to the Call Bridge's /turnServers node the following parameter values replaced by your values:

   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 32 for the completed deployment.

Figure 32: TURN servers configured in a split deployment



10 spaces and the User Experience

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 Appendix 1.5.

Note: Every member of a space has the same joining experience, set by configuring the access methods, URI, passcode(s) etc. for the space.

Figure 33 and Figure 34 , each show two PC Client 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 PC Client in London using a TURN server and Load Balancer in a different location.

Figure 33: Two PC Clients making calls to different spaces in a combined deployment

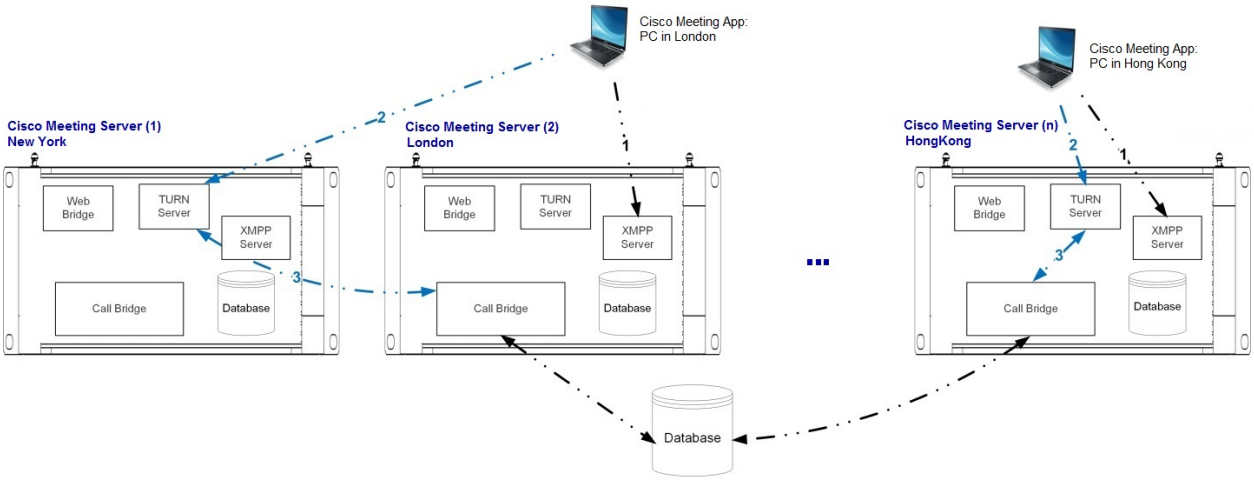
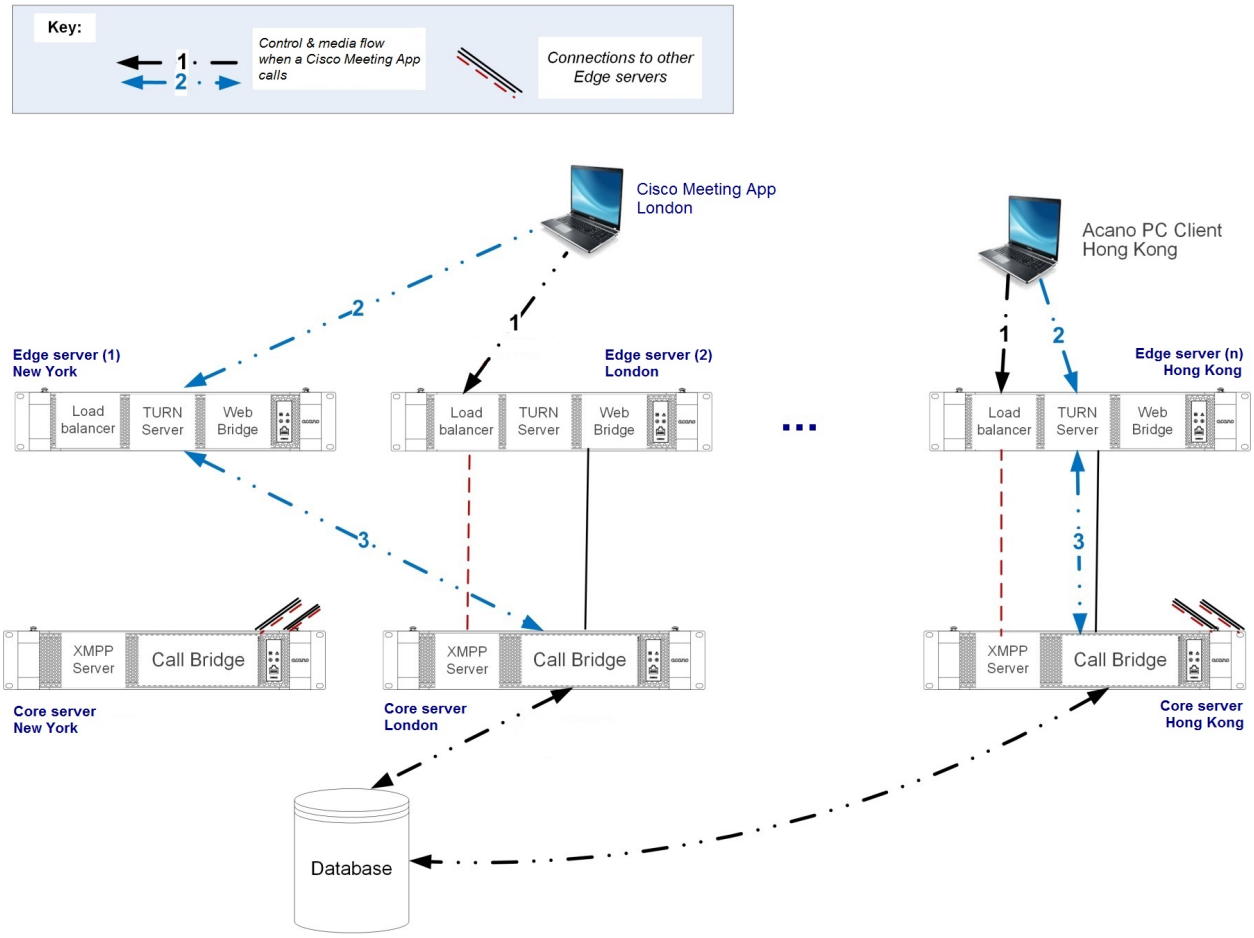


Figure 34: Two PC Clients making calls to different spaces in a split deployment



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 35 and Figure 36. 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". 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 35: Two Call Bridges calling space A in a combined deployment

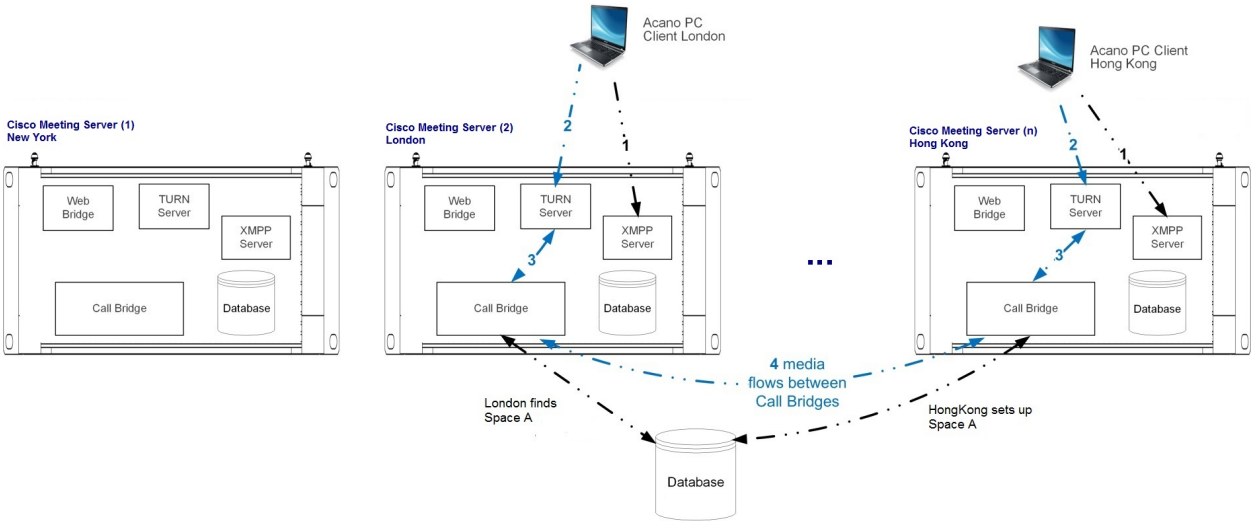
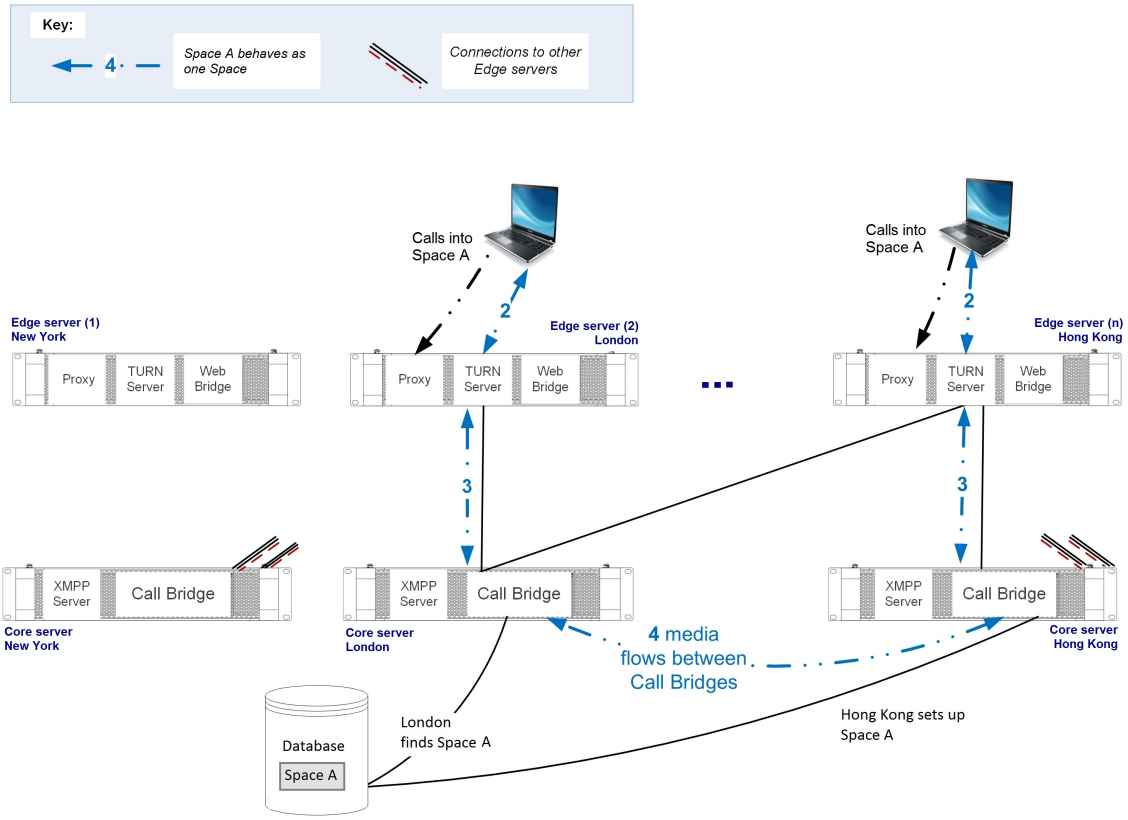


Figure 36: Two Call Bridges calling space A in a split deployment



Example views and media flows are shown in the two figures below with four participants in a space.

Figure 37: Example media flows and views when PC Clients dial into space A in a combined deployment

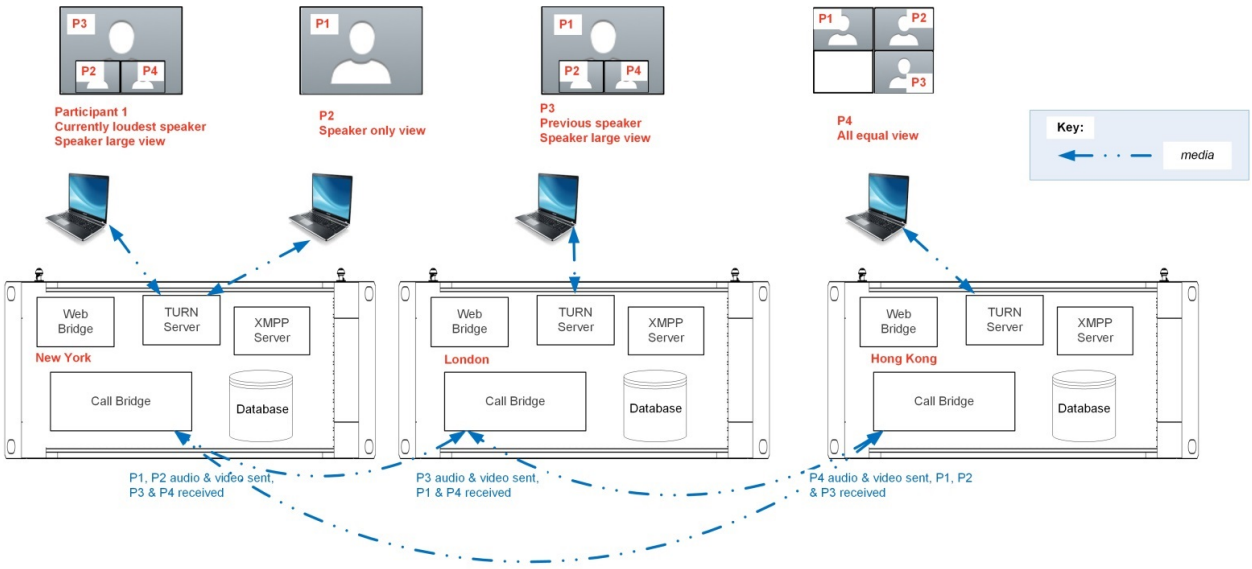
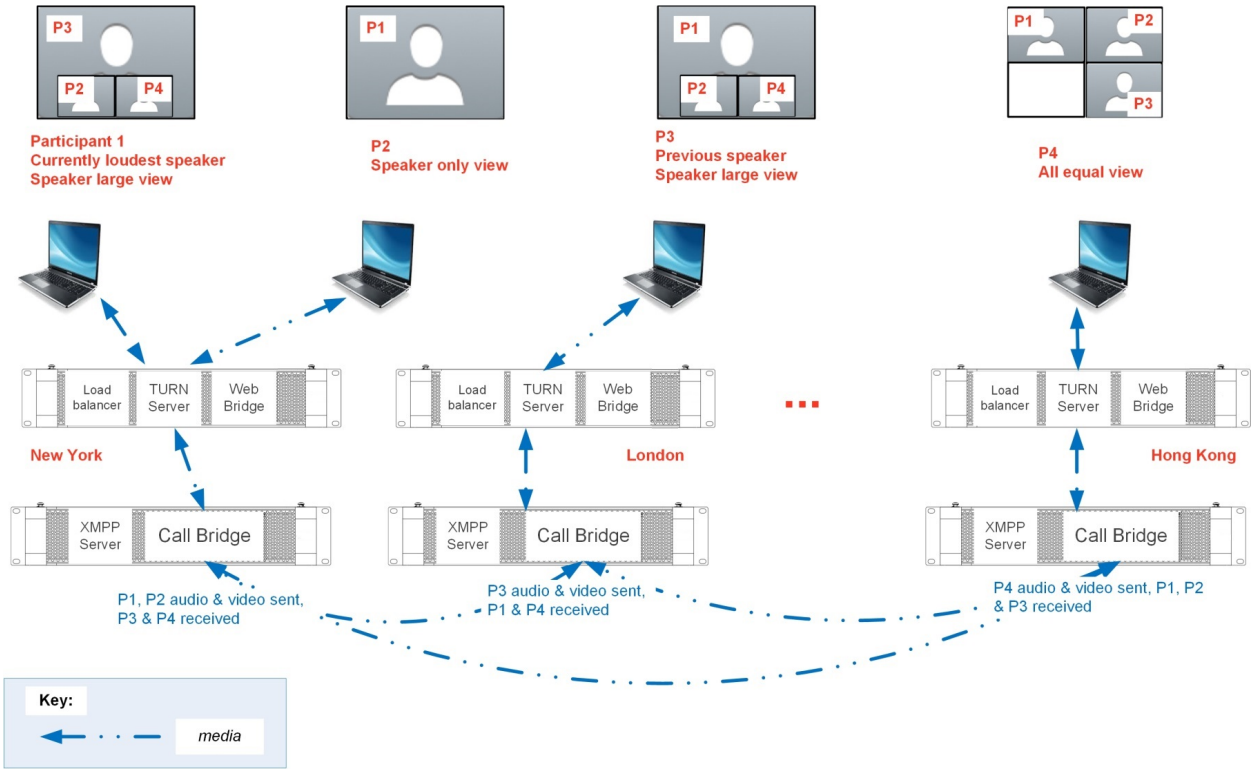


Figure 38: Example media flows and views when PC Clients dial into space A in a split deployment



10.1 Message board chat

For existing deployments that use chat message boards, chat will remain enabled when you upgrade to a later version of Meeting Server software. However for new deployments, or a deployment that did not previously use chat message boards, then the message board feature needs to be enabled. Create a new Call Profile to be applied to spaces.

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 **/callProfiles**
2. Click the **Create new** button
3. Set **messageBoardEnabled** = **true**
4. Configure other parameters as appropriate.
5. Click **Create**

11 Dial plan configuration – overview

11.1 Introduction

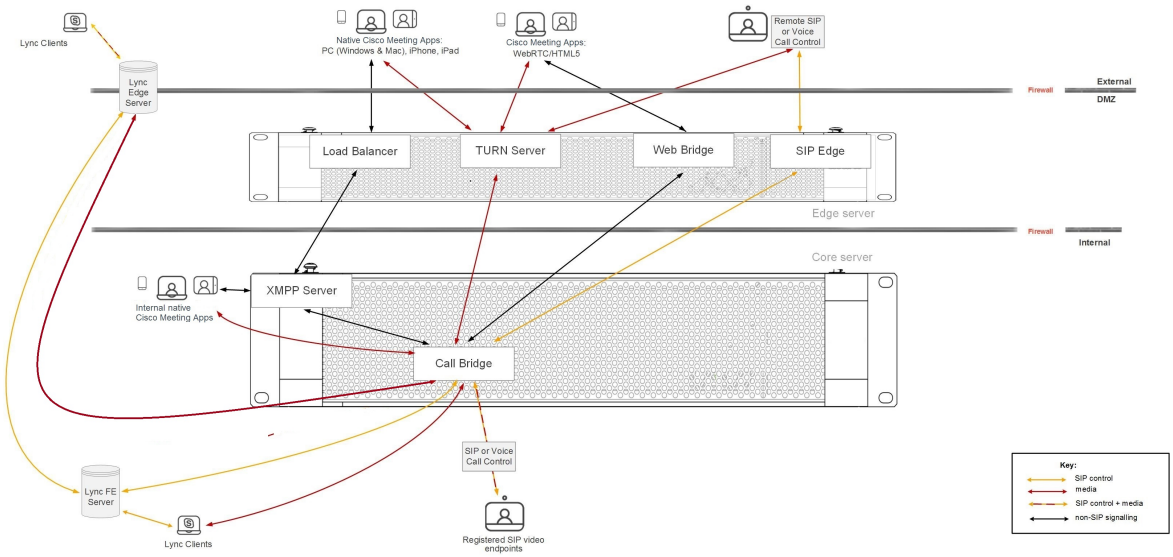
For the Meeting Server to be integrated in a SIP, Lync and voice environment, connections need to be set up from the SIP Call Control, Lync FE Server and Voice Call Control to the Meeting Server. Changes to the call routing configuration on these devices is required in order to correctly route the calls that require the Meeting Server.

Figure 39 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.

Note: Although this figure and subsequent diagrams in this Deployment Guide use an Acano X series deployment as the example, the instructions apply equally to virtualized deployments.

Figure 39: Example deployment for dial plan configuration



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 spaces, Cisco Meeting App users (native and WebRTC) and also SIP video endpoints. The subdomains vc.example.com (for SIP video endpoints) and meetingserver.example.com (for spaces and Cisco Meeting 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 Cisco Meeting 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 12 and Chapter 13 provide step-by-step instructions on configuring the total solution.

11.2 Dial plan rules for incoming calls and outbound calls.

This section explains how to use the Meeting Server API to configure the rules that determine how to handle incoming and outbound calls.

Note: The Single Combined and Single Split Depoyment Guides explain how to use the Web Admin interface to configure the dial plan rules. But scalable and resilent 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, Cisco Meeting 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

11.2.1 /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.
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.

Parameters	Type/Value	Description/Notes
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.
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.

11.2.2 /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
resolveToUsers	true|false	If set to true, calls to this domain will be matched against user JIDs (if a match is then found, that incoming call leg causes a "point to point" call to that user's Meeting App.
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 user JIDs and 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.

11.2.3 /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.

11.3 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/)	${/@example.com$/ .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.

12 Dial plan configuration – SIP endpoints

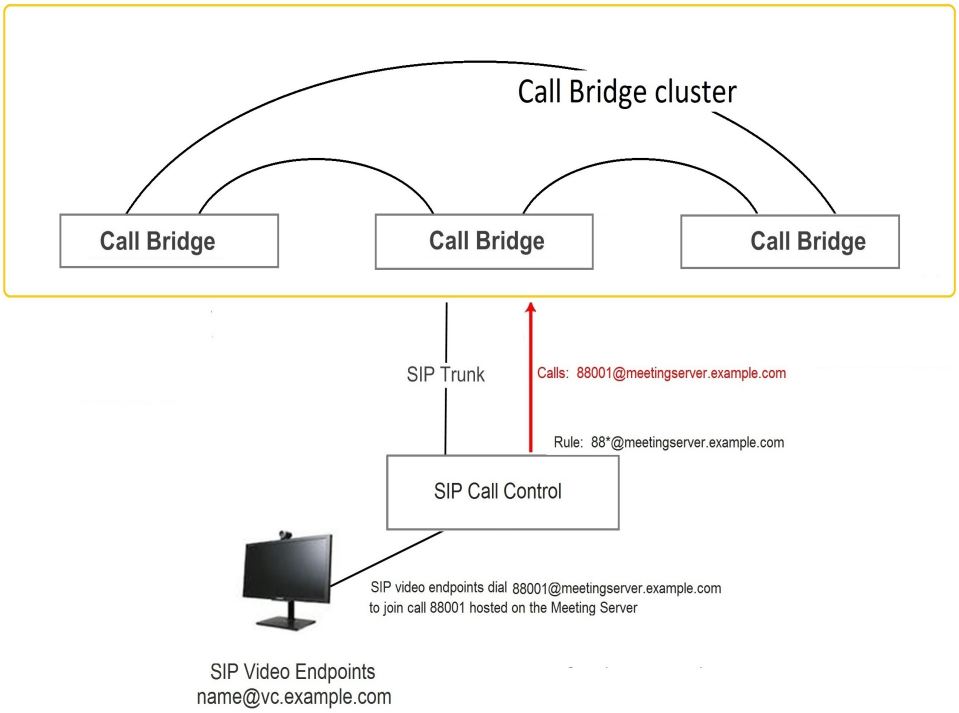
12.1 Introduction

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.

12.2 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 40: Example of SIP video endpoints calling into clustered Meeting Server hosted calls



12.2.1 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.

12.2.2 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 13.1.2

Note: More information on **/inboundDialPlanRules** is given in Section 11.2.2

---

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 13.1.2).

Note: More information on **/outboundDialPlanRules** is given in Section 11.2.1.

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 13, consider whether to:

- configure the media encryption setting, see Section 12.3,
- enable TIP support for Cisco CTS endpoints, see Section 12.4,
- configure an Interactive Voice Response (IVR), see Section 12.5

12.3 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.

12.4 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**.

12.5 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.

12.6 Next steps

Now follow the steps in Chapter 13 to configure dial plans to integrate a Meeting Server cluster with Lync deployments.

13 Dial plan configuration – integrating Lync/Skype for Business

Throughout this chapter, references to Microsoft Lync also mean Microsoft Skype for Business.

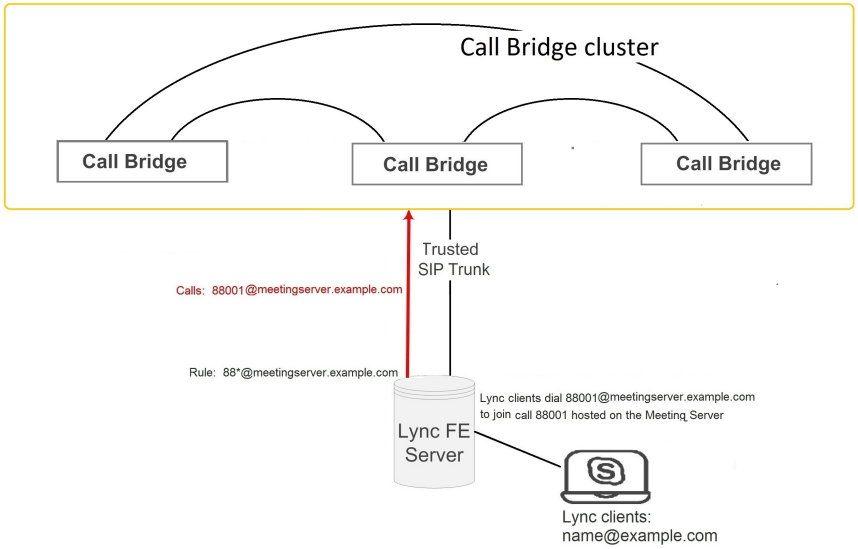
Note: For Call Bridge integration with Lync Edge, the Call Bridge needs its own login account; for a Call Bridge cluster integration with Lync, each Call Bridge in the cluster needs its own unique login account to Lync Edge.

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.

13.1 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 41: Example of Lync clients calling into meetings hosted on clustered Meeting Servers



13.1.1 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.

13.1.2 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 11.2.1.

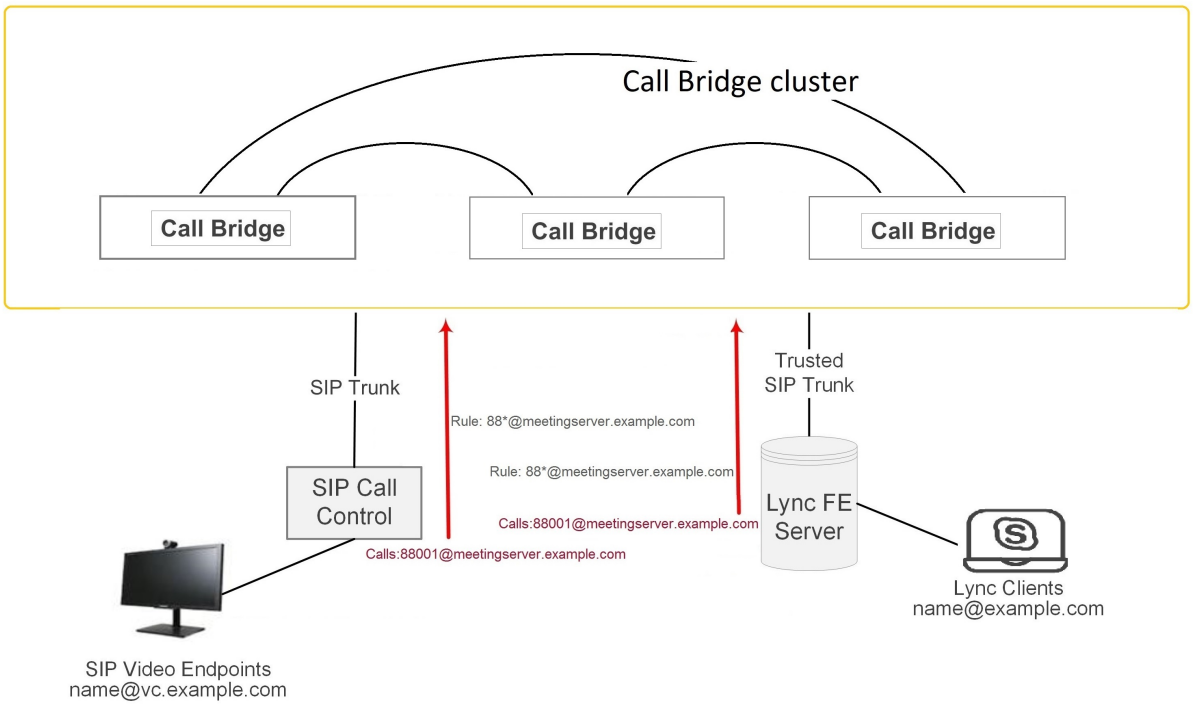
Lync clients can now dial into a call 88001 hosted on the clustered Meeting Server by dialing 88001@meetingserver.example.com.

13.2 Integrating SIP endpoints and Lync clients

To allow SIP endpoints to dial a Meeting Server space, implement the steps in Section 12.2; to allow Lync clients to dial a Meeting Server space, implement Section 13.1.

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 42: Example of SIP video endpoints and Lync clients calling into Meeting Server hosted meetings

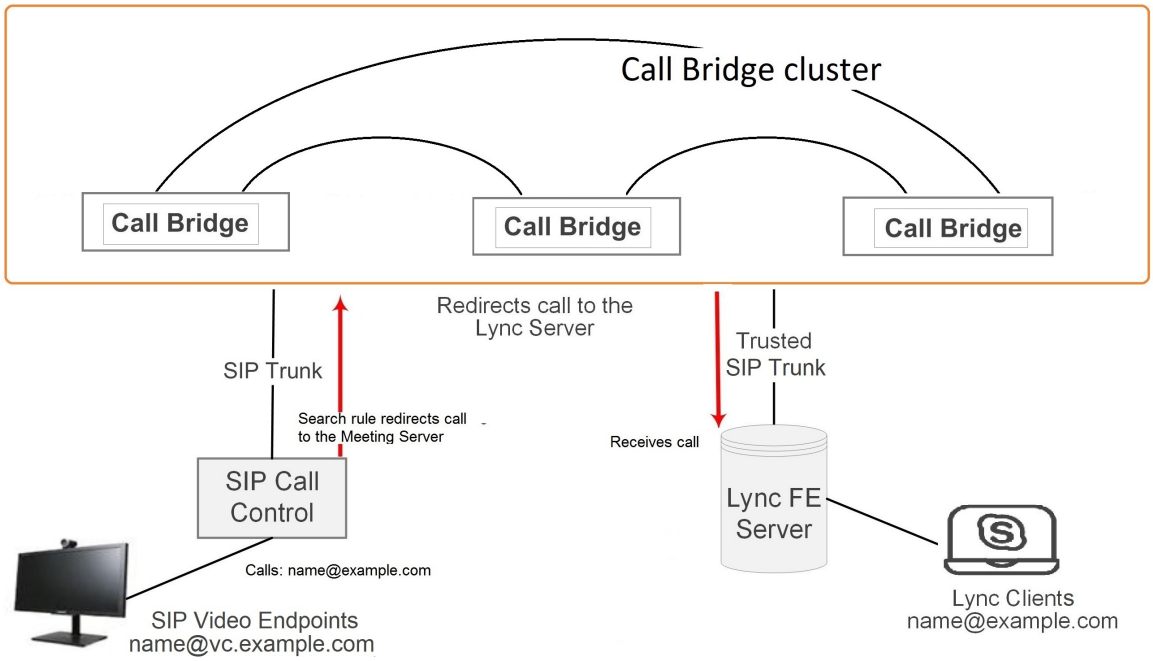


13.3 Adding calls between Lync clients and SIP video endpoints

This section assumes the configuration described for outbound calls and incoming calls has 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 43: 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.

13.3.1 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 13.1.1.

this will route Lync client calls to SIP video endpoints.

13.3.2 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.

13.3.3 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 13.1.2.

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**.

13.4 Integrating Cisco Meeting App with SIP and Lync clients

Note: Cisco Meeting App users are not permitted to call out to Lync meetings.

Refer to the sections on LDAP Configuration and Configuring the XMPP server for instructions on configuring your Meeting Server to use the Cisco Meeting App.

If you are using the same LDAP configuration to create both your Lync accounts and Cisco Meeting Apps, problems may occur if a user tries to call a Lync client when using the Meeting Server as a gateway, because the user may end up calling your Cisco Meeting App XMPP client. The Meeting Server has a table of incoming dial plan rules to prevent this.

For example, assume there is an account fred@example.com on the Meeting Serverand a **`fred@lync.example.com`** account on a Lync FE server. If a call arrives at the Meeting Server and no Call Matching rules are configured, the Meeting Server will ignore the domain and the call will go to the Meeting Server's fred@example.com account. In other words, dialing fred@xxxx will ignore xxxx and see if there is a user "fred" locally.

This is problematic because a user trying to call the Lync address fred@lync.example.com using the Meeting Server as a gateway, will end up in a call with the XMPP client logged in as fred@example.com. If the same LDAP structure has been used to create both the Meeting Server's and the Lync's user accounts, this will be a common problem.

The solution is to configure an **`/inboundDialPlanRule`** with the parameter **`name`** set to something distinct from the domain that the Lync FE server uses. In the example above, a sensible choice would be **`domain=example.com`**. Then, a call to fred@example.com will reach the Cisco Meeting App but a call to fred@lync.example.com or fred@xxxx will not. Instead, if a **`/forwardingDialPlanRule`** is set up, the Meeting Server forwards the call onwards.

13.5 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.

13.5.1 Lync Edge call flow

To establish a call from the Meeting Server to the Lync Edge server (see Figure 44 below):

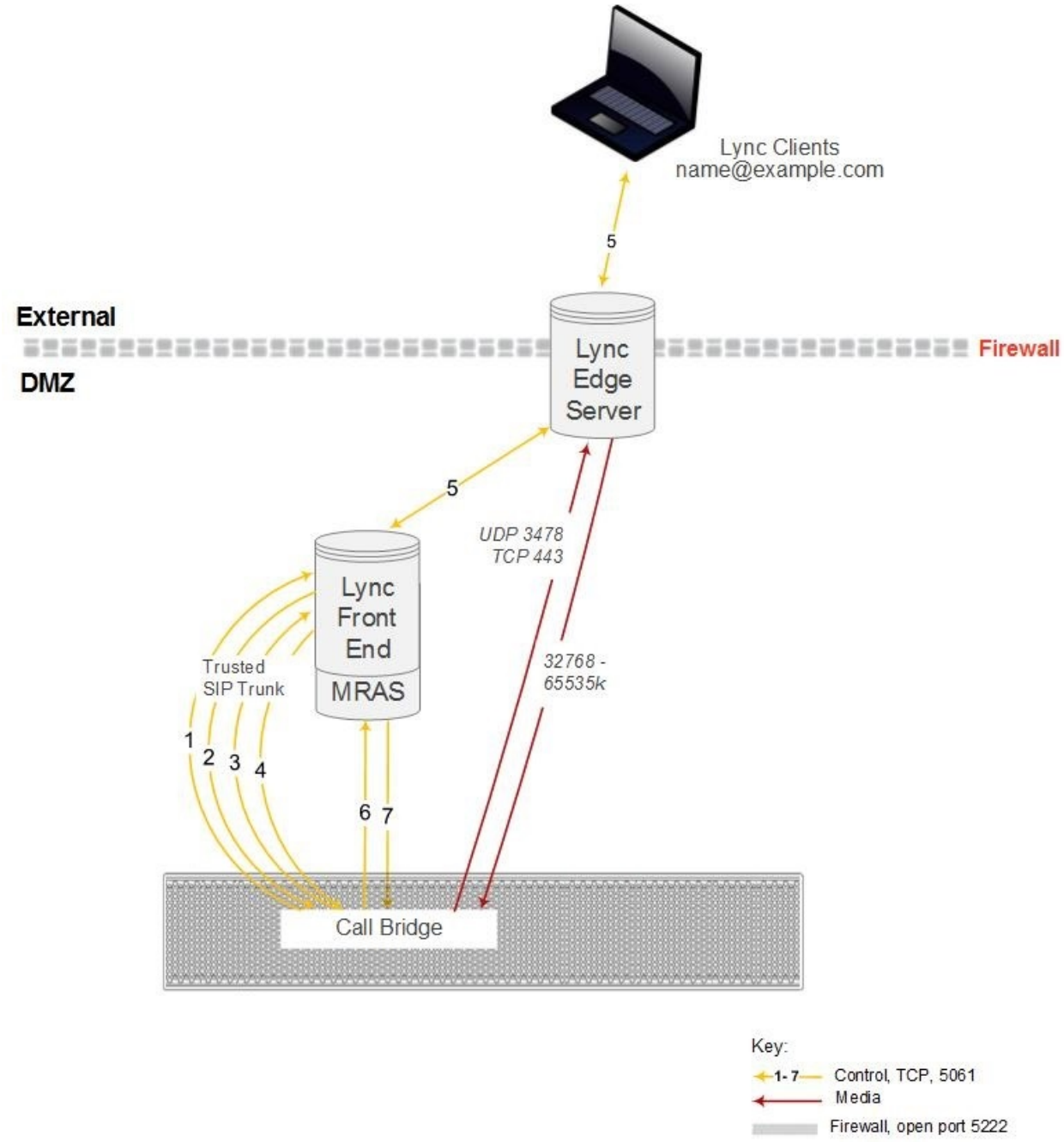
1. The Call Bridge makes a "register" SIP call to the Lync Front End server.
2. The "register" is acknowledged.
3. The Call Bridge sends a "service" to the Lync Front End server.
4. The Front End server returns the URI of the media relay authentication server (MRAS). (The Lync Edge Server acts as a MRAS.)
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 44: Call Bridge to Lync Edge server call flow



13.5.2 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 Server to 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.
- Cisco Meeting Apps continue to use the Meeting Server TURN server even if a Lync Edge server is configured. Note: Cisco Meeting 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.

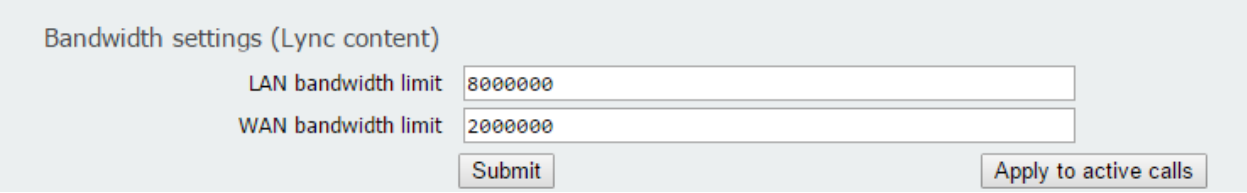
13.6 Controlling the bandwidth for sharing content on Microsoft Lync and Skype for Business calls

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 45: Setting bandwidth for Lync content sharing



13.7 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 B 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.

13.8 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; Cisco Meeting 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 A.

Configure the Lync Front End servers, then follow the task sequence in Table 9 below:

Table 9: 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.

14 Office 365 Dual Homed Experience with OBTP Scheduling

14.1 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.

14.2 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.

14.3 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.

15 SIP and Lync call traversal of local firewalls (BETA)

Note: SIP and Lync call traversal of local firewalls is a beta feature and should not be used in production environments. This edge feature will be removed from the Cisco Meeting Server software in a future version.

The Meeting Server supports traversal of local firewalls for SIP endpoints and Lync calls. The Call Bridge uses the TURN server component within the Meeting Server to traverse the local firewall and sends the SIP signal via a new SIP Edge component. A third party SIP firewall traversal device is not required.

You need to set up an Outbound dial plan rule, even if you don't plan on making outgoing calls via the SIP edge. This is because any new transaction within the call or sending content to a Lync device uses the outbound rule. Once the SIP Edge has been configured and enabled, incoming calls from SIP devices are automatically traversed across the firewall.

Note: this feature assumes that the remote SIP device can see the TURN server. It does not require the remote SIP device to be ICE aware. It also requires the remote SIP device to be able to contact the SIP Edge, the SIP Edge can either have a public IP address or sit behind a NAT with appropriate forwarding of traffic.

Figure 46 shows a schematic for SIP call traversal using the TURN server and SIP Edge component on a single Edge server in the DMZ network which is accessible to remote SIP devices via the public IP address 203.0.113.1. The Call Bridge is deployed in the private network on a Core server and accesses the TURN server and SIP Edge via the internal IP address 198.51.100.1. Figure 47 illustrates SIP call traversal using two Edge servers, one for the SIP signaling and the other for the media stream. The table below 15 lists the ports required to be open for SIP/Lync call traversal.

Figure 46: Firewall traversal for remote SIP devices using a single Edge server

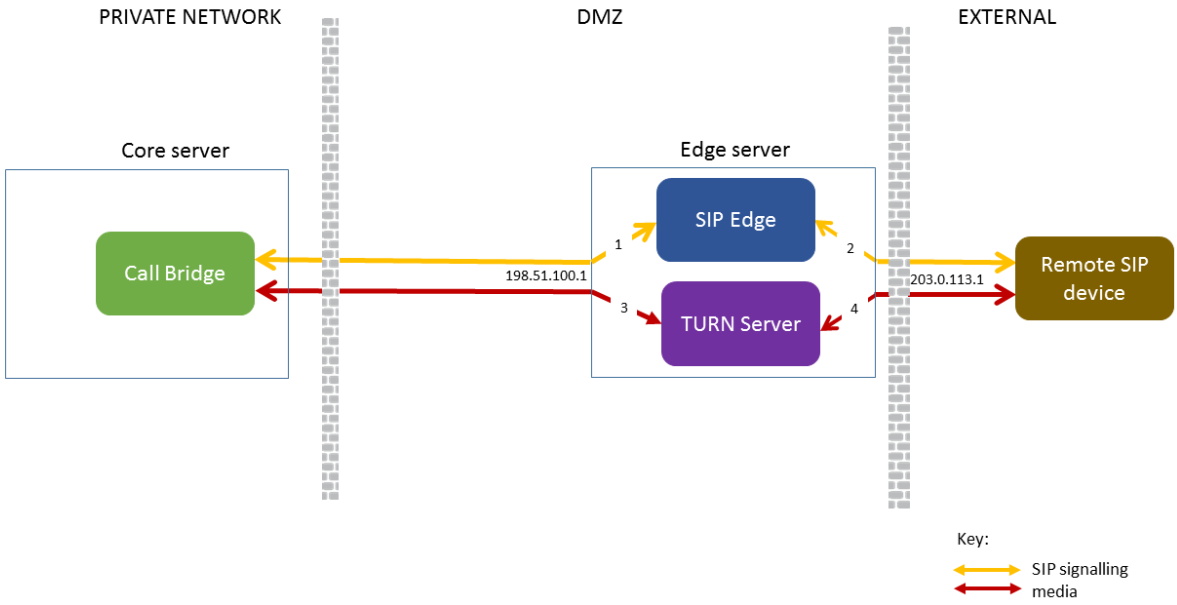
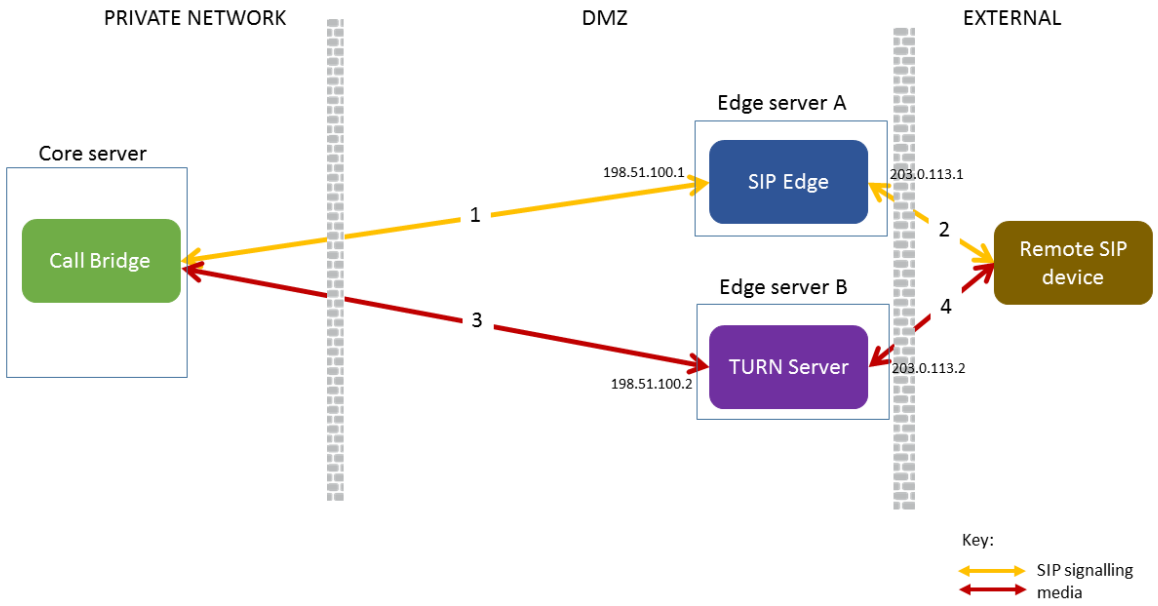


Figure 47: Firewall traversal for remote SIP devices using two Edge servers



Note: SIP endpoints or Lync clients external to the network may still need a third party device to route SIP signaling across their firewall.

Table 10 below lists the ports required to be open for SIP/Lync call traversal.

Table 10: Ports to open for SIP/Lync call traversal

Link	Component	Connecting to	Destination port to open port	Traffic type	Traffic direction with respect to component	Configurable
1	Edge server (private interface)	Call Bridge	Any unused port, e.g. 3061	SIP, TLS	Incoming from Call Bridge	Yes
2	Edge server (public interface)	Remote SIP devices	5061 (Note 1)	SIP TLS (Note 2)	Incoming/Outgoing	Yes
3	Call Bridge	TURN server	32768-65535	Media UDP (STUN, RTP)	Incoming/Outgoing	No
3	Call Bridge	TURN server	32768-65535	Media TCP (RDP)	Incoming/Outgoing	No
3	Call Bridge	TURN server	3478	Media UDP	Incoming	No.
3	Call Bridge	TURN server	3478	Media TCP	Incoming	Yes, see note 4.
4	TURN server	Remote SIP devices	32768-65535	Media UDP (STUN, RTP)	Incoming/Outgoing	No
4	TURN server	Remote SIP devices	32768-65535	Media TCP (RDP)	Incoming/Outgoing	No
3,4	TURN server	Remote SIP devices	3478	UDP	Incoming/Outgoing	Yes
3,4	TURN server	Remote SIP devices	443 (Note 3)	TCP	Incoming/Outgoing	Yes

Note 1: Port 5061 is normally used for SIP TLS

Note 2: Only SIP TLS is supported, there is no support for UDP or TCP on port 5060

Note 3: Using the MMP command **turn tls <port>** will change the TCP port that the TURN server listens on for both Call Bridge and App connections..

15.1 Configuring SIP/Lync call traversal

1. Set up a SIP Edge on the Edge server.

   a. Configure the internal interface and port for communication to the Call Bridge. If the Call Bridge is using port 5061, then use a different port.
   sipedge private <interface>:port

   for example: **sipedge private a:3061**

   b. Configure the external interface and port on the SIP Edge
   sipedge public b:5061

   for example: **sipedge public b:5061**

   Note: The SIP Edge always uses TLS for communication. Typically SIP TLS uses port 5061.

   c. If the SIP Edge is behind a NAT, then configure the public address of the NAT.
   sipedge public-ip <address>

   for example: **sipedge public-ip 203.0.113.0**

   Note: DNS records used for external connections must match the public address

   d. Set up a certificate, key file and trust bundle on the SIP Edge. These files are used to communicate with the internal Call Bridge and the external SIP server. If you have previously assigned a public CA signed certificate to the Call Bridge, then you can use the same certificate on the SIP Edge. If the SIP Edge has direct federation with a Lync Edge server, then the certificate file must be signed by a public CA trusted by the Lync deployment (as was previously required for the Call Bridge).

   Combine the Call Bridge certificate and the chain of CA certs into one file and use this as the SIP Edge certificate <certificatefile>. To enable the SIP Edge to trust the Call Bridge for the TLS trunk, use the Call Bridge certificate as the <trust-bundle>.
   sipedge certs <keyfile> <certificatefile> <trust-bundle>

   for example:

   sipedge certs sipedge.key sipedge.crt callbridge.crt

Note: SIP Edge certificates need to be signed by a public CA and trusted by the third party SIP server. Apply the certificate to the SIP Edge and the Call Bridge. For more information on certificates refer to the Certificate Guidelines.

e. Enable the SIP Edge
`sipedge enable`

2. Set up a trunk from the Call Bridge to the SIP Edge. Note that you need to configure trust between the Call Bridge and the SIP Edge before creating the trunk.

Note: Currently, only one trunk can be made to any SIP Edge, and only one trunk can be made from the same Call Bridge.

a. Set up certificates for the connection to the SIP Edge. This uses the certificate from the SIP Edge as used above.
`callbridge trust edge <certificate file>`
for example: **`callbridge trust edge sipedge.crt`**

b. Create the trunk using the IP address and port of the private interface.
`callbridge add edge <ip address>:<port>`
for example: **`callbridge add edge 198.51.100.0:3061`**

3. Set DNS records to point to SIP Edge(s) for SIP and/or Lync. DNS can point to multiple SIP Edges for resilience. Use the **`_sips._tcp<domain>`** SRV record for the external TLS connection.

4. Configure the TURN server as explained in Chapter 9. Note that TURN TLS is required for Lync content data packets as they use TCP and not UDP.

5. Create the outbound dial plan rule. Use the API to PUT to the relevant outbound dial plan rule '/api/v1/outboundDialPlanRules/ <outbound dial plan rule id>', with 'callRouting=traversal'

Points to note:

- You cannot use the Web Admin interface to select the call routing.
- Outgoing calls use the certificates that you setup for incoming calls, see step 1d above.
- The SIP Edge only supports TLS. All dial plan rules targeting the SIP Edge must be set to 'sipControlEncryption=encrypted'.
- The Call Bridge determines the next hop of the signaling by doing a DNS lookup. It then sends this information to the SIP Edge using the outbound rules.

Table 11 below outlines the call flow to establish an outgoing call from the Meeting Server to a remote SIP device via the SIP Edge server.

Table 11: Call flow from the Meeting Server to Remote SIP device via SIP Edge

See signals in 15 and Figure 4	Call Flow
1	Call Bridge uses an outbound dial plan rule to route SIP signaling via the SIP Edge server
1	Call Bridge sends a DNS request to resolve the next hop to send the request to
1	Call Bridge sends the requests to the SIP Edge with both local address and the TURN address for receiving media
2	The SIP Edge server makes the outgoing call to the remote SIP device
	The remote SIP device answers the call
3,4	Media flows between the TURN server and the remote SIP device

16 Recording meetings

16.1 Overview

There are two methods of recording meetings when using Meeting Server:

- Third-party SIP recorder
- Meeting Server internal recorder component

16.1.1 Third-party SIP recorder support

From 2.9, the Meeting Server allows configuration of a third-party external SIP recorder so that when recording is started an administrator-configured SIP URI is called instead of using the Meeting Server internal 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.

16.1.2 Meeting Server internal 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 48. 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.

The recommended deployment for production usage of the Recorder is to run it on either a dedicated VM, or as part of a combined Recorder/Streamer VM. This VM should be sized with 1 vCPU and 0.5 GB of memory per concurrent 720p30 recording, with a minimum of 4 vCPU and

a maximum of 24 vCPU. If combined with a Streamer, then the sum of both components memory and RAM needs to be allocated, subject to the same minimum and maximum vCPU value.

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.

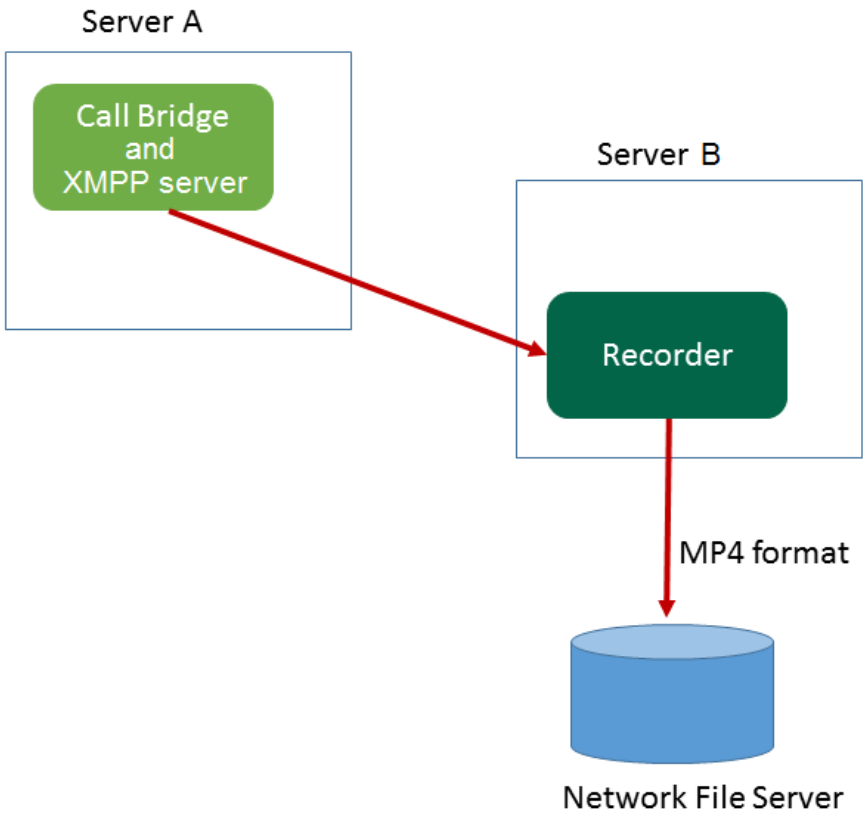
The recorder uses variable bit rate, so it is not possible to accurately predict how much storage a recording will take. Our testing has shown that the size of 720p30 recordings ranges between 300MB to 800MB for 1 hour. In terms of budgeting it would be safe to assume 1GB per hour.

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>.

Figure 48: Permitted deployment for recording: remote mode



The Recorder also supports redundant configurations, see Figure 49 and Figure 50. If you use multiple recorders then the solution load balances recordings between all recording devices and no knowledge of the physical location of recording devices is known. Every Call Bridge will use every Recorder by default. You may also associate specific Recorders to specific call bridges in the API.

Figure 49: Permitted deployments for recording: multiple recorders

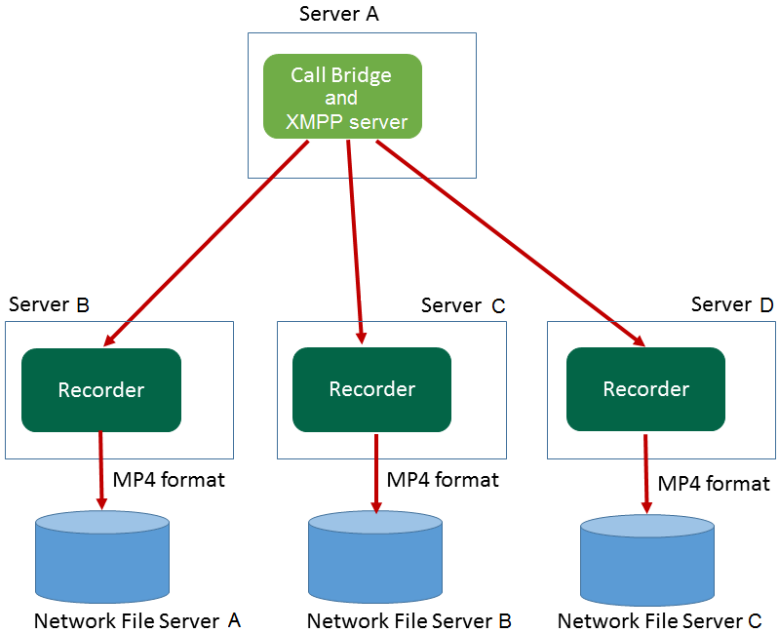
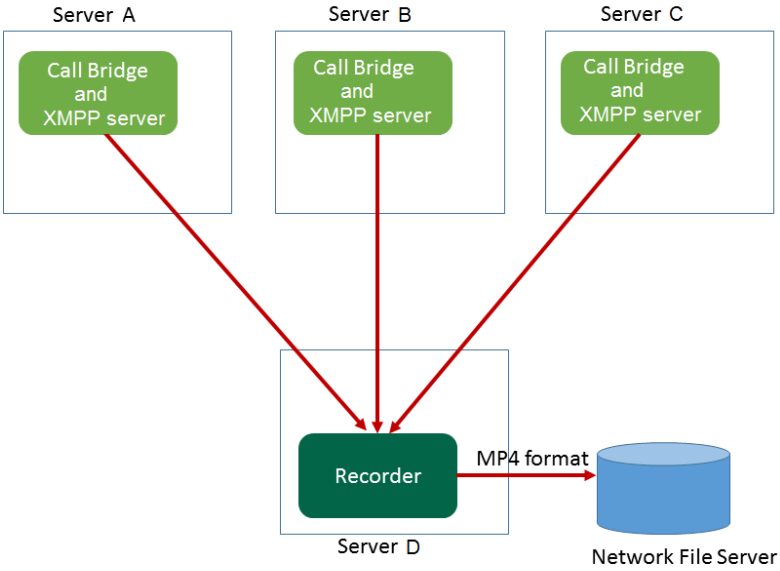


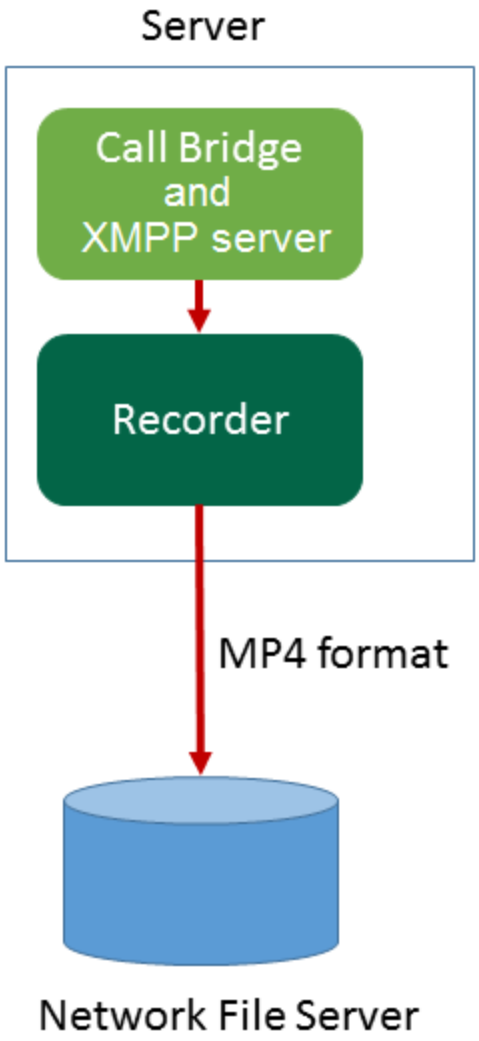
Figure 50: Permitted deployments for recording: Call Bridge cluster



For testing purposes the Recorder can be co-located on the same server as the Call Bridge hosting the conferences. This may support between 1 to 2 simultaneous recordings.

Note: Acano X-series servers used in the single combined deployment mode should only be used for testing the Recorder, they should not be used in production networks to host the Recorder.

Figure 51: Permitted deployment for testing : local mode



16.2 Configuring the Meeting Server Recorder component

Note: This section only applies to the Meeting Server Recorder component.

- Use MMP commands to enable the Recorder on a Meeting Server, specify which Call Bridges within the deployment will work with the Recorder and where to save the recordings. The additional MMP commands are given in the MMP Command Line Reference guide.
- Specify the HTTPS URL address that the Call Bridge will use to reach this recorder. Either POST the URL to the /recorders object or PUT to the /recorders/<recorder id> object.
- 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.
- 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>

Note: For the first 5 seconds after recording is started, the call will not be recorded. This is intentional and is to aid lipsync between video and audio in the recording. If you record for less than 5 seconds a small file will be saved on the NFS, but it will not play back.

16.3 Example of deploying recording using the internal recorder component

Note: This section only applies to the Meeting Server Recorder component.

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.

Note: The Recorder behaves as an XMPP client, so the XMPP server needs to be enabled on the Meeting Server hosting the Call Bridge.

This example gives the steps to deploy a recorder remote to the Call Bridge. It assumes that you already have a working Call Bridge and XMPP server.

1. Create a certificate and private key for the Recorder, following the steps described in the Certificates Guidelines for an internal CA signed certificate.
2. SSH into the MMP of the Meeting Server hosting the Recorder.
3. Configure the Recorder to listen on the interface(s) of your choice with the following command:
   `recorder listen <interface[:port] allowed list>`

   The Recorder can listen on multiple interfaces, e.g. one on public IP and one on the internal network. (However, it cannot listen on more than one port on the same interface.)

   The following is an example where interfaces are set to interface A and B, both using port 8443.
   `recorder listen a:8443 b:8443`

   To use a local Recorder, the Recorder must listen on the loopback interface lo:8443, for example
   `recorder  listen lo:8443 b:8443`
4. Upload the certificate file, key file and certificate bundle to the MMP via SFTP.
   `recorder certs <keyfile> <certificatefile> [<crt-bundle>]`
5. Add the Call Bridge certificate to the Recorder trust store using the command:
   `recorder trust <crt-bundle>`
6. Specify the hostname or IP address of the NFS, and the directory on the NFS to store the recordings
   `recorder nfs <hostname/IP>:<directory>`

   Note: The Recorder does not authenticate to the NFS.

7. Use the recorder command to list the details for the recorder, for example:

```
cms1> recorder
Enabled                  : true
Interface allowed list     : a:8443 b:8443
Key file                 : recorder0.key
Certificate file         : recorder0.cer
CA Bundle file           : recorder.crt
Trust bundle             : callbridge.crt
NFS domain name          : examplecompany_nfs
NFS directory            : /home/examplecompany/nfs
```

8. Enable the Recorder:

```
recorder enable
```

9. Create DNS A record for the Recorder and set it to resolve to the IP Address of the Ethernet interface you want the Recorder to listen on.
10. Use the API of the Meeting Server hosting the Call Bridge to configure the settings through which the Call Bridge will communicate with the Recorder. Using the Web Admin interface of the Meeting Server, select **Configuration**>**API**:
    a. Specify the HTTPS URL address that the Call Bridge will use to reach this recorder.
        i. From the list of API objects, tap the ► after **/recorders**
        ii. Either click on the **object id** of an existing recorder or create a new one
        iii. Set the **url** to reach this recorder

Note: If using a local Recorder, the URL must be the loopback interface, for example `https://127.0.0.1:8443`

---

    b. Select whether a meeting can be recorded or not and whether the recording will start without any user intervention.
        i. From the list of API objects, tap the ► after **/callProfiles**
        ii. Either click on the **object id** of an existing call profile or create a new one
        iii. Set **recordingMode** = **automatic** for a recording to occur without any user intervention, or = **manual** which requires users to manually start and stop recording using DTMF
    c. Control which users have permission to start and stop recording.
        i. From the list of API objects, tap the ► after **/callLegProfiles**
        ii. Either click on the **object id** of an existing call leg profile or create a new one
        iii. Set **recordingControlAllowed** = **true**
    d. Map the DTMF tones for starting and stopping recording.

i. From the list of API objects, tap the ► after **/dtmfProfiles**

ii. Either click on the **object id** of an existing call leg profile or create a new one

iii. Set **startRecording** and **stopRecording** to the DTMF sequence to be used by a participant to start and stop a recording. For example, setting **startRecording** = ****7** to start and **stopRecording** = ****8** to stop the recording.

11. Remember to set the permissions on your NFS to rw and change the chown and chmod permissions on the directory. For example:

```
sudo chown nobody:nogroup /record
sudo chmod -R 777 /record
```

16.3.1 Setting the resolution of the Recorder

Note: This section only applies to the Meeting Server Recorder component.

From version 2.4, you can configure the recording resolution of the Meeting Server Recorder. The resolution is configured on the Recorder component itself and is not passed to the Recorder by the Call Bridge. To configure the resolution use the MMP command: **`recorder resolution <audio|720p|1080p>`**.

If no resolution is configured, the default setting is 720p30. Audio recordings are stored in .mp4 file format.

Table 12 provides typical specifications for the different recorder settings, the recommendations are based on our internal testing.

Table 12: Recorder resolution specifications

Recorder setting	Percentage of physical core per recording	RAM required per recording	Typical disk usage per hour	Planned disk usage per hour (recommended)
1080p	100%	1GB	1GB to 1.6GB	2GB
720p	50%	0.5GB	300MB to 800MB	1GB
audio	20%	125MB	70MB	100MB

16.3.2 Example of setting the recording resolution

This example assumes you already have a working Recorder.

1. SSH into the MMP of the Meeting Server hosting the Recorder.
2. Disable the Recorder to change the configuration.
   ```
   recorder disable
   ```

3. Configure the Recorder to record meetings at the specified resolution using the MMP command:
   ```
   recorder resolution <audio|720p|1080p>
   ```
   For example:
   ```
   recorder resolution 1080p
   ```
4. Re-enable the Recorder so that it picks up the new configuration.
   ```
   recorder enable
   ```

16.4 Configuring a 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.
  - Use the Web Admin interface of the Meeting Server, select **Configuration**>**API**
  - From the list of API objects, tap the ► after **/callProfiles**
  - Either click on the **object id** of an existing call profile or create a new one
  - 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.

16.5 Finding out recording status

Note: This section applies to both the Meeting Server internal Recorder component and an external third-party SIP recorder.

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`**).

16.6 Recorder licensing

Note: This section applies to both the Meeting Server internal Recorder component and an external third-party SIP recorder.

You will need a license for each Call Bridge, and one or more licenses for recording which is loaded on the Call Bridge server, not the Recorder server. A recording license supports 1 concurrent recording. Contact your Cisco sales representative to discuss your licensing requirements.

16.7 Recording indicator for dual homed conferences

Note: This section applies to both the Meeting Server internal Recorder component and an external third-party SIP recorder.

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 13 below shows the icons that Meeting Server will display to indicate that a dual homed conference is being recorded.

Table 13: 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: Cisco Meeting 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 Cisco Meeting App video pane.

16.8 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.

16.8.1 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 14: Port Requirements

Component	Connecting to	Destination port to open
Call Bridge	NFS (version 3)	2049

Component	Connecting to	Destination port to open
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

16.8.2 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.

17.1 Streaming meetings

The Streamer component adds the capability of streaming meetings held in a space to the URI configured on the space.

An external streaming server needs to be configured to be listening on this URI. 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. However, we have only tested against Vbrick as an external streaming server.

The Streamer connects to an external server using RTMP with an overall bitrate of 2Mbps. The video is encoded using H.264 at 720p30, while the audio is 64kbps AAC-LC. All traffic between the Streamer and the external streaming server is unencrypted.

The Streamer should be hosted on another Meeting Server instance than the server hosting the Call Bridge, see Figure 52. If the Streamer is hosted on the same server as the Call Bridge (local), then it should only be used for testing purposes.

The recommended deployment for production usage of the Streamer is to run it on a separate VM. This VM should be sized with 1 vCPU and 1GB of memory per 6 concurrent streams, with a minimum of 4 vCPUs and a maximum of 32vCPUs.

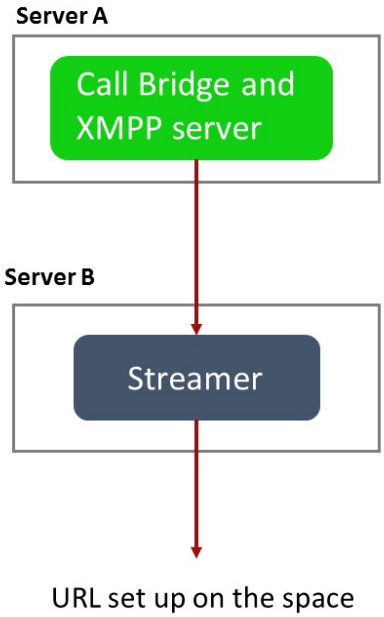
Note: These VM specifications are currently being evaluated, and the sizes are likely to be reduced.

For more details on VM specification see Unified Communications in a Virtualized Environment – Cisco (https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/cisco-collaboration-virtualization.html) and the Cisco white paper on Load Balancing Calls Across Cisco Meeting Server.

Where possible, it is recommended that the Streamer is deployed in the same physical locality as the Call Bridge to ensure low latency and high network bandwidth. If there are network connection issues between the Call Bridge and the Streamer, then the resultant stream could be affected.

Note: you may need to open firewall ports if the streaming destination URIs are on the external side of a firewall.

Figure 52: Permitted deployment for streaming: remote mode



The Streamer also supports redundant configurations, see Figure 53, Figure 54, Figure 55 and Figure 56. If you use multiple streamers then the solution load balances between available streaming devices. To restrict the use of specific Streamers to specific Call Bridges use the Call Bridge Group functionality.

Figure 53: Permitted deployments for streaming: multiple streamers

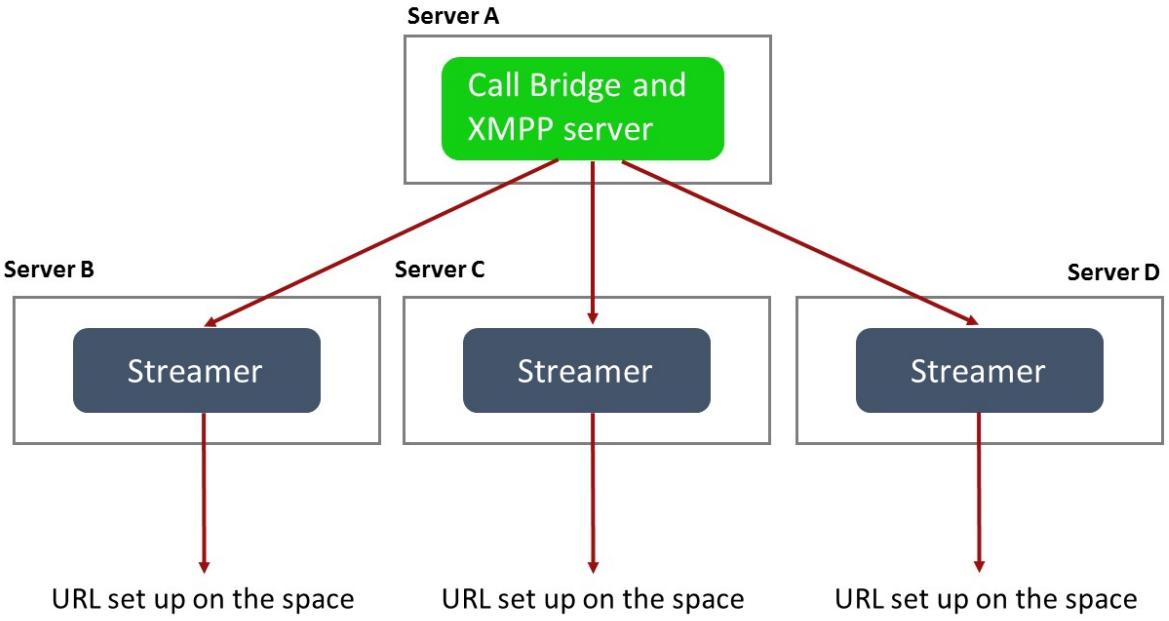
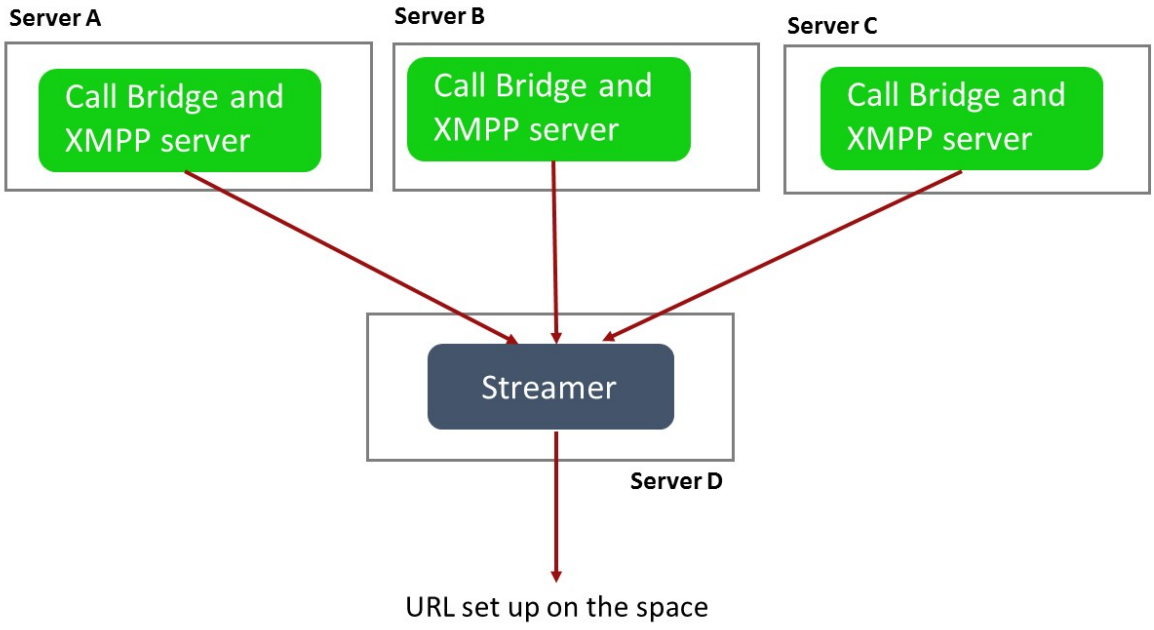


Figure 54: Permitted deployments for streaming: Call Bridge cluster



If your deployment has multiple Call Bridge and multiple Streamers then every Call Bridge will use every Streamer (see Figure 55), unless the **`callBridgeGroup`** and **`callBridge`** parameters have been set for each Streamer on API object **`/streamers`**.

Figure 55: Permitted deployments for streaming: Call Bridge cluster with multiple Streamers and no Call Bridge Groups set up

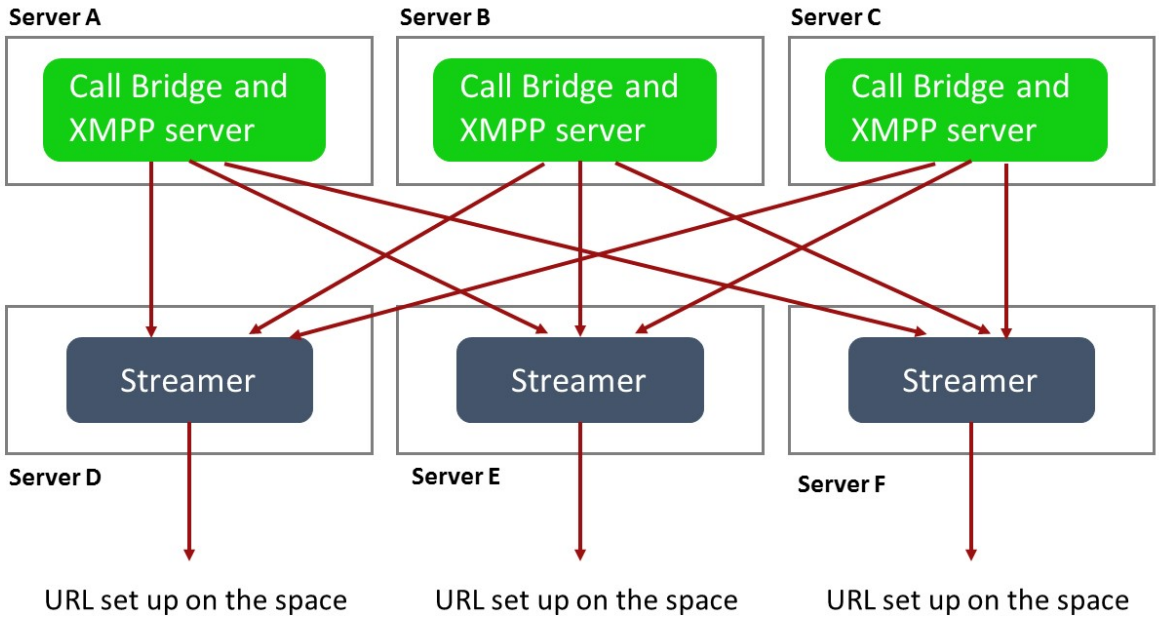
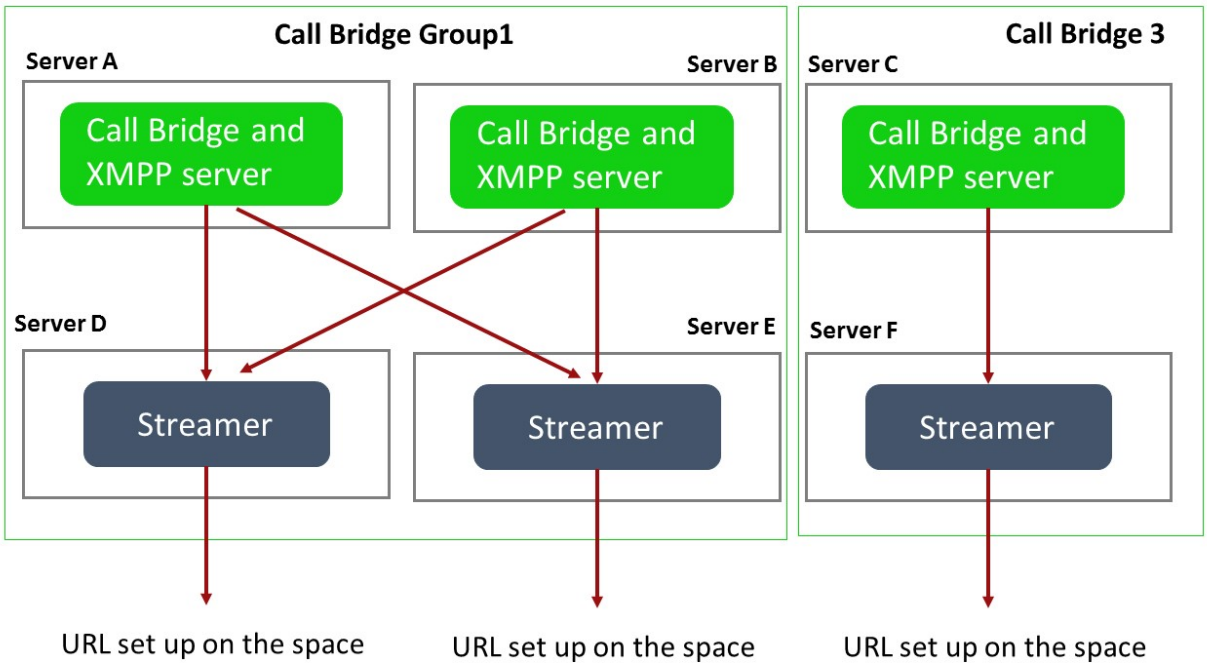


Figure 56: Permitted deployments for streaming: Call Bridge cluster with multiple Streamers and a Call Bridge Group and Call Bridge set up



For testing purposes the Streamer can be co-located on the same server as the Call Bridge. This may support between 1 to 2 simultaneous streamings.

Note: Acano X series servers used in the single combined deployment mode should only be used for testing the Streamer, they should not be used in production networks to host the Streamer.

17.2 Overview of steps to configuring the Streamer

Use MMP commands to enable the Streamer on a Meeting Server, then use the API to specify which Call Bridges within the deployment will work with the Streamer and the url to stream to.

- Specify the HTTPS url address that the Call Bridge will use to reach this streamer. Set the **`url`** parameter for the streamer using the **`/streamers`** object.
- Create a new call profile or update an existing one to control whether meetings are streamed. This call profile is then applied to spaces to allow or prevent the streaming of meetings. Set the **`streamingMode`** parameter on **`/callProfiles`** to:

  `automatic`- streaming occurs without any user intervention. The meeting will still occur even if streaming cannot take place.

manual - participants in the meeting can manually start and stop the streaming using DTMF.

disabled - meetings in spaces with this **/callProfile** applied, will not stream.

- Specify the destination URL if streaming is initiated. Set the **streamUrl** parameter on each space (**/coSpaces**) that allows streaming.
- Select the DTMF tones for the **startStreaming** and **stopStreaming** parameters for **/dtmfProfiles**.
- Decide whether call legs associated with a call leg profile are permitted to start and stop streaming. Set the **streamingControlAllowed** parameter on **/callLegProfiles** to **true** or **false**.

17.3 Example of deploying streaming

Note: The Streamer behaves as an XMPP client, so the XMPP server needs to be enabled on the Meeting Server hosting the Call Bridge.

This example gives the steps to deploy a streamer remote to the Call Bridge. It assumes that you already have a working Call Bridge and XMPP server.

1. Create a certificate and private key for the Streamer, following the steps described in the Certificates Guidelines for an internal CA signed certificate.
2. SSH into the MMP of the Meeting Server hosting the Streamer.
3. Configure the Streamer to listen on the interface(s) of your choice with the following command:
   streamer listen <interface[:port] allowed list>

   The Streamer can listen on multiple interfaces, e.g. one on public IP and one on the internal network. (However, it cannot listen on more than one port on the same interface.)

   The following is an example where interfaces are set to interface A and B, both using port 8445.
   streamer listen a:8445 b:8445

   To use a local Streamer, the Streamer must listen on the loopback interface lo:8445, for example
   streamer listen lo:8445 b:8445
4. Upload the certificate file, key file and certificate bundle to the MMP via SFTP.
   streamer certs <keyfile> <certificatefile> [<crt-bundle>]
5. Add the Call Bridge certificate to the Streamer trust store using the command:
   streamer trust <crt-bundle>

6. Use the streamer command to list the details for the streamer, for example:

```
cms1> streamer
Enabled                  : true
Interface allowed list     : a:8445 b:8445
Key file                 : streamer0.key
Certificate file         : streamer0.cer
CA Bundle file           : streamer.crt
Trust bundle             : callbridge.crt
```

7. Enable the Streamer:

```
streamer enable
```

8. Create DNS A record for the Streamer and set it to resolve to the IP Address of the Ethernet interface you want the Streamer to listen on.
9. Use the API of the Meeting Server hosting the Call Bridge to configure the settings through which the Call Bridge will communicate with the Streamer. Use the Web Admin interface of the Meeting Server, and select **Configuration**>**API**:

   a. Specify the HTTPS url address that the Call Bridge will use to reach this streamer. Tap the ► after **/streamers** and either click on the **Create new** button, or select the **object id** of an existing streamer, and set the **url** parameter for the streamer.

   Note: If using a local Streamer, the URL must be the loopback interface, for example `https://127.0.0.1:8445`

   ---

   b. Set the the **streamUrl** parameter on each space (**/coSpaces**) that allows streaming, **streamUrl** is the destination URL if streaming is initiated.

   Note: some streaming services require username and password, others provide a unique stream key. For example, for vBrick:

   ```
   streamUrl=rtmp://<username>:<password>@<vbrick
   IP/FQDN>/live/PullStream1
   ```

   and for YouTube:

   ```
   streamUrl=rtmp://a.rtmp.youtube.com/live2/<stream key>
   ```

   ---

   c. Create a new call profile that allows manual streaming for meeting in spaces. Tap the ► after **/callProfiles**, click on the **Create new** button. Set the **streamingMode** parameter to **manual**, to require a participant in the meeting to start and stop the streaming using DTMF.

   d. Map the DTMF tones for starting and stopping streaming to **7 and **8 respectively. Tap the ► after **/dtmfProfiles**, click the **Create new** button or select the **object id** of an existing dtmf profile. Set the **startStreaming** parameter to ****7** and the **stopStreaming** parameter to ****8**. Click **Save** or **Modify**.

e. Apply the call profile created above (step c), to those spaces that will allow participants to manually start and stop the streaming of meetings.

17.4 Streamer licensing

You will need one or more licenses for streaming which is loaded on the Meeting Server hosting the Call Bridge, not the server hosting the Streamer. 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.

18 LDAP configuration

If you plan for users to utilize the Cisco Meeting 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 Cisco Meeting 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.

18.1 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'

18.2 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 18.4.

     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 207.

Note: Each imported user must have a unique XMPP 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.

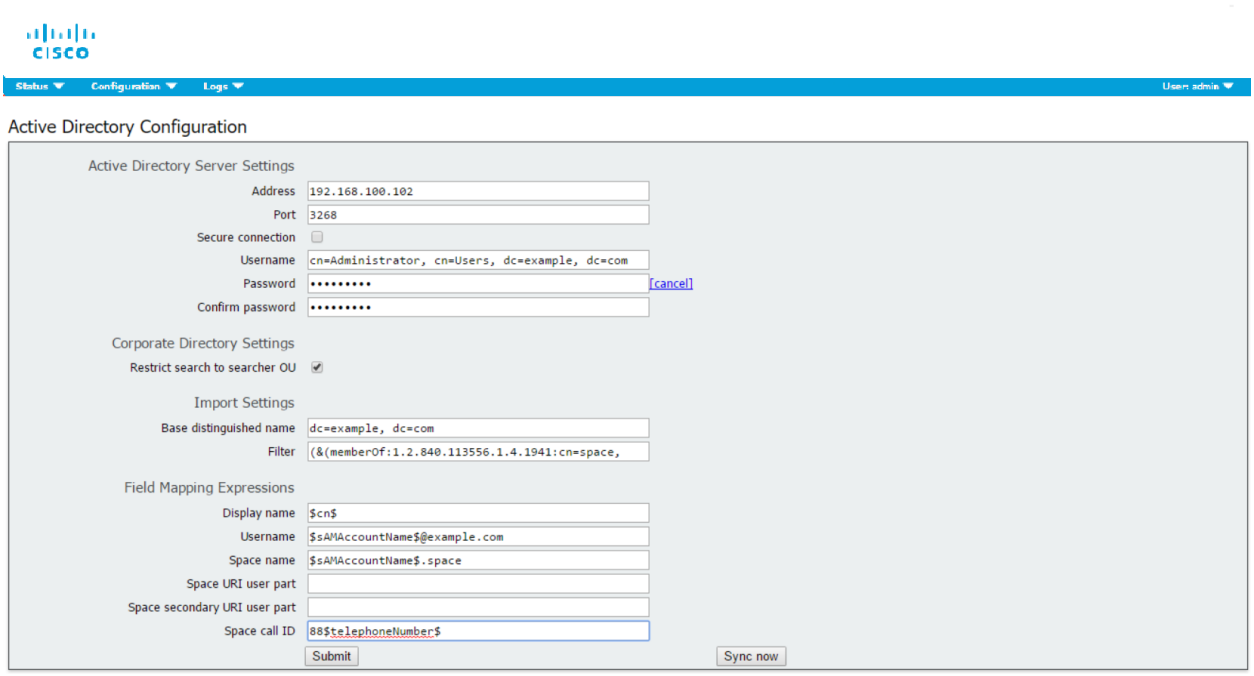
18.3 Example

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	XMPP id
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`

18.4 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@xmpp.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:                    $mail|'/@test/@xmpp/'$
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

18.5 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.

19 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.

19.1 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.

19.2 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.

19.3 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 19.4 . 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.)

19.4 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**.

19.5 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.

19.6 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**.

19.7 iX troubleshooting

Table 15: 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

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 16 below to configure CAC for your deployment.

Table 16: 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 17: 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 18: 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.

21 Diagnostic tools to help Cisco Support troubleshoot issues

21.1 Log bundle

The Meeting Server can produce a log bundle containing the configuration and state of various components in the Meeting Server. This log bundle will aid Cisco Support speed up their analysis of your issue. It will include some of the following files:

- syslog
- live.json
- dumps
- db

If you need to contact Cisco support with an issue, follow these steps to download the log bundle from the Meeting Server.

1. Connect your SFTP client to the IP address of the MMP.
2. Log in using the credentials of an MMP admin user.
3. Copy the file logbundle.tar.gz to a local folder.
4. 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.
5. 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.

21.2 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

21.3 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)
```

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 19: DNS records required for deployment

Type	Example, Description and Resilience Considerations
SRV(*)	**`_xmpp-client._tcp.example.com`** Resolves to: The A record **`xmpp.example.com`** below. Usually this is port 5222. Description: Used by clients to login. The SRV record must correspond to the domain used in your XMPP usernames. Resilience considerations: One SRV record can be created for each XMPP server/Load Balancer such that multiple results are returned in response to a DNS lookup. Clients choose a destination for XMPP traffic based on the priority and weight information.
SRV(*)	**`_xmpp-server._tcp.example.com`** Resolves to: The A record **`xmpp.example.com`** below. Usually this is port 5269. Description: Used to federate between XMPP servers. The SRV record must correspond to the domain used in your XMPP usernames. Resilience considerations: One SRV record can be created for each XMPP server/Load Balancer such that multiple results are returned in response to a DNS lookup. Clients choose a destination for XMPP traffic based on the priority and weight information.

Type	Example, Description and Resilience Considerations
A	**xmpp.example.com** Resolves to: IP address of either the XMPP server or a Load Balancer which is configured to trunk to the XMPP server. Description: Used by clients to login. Resilience considerations: One A record per XMPP server or Load Balancer.
SRV(*)	**_xmpp-component._tcp.<domainNameofXMPPserver>** Resolves to: The A record of an XMPP server in the cluster. Usually this is port 5223. Description: Required if you plan to use DNS to connect between Call Bridges and XMPP servers. Resilience considerations: One SRV record for each XMPP server in the cluster.
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 load balancing on DNS, then a single DNS entry based on IP or location, or 3) a single IP of a 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.

Type	Example, Description and Resilience Considerations
A / AAAA	**ukedges.example.com** **nyedges.example.com** Resolves to: IP addresses of any local Load Balancers. Description: Used in split deployments only by the Core server to create a trunk to a Load Balancer running on the Edge server. Resilience considerations: Each Core server in a given datacenter should trunk to only the Edge servers within that datacenter. In our example, ukedges.example.com would return the IP address of all Load Balancers within the UK datacenter.
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.

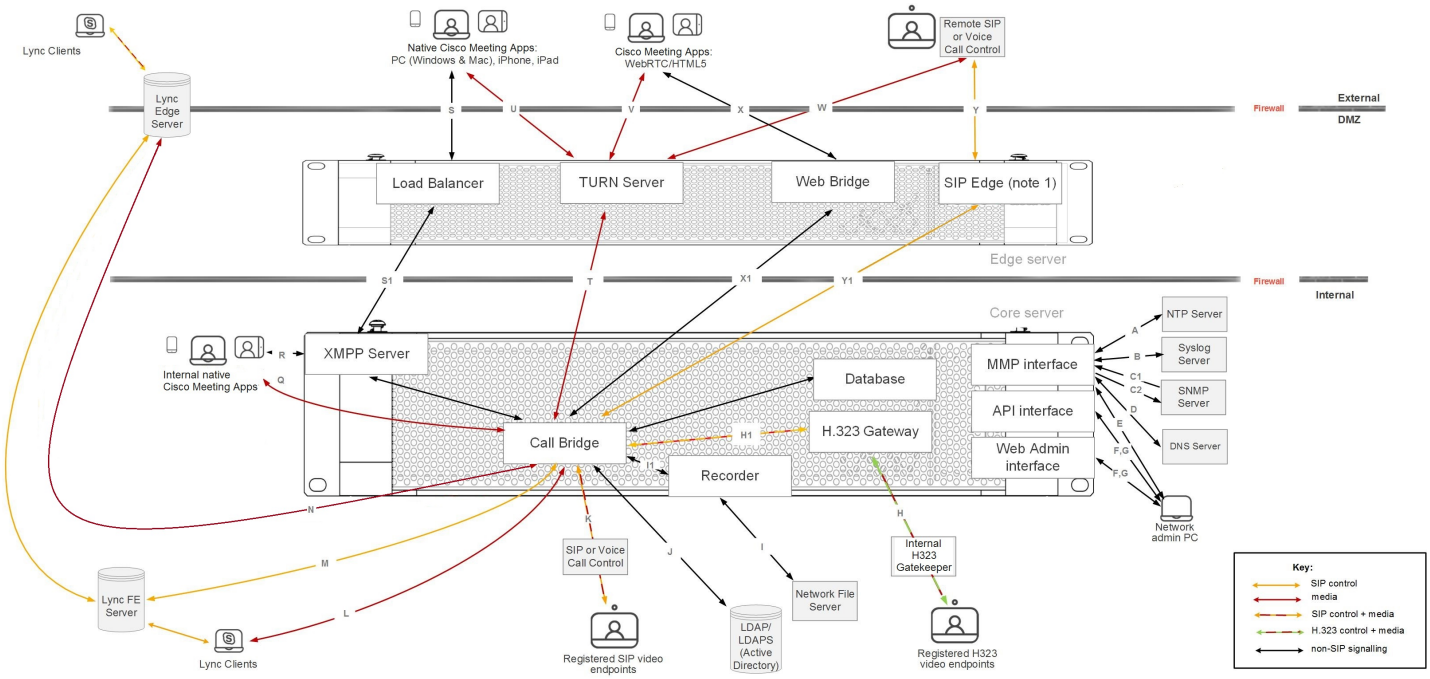
Type	Example, Description and Resilience Considerations
A / AAAA	**fe.<yourLyncdomain>** Resolves to: IP address of the Lync FE server. Description: You will need one record for each individual FE server.
SRV(*)	**_sipfederationtls._tcp.<yourSIPdomain>** Resolves to: The FQDN of the Meeting Server. Description: This record is required for Lync federation.
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 57: Ports that must be open in a scalable and resilient server deployment



Note:
1) The figure above shows the XMPP server listening on an external port. If you prefer the XMPP server to not listen on one of the interface ports (A-D), then instead configure the Load Balancer to listen on the external port and have the Load Balancer relaying the information to the XMPP server.
2) The SIP Edge component is a beta feature

B.1 Configuring the Meeting Server

Table 20 lists the ports to use to configure the Meeting Server.

Table 20: 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 configurable through MMP
G	API or Web Admin	443	HTTPS	TCP	Incoming	Port configurable through MMP

B.2 Connecting services

Use Table 21 to identify which ports are used to connect different services to the Cisco Meeting App .

Table 21: 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	

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
D	MMP/Call Bridge/Web Bridge	DNS server	53	TCP or UDP	Outgoing	On X series servers both the Admin port and the interface being used (A to D) need to be able to access the DNS server on port 53.
	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 22 to identify which ports are used to connect to the components in the Meeting Server.

Table 22: 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
H1	Call Bridge	H.323 Gateway	6061	TCP (SIP)	Outgoing	Port configurable through MMP
H	H.323 Gateway	H.323 Gatekeeper	1720	TCP (H.225)	Incoming	Port not configurable
			port on H.323 Gatekeeper for next hop	TCP (H.225)	Outgoing	
H	H.323 Gateway	H.323 Gatekeeper	1024-65535 (note 1)	TCP (H.245)	Incoming	Port not configurable
			port on H.323 Gatekeeper for next hop	TCP (H.245)	Outgoing	

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
H	H.323 Gateway	H.323 Gatekeeper	32768-65535 (note 2)	UDP media	Incoming and outgoing	
I1	Call Bridge	Recorder	8443		Outgoing	Port configurable through MMP. For a local recorder use the loopback interface, eg lo:8443
I	Recorder	Network File Server (NFS)				Use the MMP command **`recorder nfs <hostname/IP<directory>`** to specify where to store the recordings on the NFS
J	Call Bridge	LDAP/LDAPS (Active Directory)	389/636 (note 3)	TCP/TCP (SIP TLS)	Outgoing	Port configurable through Web Admin interface
K	Call Bridge	Internal registered SIP endpoint or voice call control	5060	SIP UDP	Incoming and outgoing	
K	Call Bridge	Internal registered SIP endpoint or voice call control	5060	TCP (SIP)	Incoming and outgoing	
K	Call Bridge	Internal registered SIP endpoint or voice call control	5061	TCP (SIP TLS)	Incoming and outgoing	
K	Call Bridge	Internal registered SIP endpoint or voice call control	32768-65535	UDP (STUN RTP, BFCP)	Incoming	
L	Call Bridge	Lync client, AVMCU	32768-65535	UDP (STUN RTP)	Incoming	

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
L	Call Bridge	Lync client, AVMCU	1024-65535 (note 1)	UDP (STUN RTP)	Outgoing	
L	Call Bridge	Lync client, AVMCU	32768-65535	TCP (RDP)	Incoming	
L	Call Bridge	Lync client, AVMCU	1024-65535 (note 7)	TCP (RDP)	Outgoing	
M	Call Bridge	Lync FE server	5061	TCP (SIP TLS)	Incoming and outgoing	
N	Call Bridge	Lync edge server	3478	UDP	Outgoing	
N	Call Bridge	Lync edge server	443	TCP	Outgoing	
N	Call Bridge	Lync edge server	32768-65535 (note 2)	UDP (STUN RTP)	Incoming	
	Call Bridge	XMPP server				Internal to Meeting Server, does not require open ports
Q	Call Bridge	Internal native Cisco Meeting Apps	32768-65535	UDP (STUN RTP)	Incoming	
Q	Call Bridge	Internal native Cisco Meeting Apps	1024-65535 (note 1)	UDP (STUN RTP)	Outgoing	
R	XMPP server	Internal native Cisco Meeting Apps	5222	TCP	Incoming	
S	Load Balancer	External nativeCisco Meeting Apps	5222	TCP	Incoming	

Code	Component	Connecting to	Destination port to open	Traffic type	Traffic direction with respect to component	Additional information
S1	XMPP server	Load Balancer	4999	TCP	Outgoing	
T, U,V,-W	TURN server	Call Bridge and remote devices	32768-65535 (notes 2 and 4)	Media UDP (STUN RTP)	Incoming and outgoing	
T, U,V,-W	TURN server	Call Bridge and remote devices	32768-65535 (notes 2 and 4)	Media TCP (STUN RTP)	Incoming and outgoing	
T, U,V,-W	TURN server	Call Bridge and remote devices	3478 (note 4)	UDP (STUN)	Incoming	
T, U,V,-W	TURN server	Call Bridge and remote devices	3478 (note 4)	TCP (STUN)	Incoming	
T, U,V,-W	TURN server	Call Bridge and remote devices	443 (notes 4, 5, 6)	UDP (STUN)	Incoming	
T, U,V,-W	TURN server	Call Bridge and remote devices	443 (notes 4, 5, 6)	TCP (STUN)	Incoming	
X	Web Bridge	WebRTC clients	80	TCP (HTTP)	Incoming	
X	Web Bridge	WebRTC clients	443 (notes 6 and 8)	TCP (HTTPS)	Incoming	
X1	Call Bridge	Web Bridge	443	TCP	Outgoing	
Y	SIP Edge server (public interface)	Remote SIP devices	5061	TCP (SIP TLS) (note 9)	Incoming and outgoing	Port configurable through MMP
Y1	SIP Edge server (private interface)	Call Bridge	any unused port e.g. 3061	TCP (SIP TLS)	Incoming	Port configurable through MMP

Code	Com-ponent	Connecting to	Destination port to open	Traffic type	Traffic dir-ection with respect to com-ponent	Additional information
	Call Bridge	Database				Internal to Meeting Server, does not require open ports

Note:
1) Exact range depends on far end.
2) Although the range is shown as 32768-65535, currently only 50000-51000 is used. A wider range is likely to be required in future releases.
3) 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.
4) If the media ports (32768-65535) are not open then TCP/UDP port 3478/443, used to connect to the TURN server, will be used to relay media.
5) UDP/TCP port 443 can be changed. Using the MMP command turn tls <port> will change the second UDP/TCP port that the TURN server listens on.
6) The TURN server will not listen on port 443 on the loopback interface. This is to avoid port clashes with other services that may be running on port 443 on the loopback interface.
7) Exact range depends on configuration of Lync server.
8) To run both the TURN server and the Web Bridge on port 443 requires the two components to be run on different interface: port combination, if this is not possible then use port 447 for the TURN server.
9) Port 5061 only supports SIP TLS, there is no support for UDP or TCP.
10) Lync clients includes AVMCU–the same ports need opening.

B.4 Additional ports required for Scalability and Resilience

Figure 58: Additional ports required to be open in a scalable and resilient multiple server deployment

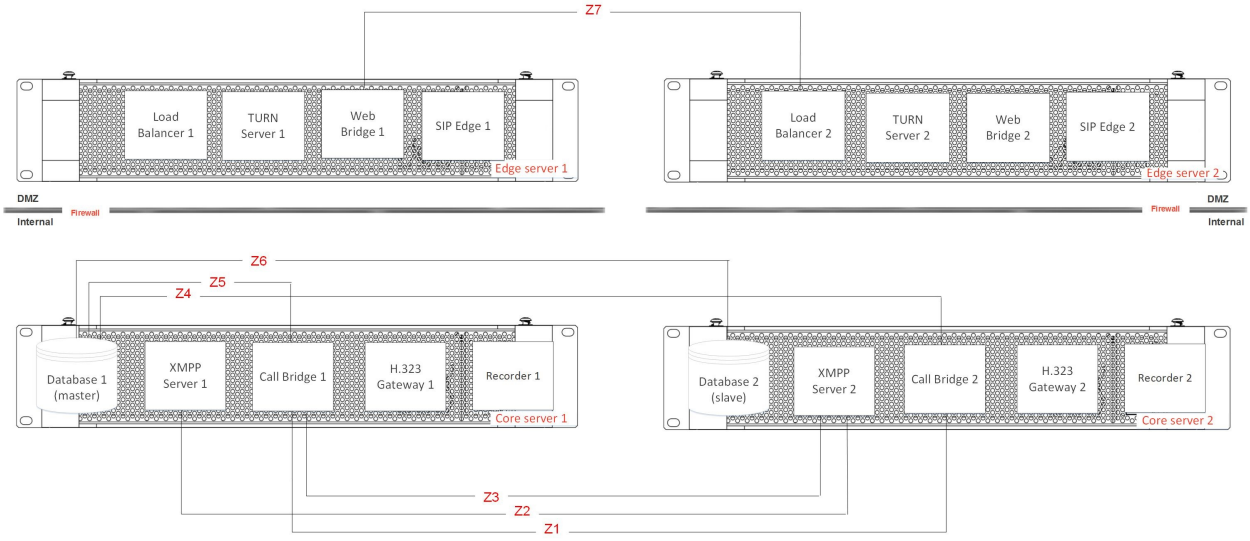


Table 23: Additional ports required to be open in a scalable and resilient multiple server deployment

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 (SIP)	Incoming and outgoing	Figure 58 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	5060	UDP (SIP)	Incoming and outgoing	
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	XMPP server 1	XMPP server 2	5222	TCP	Incoming and outgoing	
Z3	Call Bridge1	XMPP server 2 (note 2)	5223	TCP	Outgoing	Configurable through Web Admin Interface
Z4	Call Bridge1	Database 1 (primary)	n/a	TCP		
Z5	Call Bridge2	Database 1 (primary)	5432	TCP	Incoming and outgoing	
Z6	Database 1 (primary)	Database 2 (replica)	5432	TCP	Incoming and outgoing	Database cluster / replication ports
Z7	Web Bridge1	Load Balancer 2	5222	TCP	Outgoing	

Note:
1) Exact range depends on far end
2)This port is not required if the Call Bridge and XMPP server are on the same Core server but for deployments with multiple Core servers, each Call Bridge must be able to reach all of the XMPP servers for resilience

B.5 Ports open on loopback

The ports listed in Table 24 are open on the loopback interface.

Table 24: 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	
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

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, ACU, 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, ACU, 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.

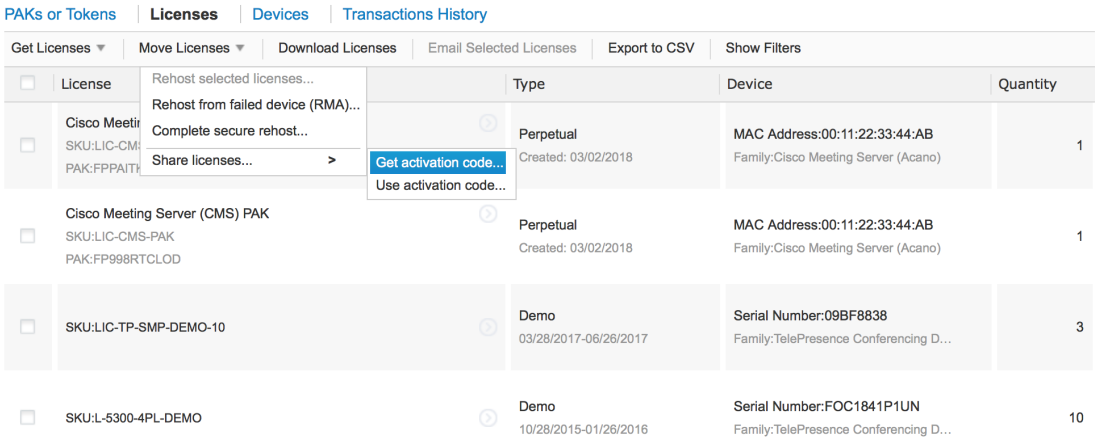
   Note: You can open the **cms.lic** in a text editor to check on the features purchased.

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

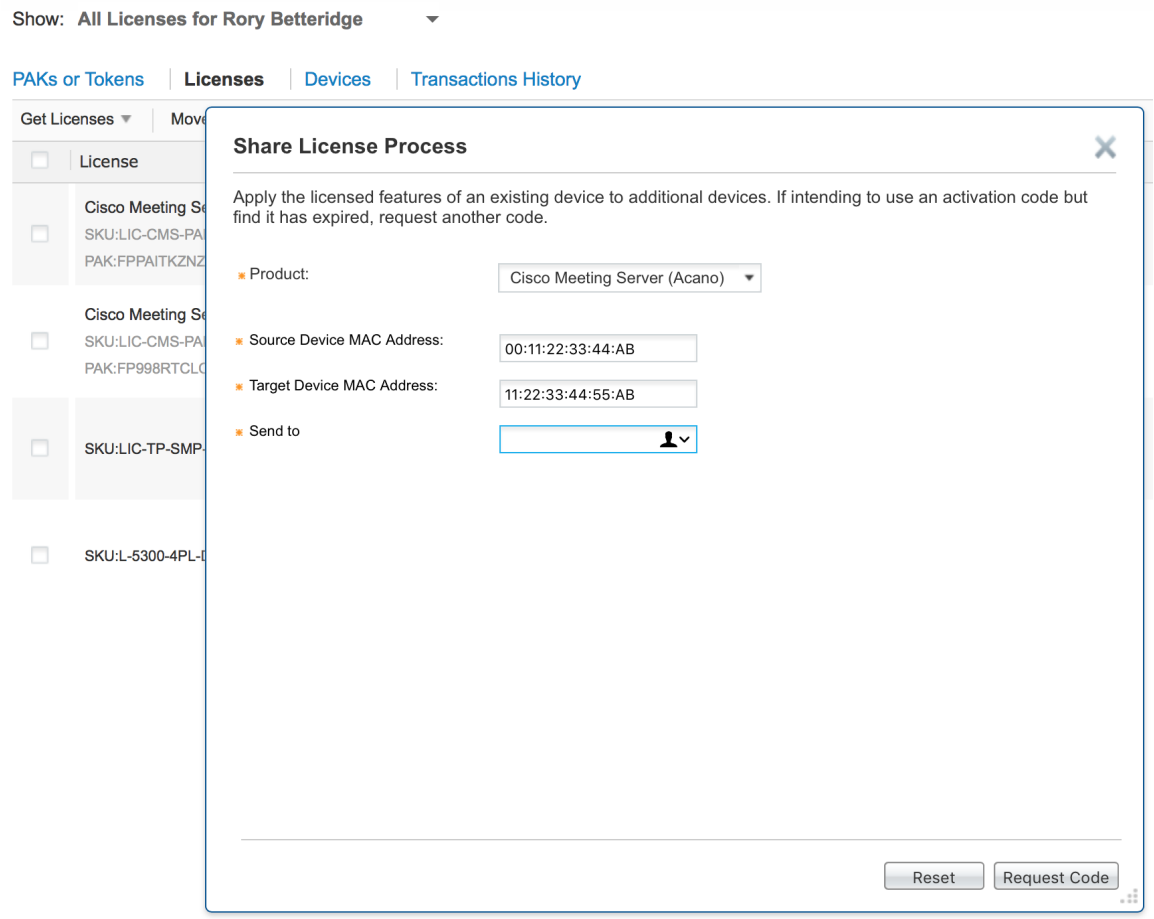
4. You now need to share the feature licenses registered to the primary node with the other secondary nodes. For each Meeting Server node that you registered in step 3, complete steps a to k below.

a. In the Licence Registration Portal, select the **Licenses** tab, and navigate to **Move Licenses** > **Share Licenses** > **Get Activation Code**.

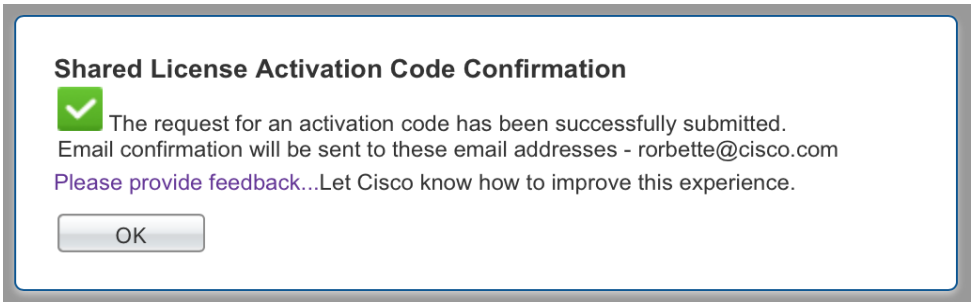


b. Select **Cisco Meeting Server** and enter the MAC address of your first node (used in steps 1 and 2) in the **Source Device MAC Address:** field.

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.

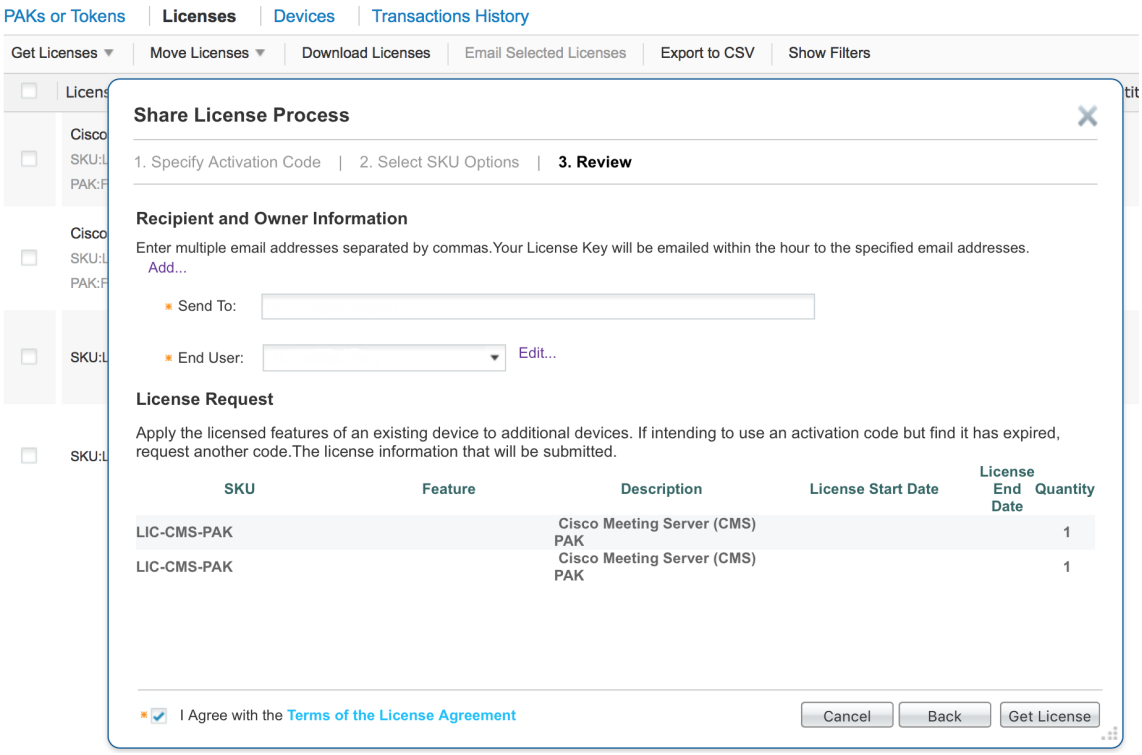


e. You will receive an activation code by email for the target node you entered in step c.

```
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
```

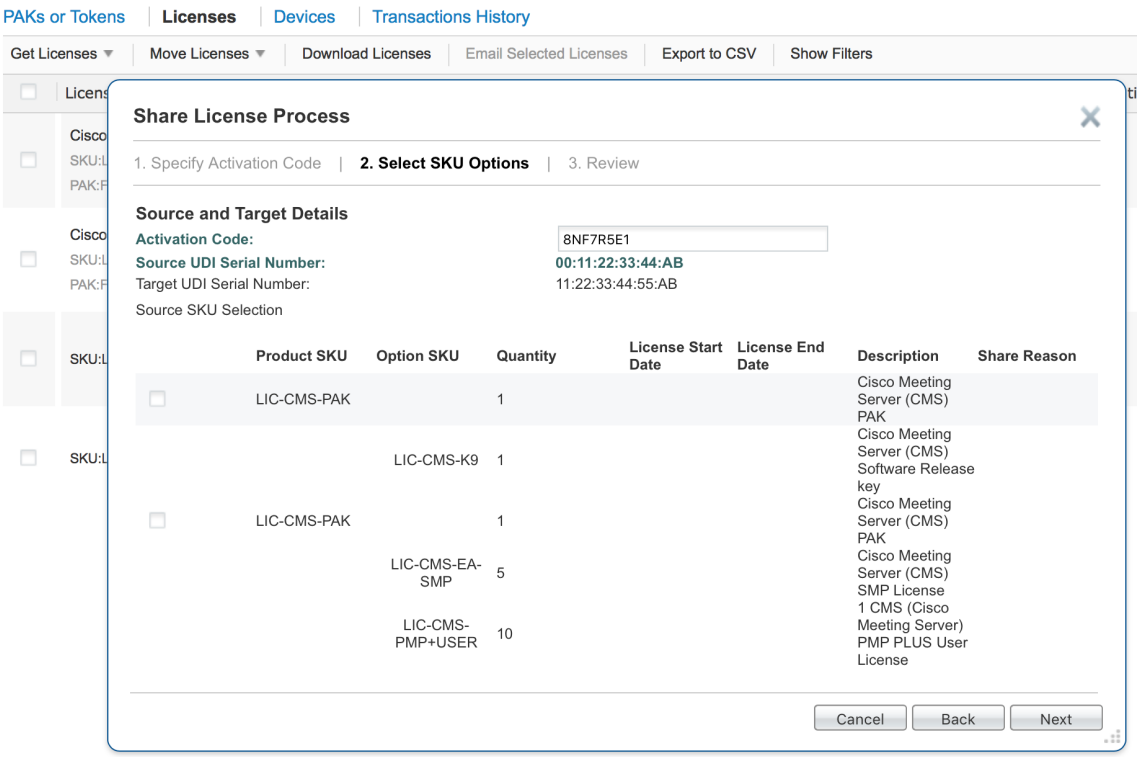
f. Return to the **Licenses** tab, and navigate to **Move Licenses** > **Share Licenses** > **Use Activation Code**.

g. Enter the activation code you received in step e and click **Next**

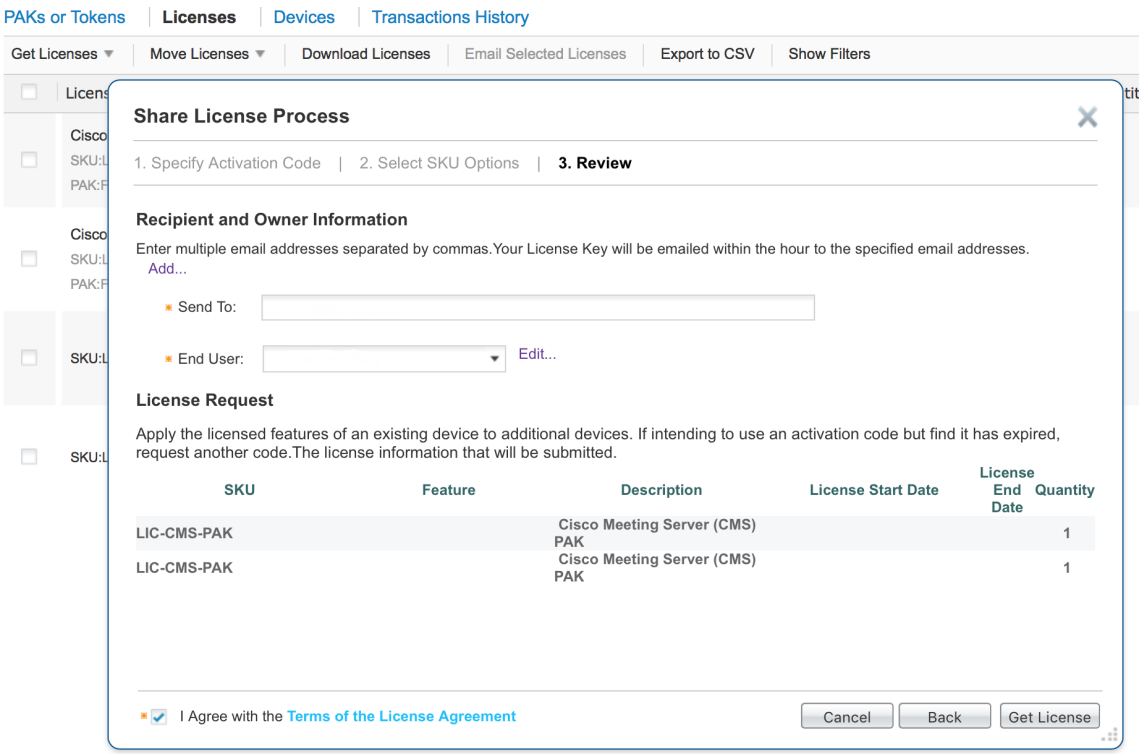


h. Tick the boxes for the features you registered in step 2, and click **Next**.

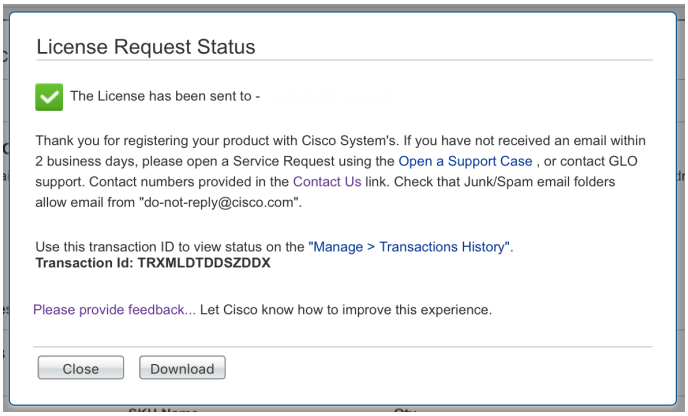
Note: You don't need to tick the box for the **software release key** as this secondary node has already been registered.



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**.

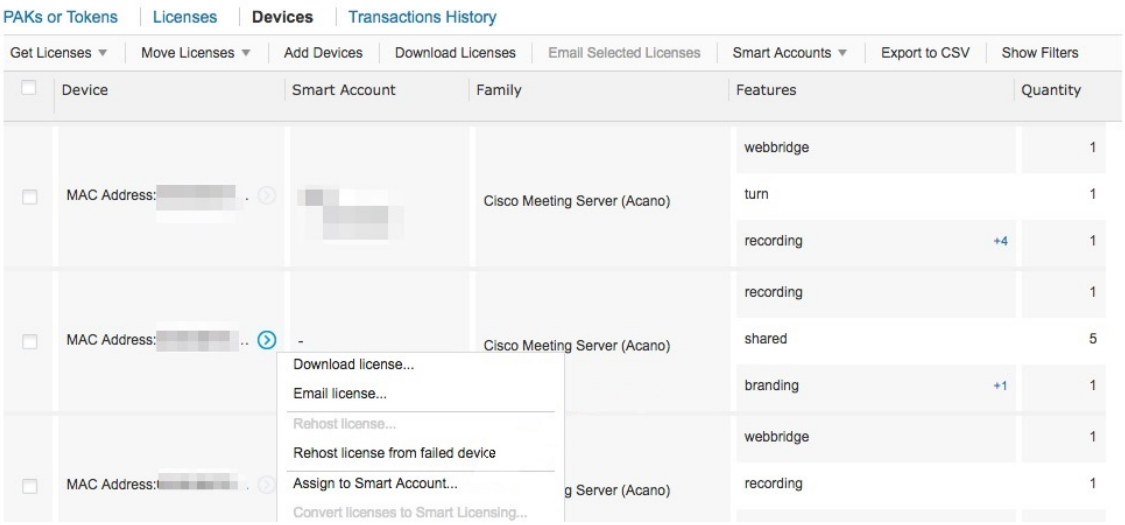


j. Make a note of the "transaction ID" in case there is an issue with the license being released.



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.)
3. Remove the component connection from the Call Bridge Web Admin Interface: to do this, delete the XMPP server details on the **Configuration > General page**. (If this step is missed the XMPP server may continue to use that Call Bridge for authentication despite it no longer having a connection to the database cluster.)

Appendix E Call capacities by Cisco Meeting Server platform

Table 25 below details maximum call capacities on Meeting Servers by upgrading to later software versions. Bold indicates a new feature in that software version. 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 25:Evolution in Meeting Server call capacity

Software version		**2.6 and 2.7**		**2.8 and 2.9**		
Cisco Meeting Server platform		**1000**	**2000**	**1000 M4**	**1000 M5**	**2000**
Individual Meeting Servers or Meeting Servers in a cluster (notes 1,2 3 and 4)	1080p30 720p30 SD Audio	48 96 192 3000	350 700 1000 3000	48 96 192 **1700**	48 96 192 **2200**	350 700 1000 3000
	HD participants per conference per server	96	450	96	96	450
	WebRTC con-nections per Web Bridge 2	100	100	100	100	100
Meeting Servers in a Call Bridge Group	Call type supported	Inbound SIP Outbound SIP Cisco Meeting App		Inbound SIP Outbound SIP Cisco Meeting App	Inbound SIP Outbound SIP Cisco Meeting App	Inbound SIP Outbound SIP Cisco Meeting App
	1080p30 720p30 SD Audio Load limit	48 96 192 3000 96,000	250 **700** 1000 3000 700,000 (note 5)	48 96 192 1700 96,000	48 96 192 2200 96,000	250 700 1000 3000 700,000 (note 5)
	Number of HD participants per conference per server	96	450	96	96	450
	WebRTC con-nections per Web Bridge 2	100	100	100	100	100

Note 1: Maximum of 24 Call Bridge nodes per cluster; cluster designs of 8 or more nodes need to be approved by Cisco, contact Cisco Support for more information.

Note 2: 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.

Note 3: Up to 16,800 HD concurrent calls per cluster (24 nodes x 700 HD calls) applies to SIP or web app calls.

Note 4: A maximum of 2600 participants per conference per cluster depending on the Meeting Servers platforms within the cluster.

Note 5: From version 2.6, the call capacity for Cisco Meeting Server 2000 with Call Bridge Groups enabled, has increased to 700 HD calls, and the loadlimit has increased from 500000 to 700000. The load calculation for the different call resolutions has been updated to match the new 700000 limit. Load limits for other Meeting Server platforms stay as they were previously; these changes only apply to the Cisco Meeting Server 2000.

Note 6: Table 25 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.

E.1 Cisco Meeting Server web app call capacities

This section details call capacities for deployments using Web Bridge 3 and web app.

E.1.1 Cisco Meeting Server web app call capacities – internal calling

Internal calling means that clients can reach the Call Bridge and Web Bridge 3 without going through Cisco Expressway as a reverse proxy or TURN server for media. The web app call capacities for internal calling are shown in Table 26.

Table 26: Cisco Meeting Server web app call capacities – internal calling

Cisco Meeting Server platform	Call Type	Cisco Meeting Server 1000 M4/M5	Cisco Meeting Server 2000	Cisco Meeting Server 1000 (24-node cluster)	Cisco Meeting Server 2000 (24-node cluster)
Individual Meeting Servers or Meeting Servers in a cluster or Meeting Servers in a Call Bridge Group	Full HD	38	280	912	6720
	HD	75	560	1800	13440
	SD	150	800	3600	19200
	Audio calls	500	1000	12000	24000

Web bridge 3 capacity can be scaled by clustering Call Bridges and adding a Web Bridge 3 instance for each Call Bridge. For maximum Web Bridge 3 capacity, deploy the Web Bridges co-resident with each Call Bridge instance. When deploying Web Bridge 3 on a 1:1 basis with

Call Bridge and using internal calls, the maximum web app calls supported is the single instance value (see Table 26) multiplied by the number of instances (up to 24 call bridge nodes).

Note: There is a maximum of 24 Call Bridge plus 24 Web Bridge nodes per cluster; cluster designs of 8 or more pairs of nodes need to be approved by Cisco, contact Cisco Support for more information.

Note: The call setup rate for the cluster should not exceed 20 calls per second for Cisco Meeting Server web app calls.

E.1.2 Cisco Meeting Server web app call capacities – external calling

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: If you are deploying Web Bridge 3 and web app you must use Expressway version X12.6 or later, earlier Expressway versions are not supported by Web Bridge 3.

Table 27: Cisco Meeting Server web app call capacities – external calling

Setup	Call Type	CE1200 Platform	Large OVA Expressway
Cisco Expressway Pair (X12.6 or later)	Full HD	150	150
	Other	200	200

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.1.3 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 Table 26 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 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, 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 Cisco Meeting 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 Cisco Meeting 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 Cisco Meeting App users to a Lync meeting, and can use conference controls to mute Cisco Meeting App users or disconnect them. For Cisco Meeting App users joining a Lync scheduled conference, they will see the video from up to five Lync participants, as well as video from the Cisco Meeting App users. Lync users see video in a gallery format from all of the Cisco Meeting App users, as well as the Lync users in the meeting. Both Lync users and Cisco Meeting 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.

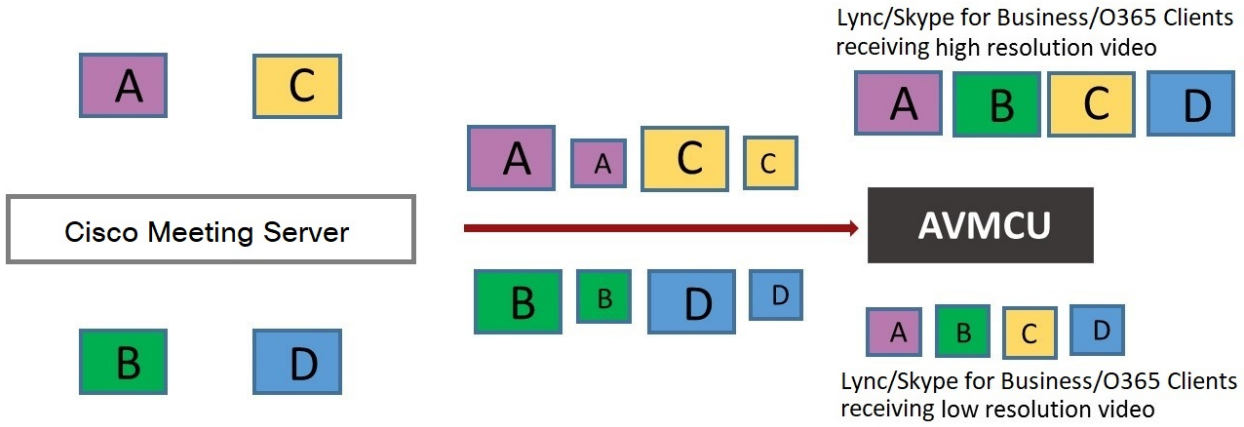
G.2 Consistent meeting experience in dual homed conferences

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 59. 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 59: 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 Cisco Meeting App users.

- Both Lync client users and Cisco Meeting App users see familiar screen layouts.
- Both Lync client users and Cisco Meeting 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 Cisco Meeting 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 Cisco Meeting App users in spaces. and in direct calls between a Cisco Meeting 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 Cisco Meeting 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 Cisco Meeting 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,
- Cisco Meeting 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.
- Cisco Meeting 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 Cisco Meeting App **users:**

- Information on local muting and unmuting of a Cisco Meeting App user is not passed to Lync clients in dual homed conferences. However, if a Lync client remotely mutes a Cisco Meeting App user and the Cisco Meeting App unmutes itself, the Meeting Server tells the Lync clients about the unmuting.
- When a remote participant tries to unmute a Cisco Meeting App user, the Cisco Meeting App user will remain locally muted. Note: other participants will still see them as unmuted, although they are actually muted.
- The Cisco Meeting App shows the mute/unmute state using its own icons. Meeting Server icons are not overlaid on the Cisco Meeting 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 13.5.

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 Cisco Meeting 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

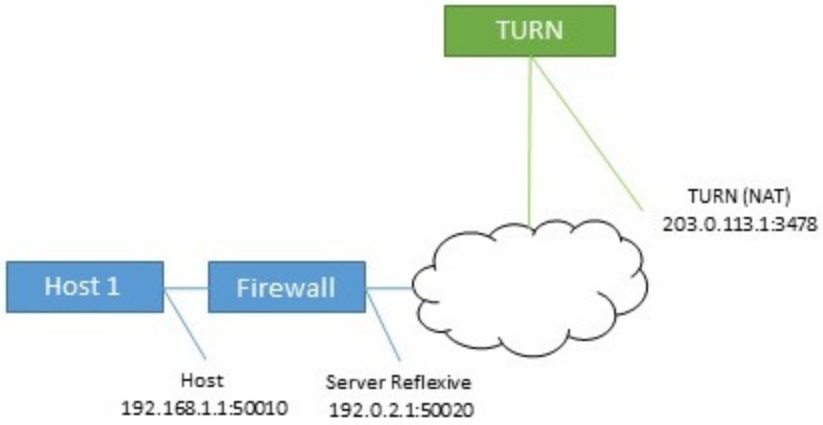
H.1.1 Host candidate

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

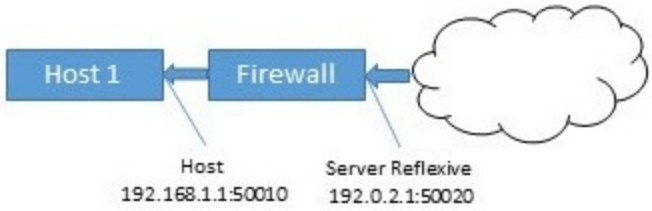
The server reflexive candidate is the address that the TURN server sees incoming packets coming from. To determine this, the host sends packets to a defined port on the TURN server (normally port 3478) and the TURN server replies with information about where the packets came from.

Figure 60: 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.

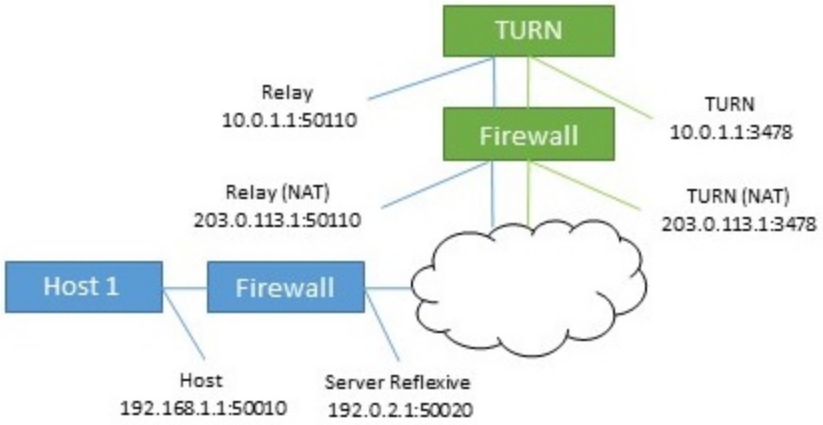
Figure 61: Effect of a host behind a firewall carrying out NAT



H.1.3 Relay candidate

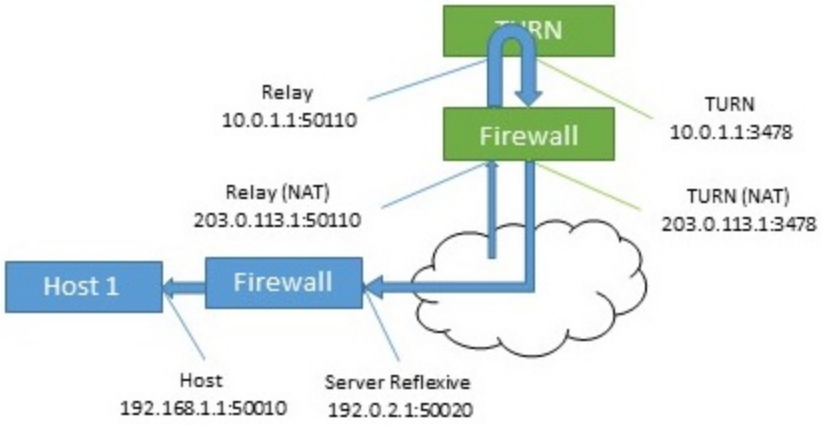
The final candidate is the relay candidate. This candidate is created by the TURN server in response to requests from the host. The relay address of this candidate is the TURN server interface address, when NAT is used the relay address is changed to an address from NAT.

Figure 62: Relay candidate



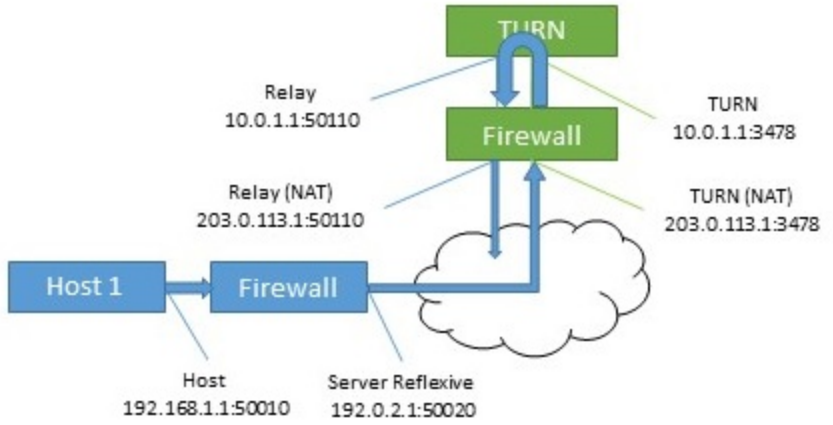
Data sent to this relay address is then sent back to the host via the TURN server.

Figure 63: 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 64: 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

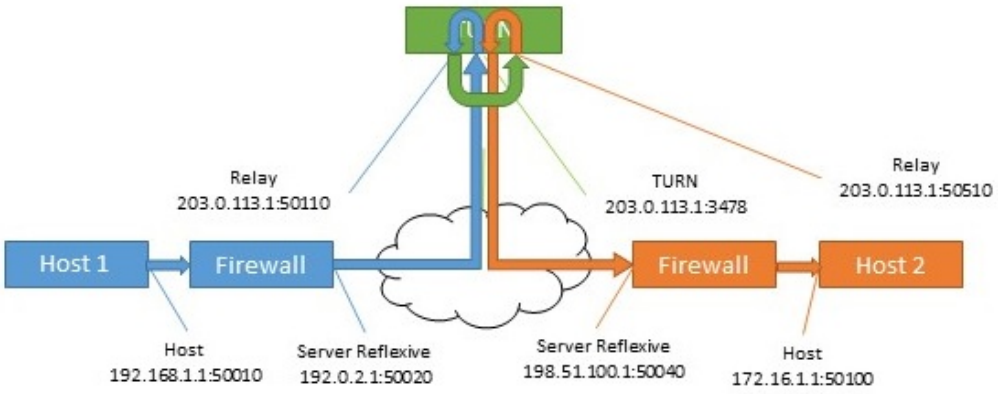
Source	Destination Type	Destination address
Host (192.168.1.1:50010)	Host	172.16.1.1:50100
Host (192.168.1.1:50010)	Server Reflexive	198.51.100.1:50040
Host (192.168.1.1:50010)	Relay	203.0.113.1:50510
Relay (10.0.1.1:50110)	Host	172.16.1.1:50100

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 65: 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 66: Host to host media path using relay to relay path (with NAT)

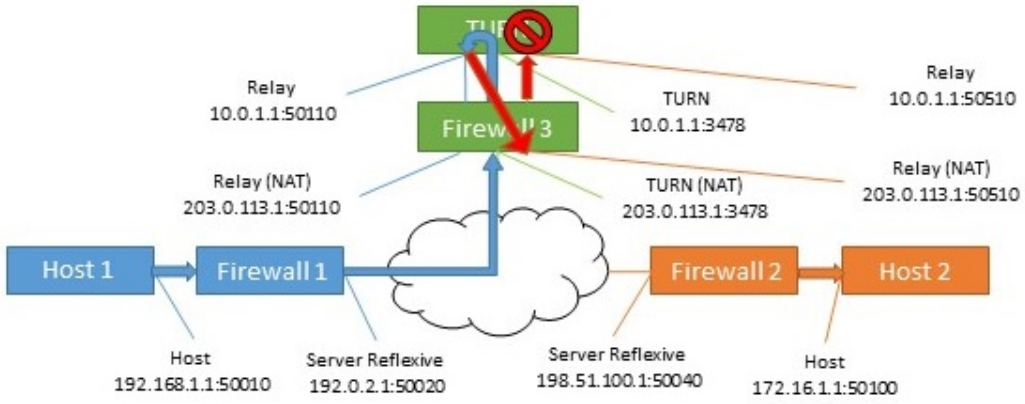


Table 30: Host to host media path using relay to relay path (with NAT)

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.

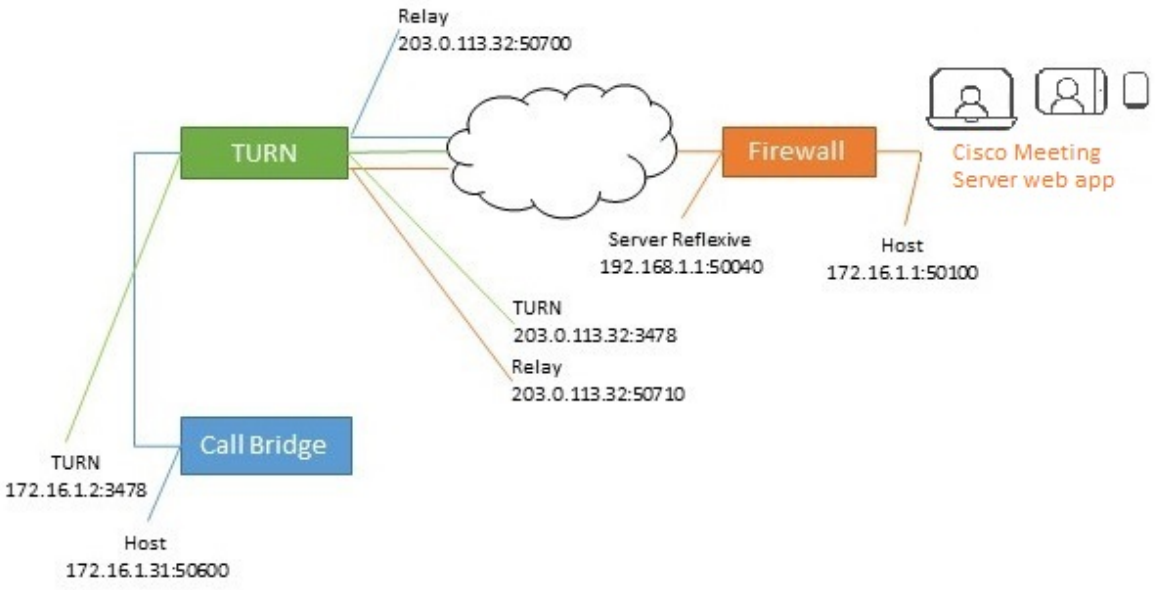
For details on how to enable this functionality, refer to your firewall documentation.

H.4 TURN server, NAT and the Cisco Meeting App

The effect of NAT on external Cisco Meeting App users needs to be considered in deployments where one Meeting Server is configured as a Core server with an internal interface, while another Meeting Server is configured as an Edge server set up on with two interfaces (internal and external). For Cisco Meeting App users working remotely, the Cisco Meeting App may be unable to see any ephemeral UDP ports.

In this case there is no server reflexive candidate for the Call Bridge, since the address seen by the TURN server is the same as the host candidate.

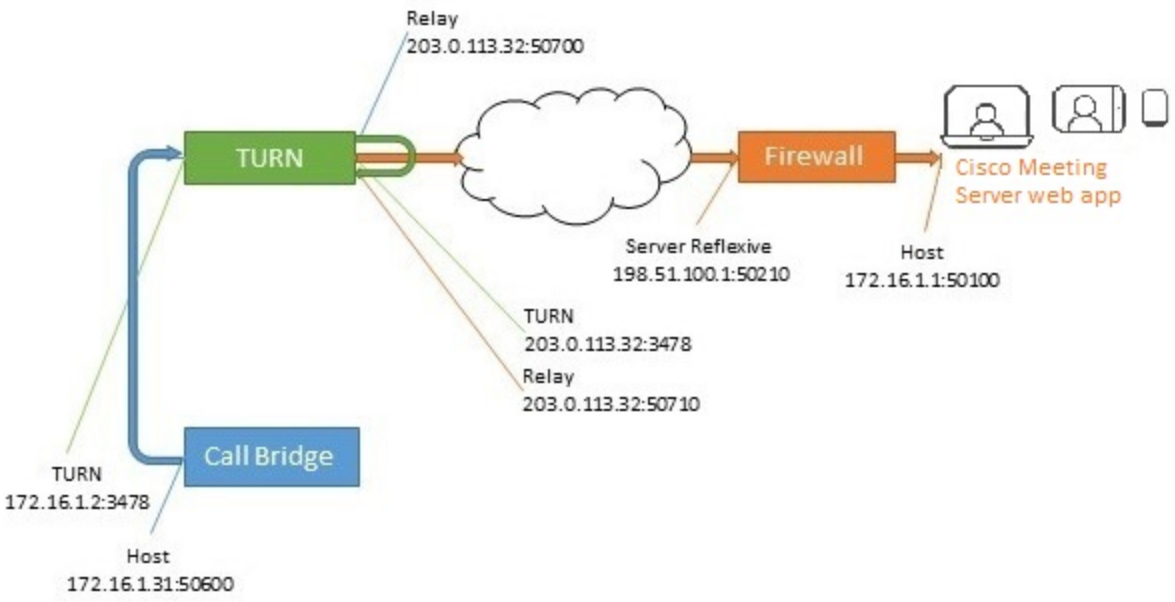
Figure 67: Split Meeting Server deployment with external Cisco Meeting App users (no NAT)



Since the Call Bridge running on the Core server is only on the internal network it has no route to the Cisco Meeting App's host address, server reflexive or the relay address. Likewise the Cisco Meeting App cannot see the Call Bridge's host, or its relay address.

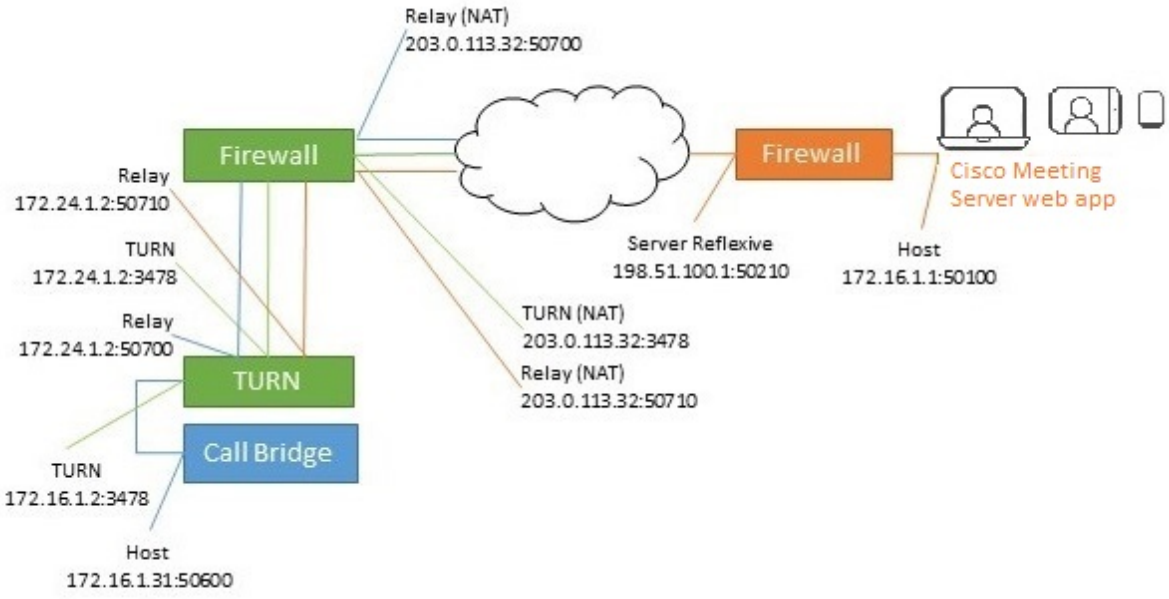
However, the relay ports can see each other, and therefore a path for media can be established.

Figure 68: Relay ports establishing the media path



As in the general case, when the TURN server is behind a NAT this picture is further complicated.

Figure 69: Split Meeting Server deployment with external Cisco Meeting App users (with NAT)



The solution for this is identical to the general case. The source address of traffic needs to be rewritten by the firewall so that it appears as coming from the correct address.

Figure 70: Relay ports establishing the media path

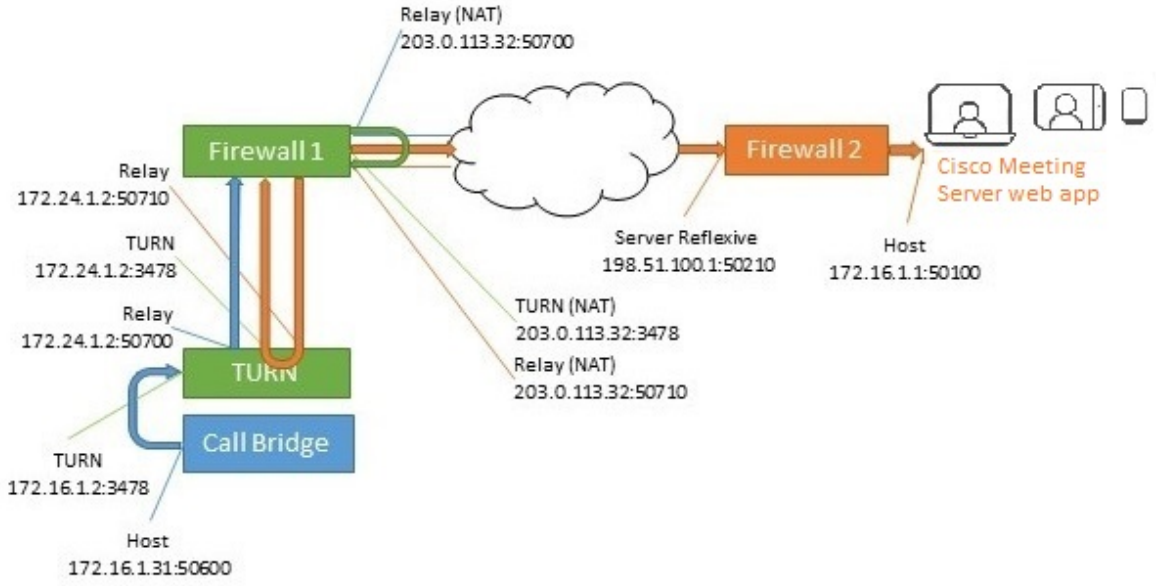


Table 32: Host to host media path using relay to relay path (with hairpin NAT)

Source address (in packets)	Destination	Action at destination
172.16.1.31:50600	172.16.1.2:3478	TURN internally maps this to the relay address for this source, and sends to the far end's relay.
172.24.1.2:50700	203.0.113.32:50710 via Firewall	Firewall 1 rewrites both source and destination addresses.
203.0.113.32:50700	172.24.1.2:50710	TURN server internally maps traffic from relay to assigned host.
172.24.1.2:3478	198.51.100.1:50510 via Firewall	Firewall 1 rewrites source address.
203.0.113.32:3478	198.51.100.1:50510	Firewall 2 rewrites destination address.
203.0.113.32:3478	172.16.1.1:50100	Arrives at final destination.

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
- CMA user settings
- API

I.1 General

Use the **Configuration > General** page to set up and configure:

- **XMPP server settings**. Use these fields to configure the settings through which the Call Bridge communicates with the XMPP server. See *Web Admin interface settings for XMPP*Connecting Call Bridges to the XMPP Server . Use MMP commands to configure the XMPP server itself. See Configuring the MMP. Note that you only need to configure and enable the XMPP server if you are using the Recorder or Streamer components or any Cisco Meeting App (including the WebRTC Client).
- **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 9.1.
- **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*.
- **Web bridge settings**. Use these settings if you are using Cisco Meeting App These settings allow the Call Bridge to communicate with the Web Bridge server. While you can still configure one Web Bridge via the Web Admin Interface, if you have multiple Web Bridges

we strongly suggest that you use only the API to configure them all, as described in Section 8.3

- **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*.
- **External access**.Use these settings to enter the URI for the Web bridge that Cisco Meeting App will use to access it. This field must be filled in manually on every Call Bridge in the cluster for Cisco Meeting App to generate WebRTC URLs.
  Enter the IVR telephone number to add an extra option in the invite list, and to provide a call in "Phone" line to the meeting email and invitation templates.

I.2 Active Directory

If you want users to use Cisco Meeting 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 (CMA), 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 Meeting 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.

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 CMA user settings

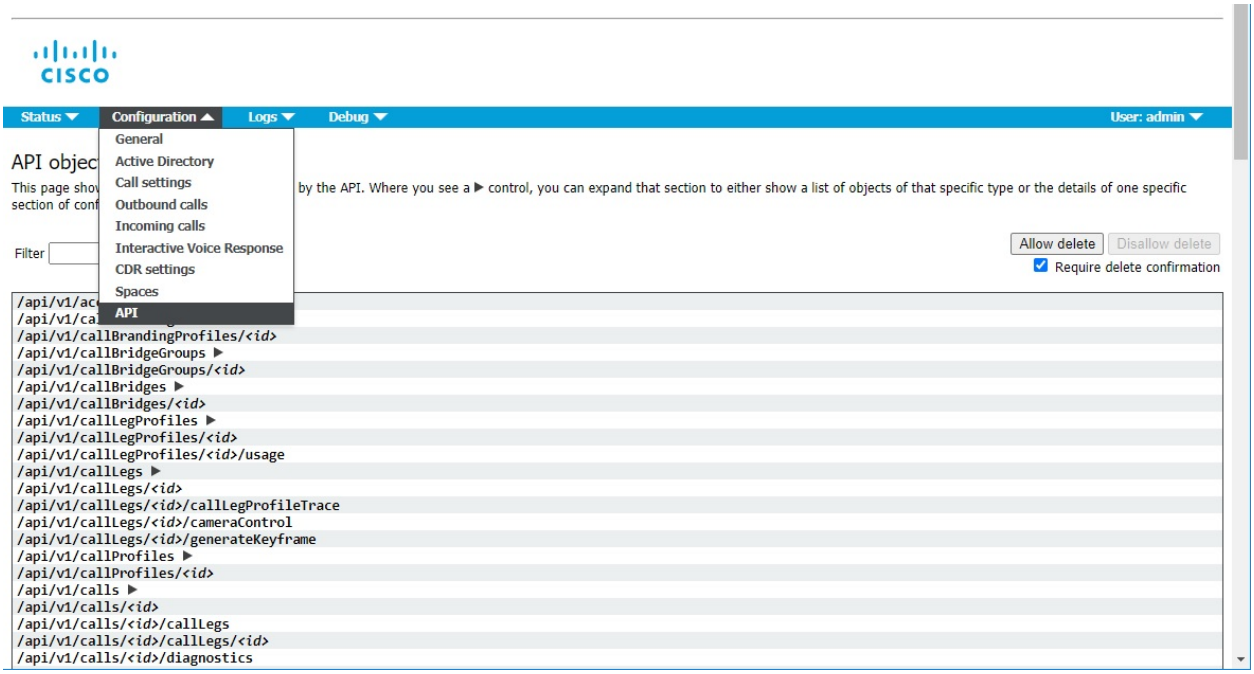
Use the **Configuration > CMA user settings** page to allow or not allow incoming calls to Cisco Meeting App users.

By default incoming calls to Cisco Meeting Apps are allowed, however this behavior can be changed so that incoming calls are not allowed to users of the Cisco Meeting App.

I.9 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 71.

Figure 71: 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.

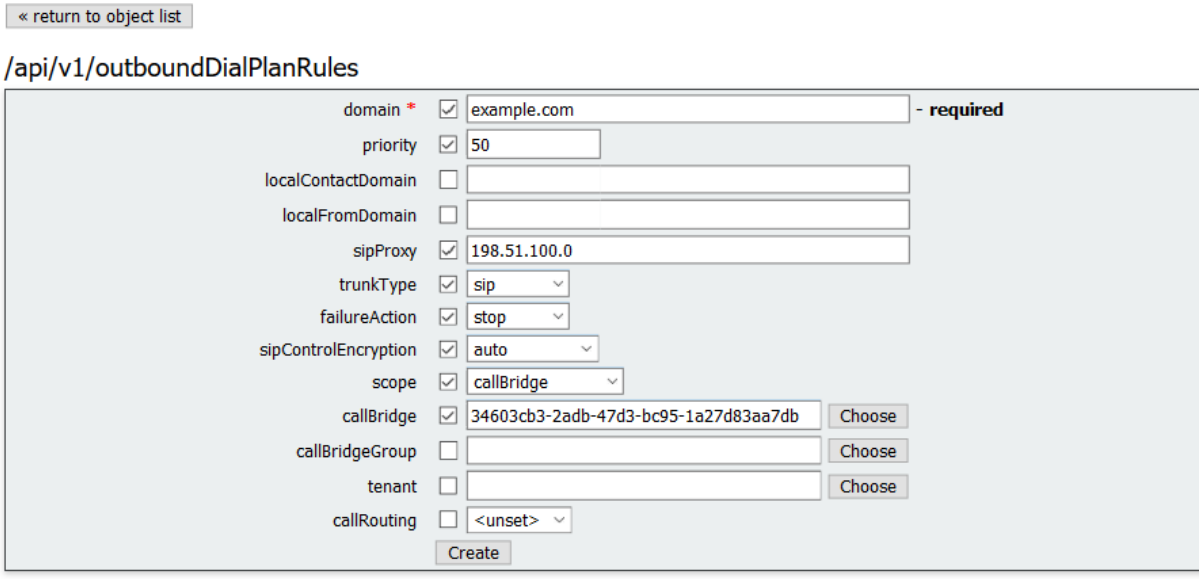
The following examples are provided:

- 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.

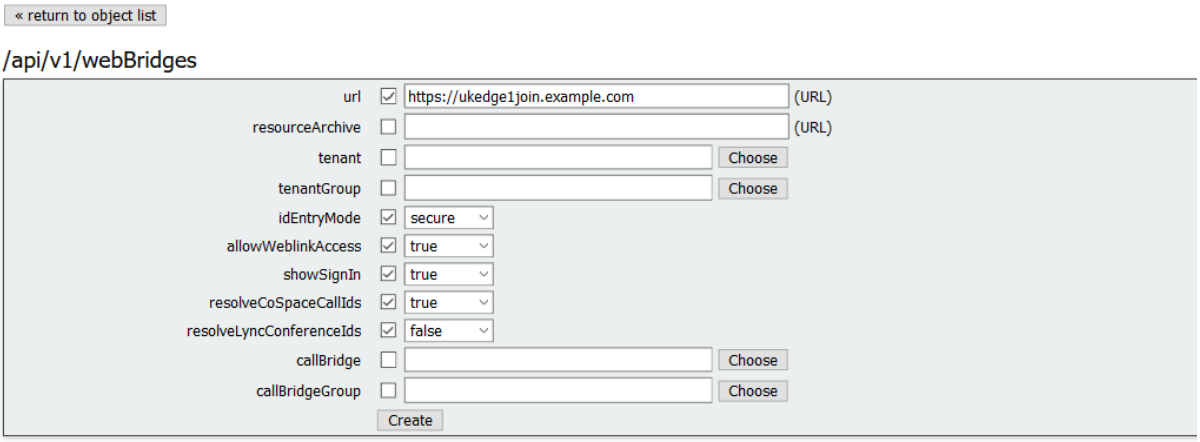
7. Verify the parameters of the outbound dial plan rule before clicking **on return to object list** button at top of page.

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.

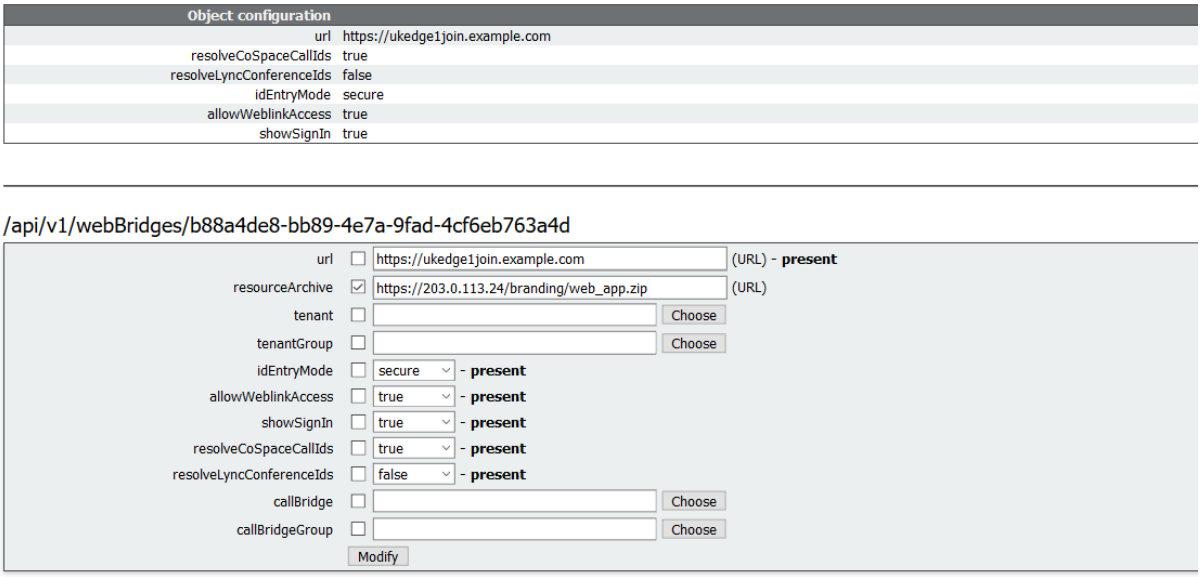


4. Click **Create**.

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.

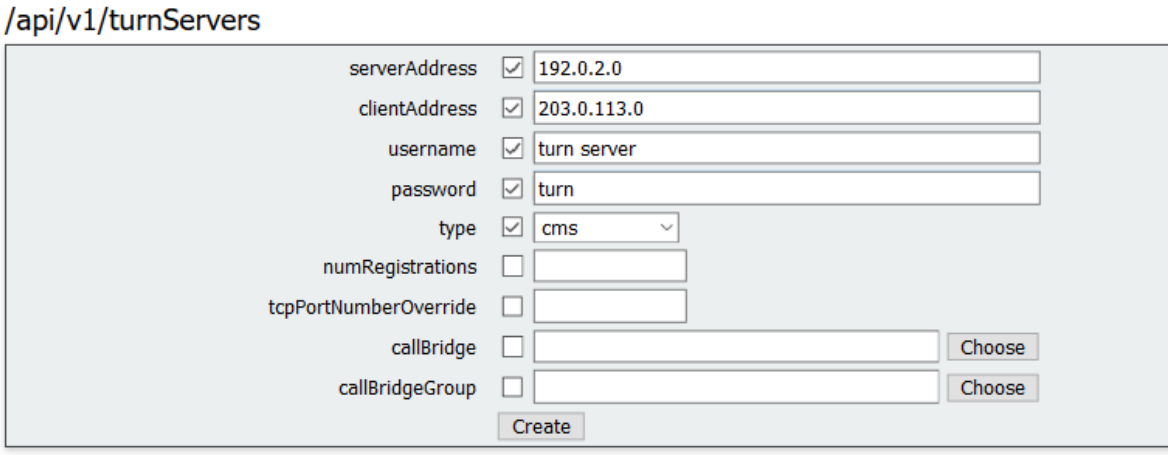
J.4 Setting up the TURN Server and connecting to the Call Bridge

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 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 Cisco Meeting 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 Cisco Meeting Apps should use to reach this TURN server.



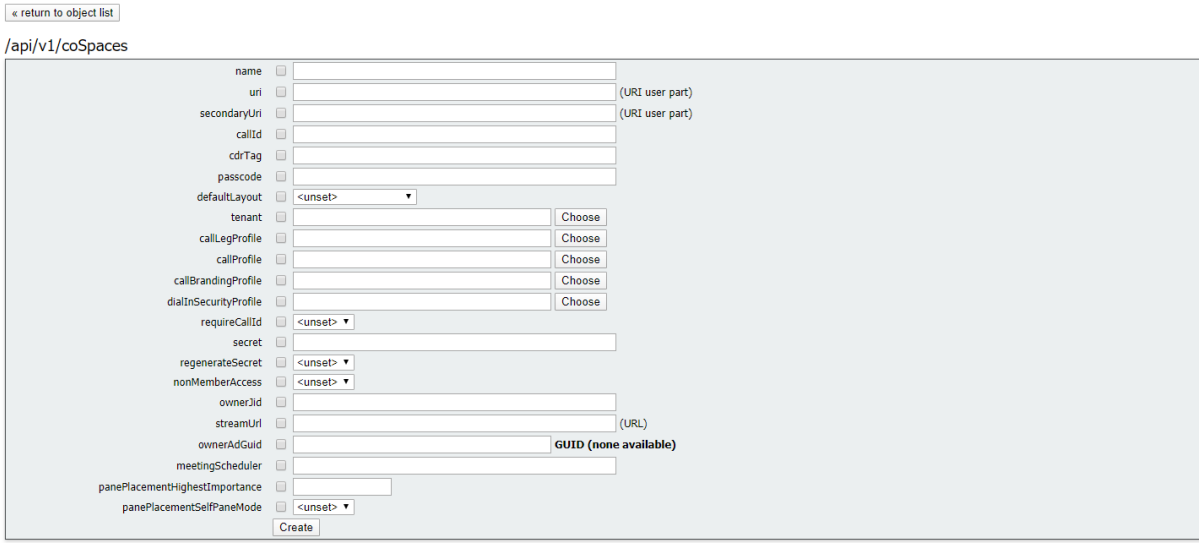
4. Click **Create**
5. Verify the parameters of the TURN server before clicking **on return to object list** button at top of page.

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

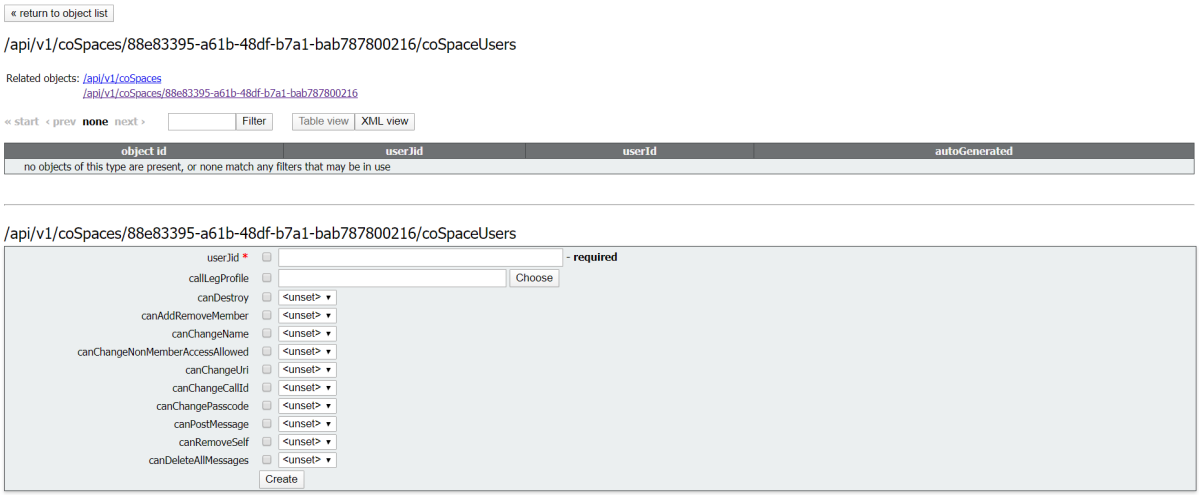
3. Enter the **name**, **uri**, **secondary uri**, **call id**, **passcode** and any other parameters as appropriate



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**.



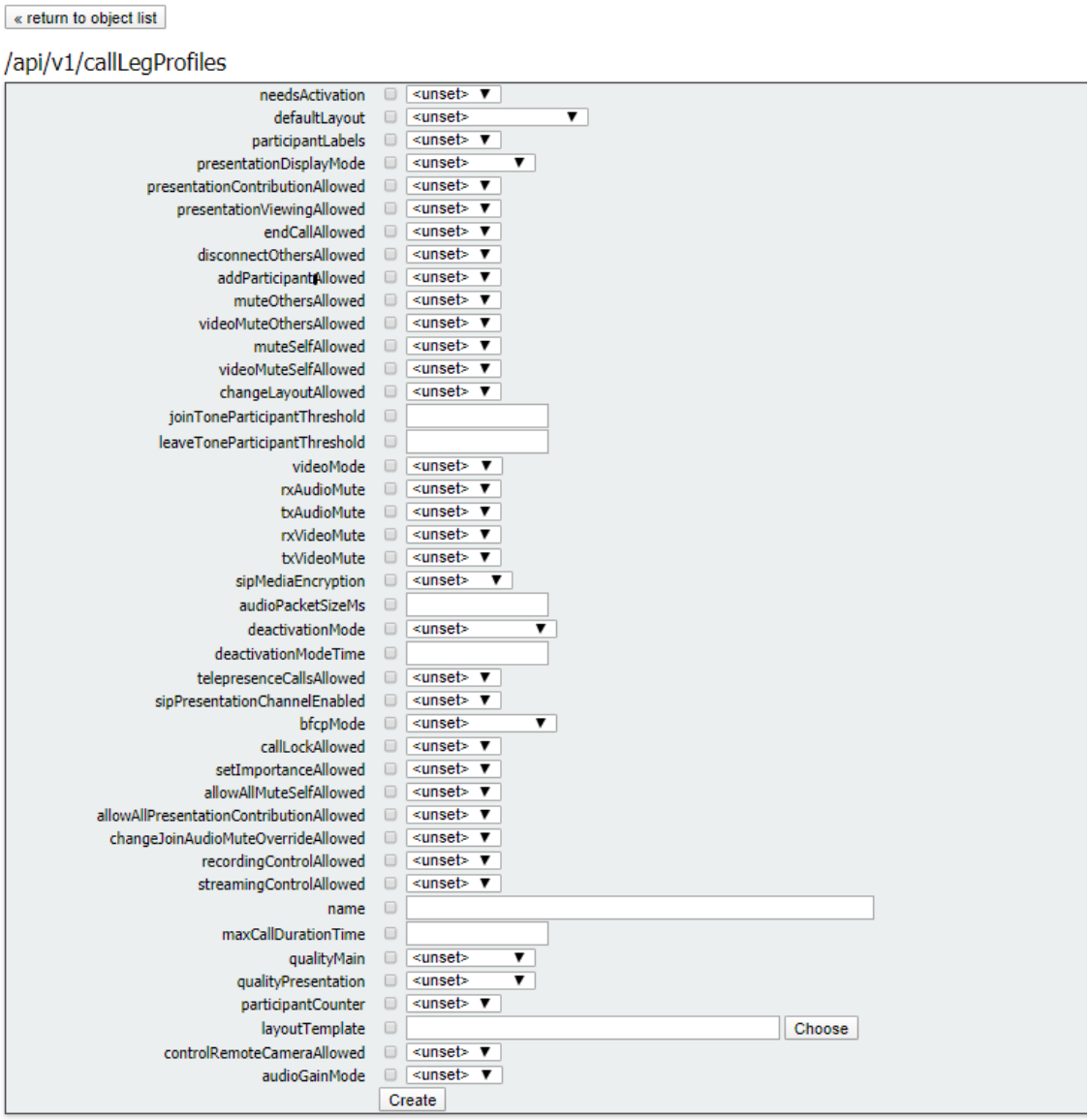
3. Click **Create**
4. Repeat for all other Members that need adding to the space.

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 space by 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.

Appendix K Deploying Web Bridge 3 to use Cisco Meeting Server web app

In version 2.9, Meeting Server introduces the new Cisco Meeting Server web app which is a browser-based client for Cisco Meeting Server that lets users join meetings (audio and video). To use this feature you need to deploy the new Web Bridge 3. In addition, Meeting Server version 2.9 still offers the original Cisco Meeting App WebRTC (also referred to here as Web Bridge 2).

In this release Cisco Meeting Server web app is not yet fully featured. It is intended that in due course it will support virtually the same feature set and supersede Cisco Meeting App WebRTC.

Note: For a full list of features that are not currently supported by web app in 2.9 and those features that we plan to support in the future, see Cisco Meeting Server web app Important Information for more details.

Note: The Web Bridge 3 component that supports web app cannot be run on the Acano X-series. However, the Acano X-Series can still run the Call Bridge and be part of the same cluster. Web Bridge 3 will need to be run on Cisco Meeting Server 1000 and 2000 platforms and other specification-based VM.

Note: Web app does not require XMPP. The XMPP component will be removed from a future version. Cisco Meeting App WebRTC still requires XMPP.

CAUTION: Important notes for Expressway users
If you are deploying Web Bridge 3 and web app you must use Expressway version X12.6 or later, earlier Expressway versions are not supported by Web Bridge 3. If you are deploying solely Web Bridge 2 and Meeting App for WebRTC you can continue to use Expressway versions earlier than X12.6.

K.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.
- A port must be opened on an interface (using **`webbridge3 c2w listen`**) to allow the callbridge to connect to the webbridge3 (the webbridge listens on that port). This is why

you have to give the address with this port when you do the API request to tell a callbridge about this webbridge. This connection must be secured with certificates.

- We recommend you protect that opened port from external access – it only needs to be reachable from callbridges.
- The callbridge uses the certificate set using **`callbridge certs`** and the webbridge uses the certificate set using **`webbridge3 c2w certs`**.
- The webbridge will trust certificates of callbridges that have been signed by one of those in its trust store, set by **`webbridge3 c2w trust`**.
- The callbridge will trust webbridges that have certificates signed by one of those in its trust store, set by **`callbridge trust c2w`**.
- The webbridge3 https certifcates and ports are the same as for webbridge2, it allows you to reach the web client using https and can be used in the same deployment at the same time.
- If the webbridge3 c2w certificate requires extended key usage, it should be "server authentication", and the callbridge certificate extended key usage should be "client authentication". However, these extensions are optional and if the certificate doesn't have them, the Web Bridge 3 will assume any usage is possible.
- You do not need a certificate signed by a public authority – you can use self-signed certificates created within the MMP.
- The SAN/CN must match the FQDN or IP address that is used in the c2w:// url used to register the Web Bridge 3 in the callbridge API. (If this does not match, the callbridge will fail the TLS negotiation, rejecting the certificate presented by the webbridge, and will fail to connect with the webbridge.)
- For general certificate information, see the Certificate Guidelines appropriate for your deployment.

The figures below show the flow of a typical Web Bridge 2 setup compared to that for Web Bridge 3.

Figure 72: Web Bridge 2 setup flow diagram

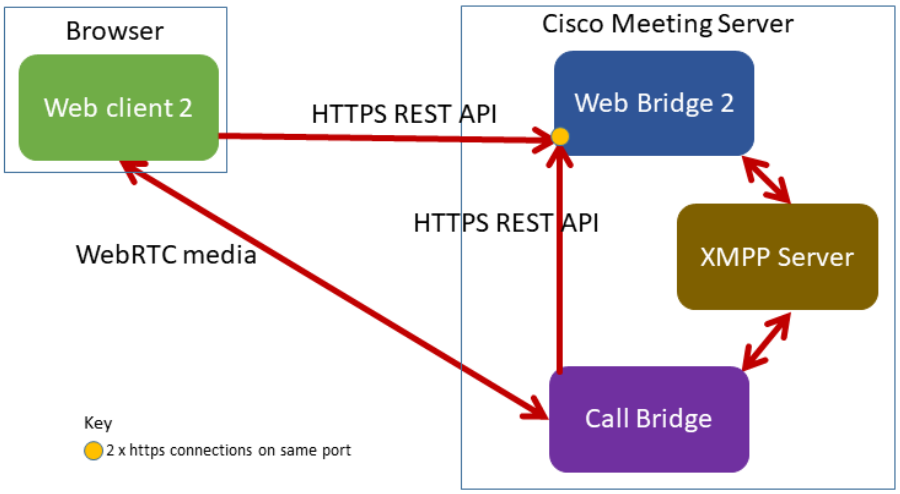
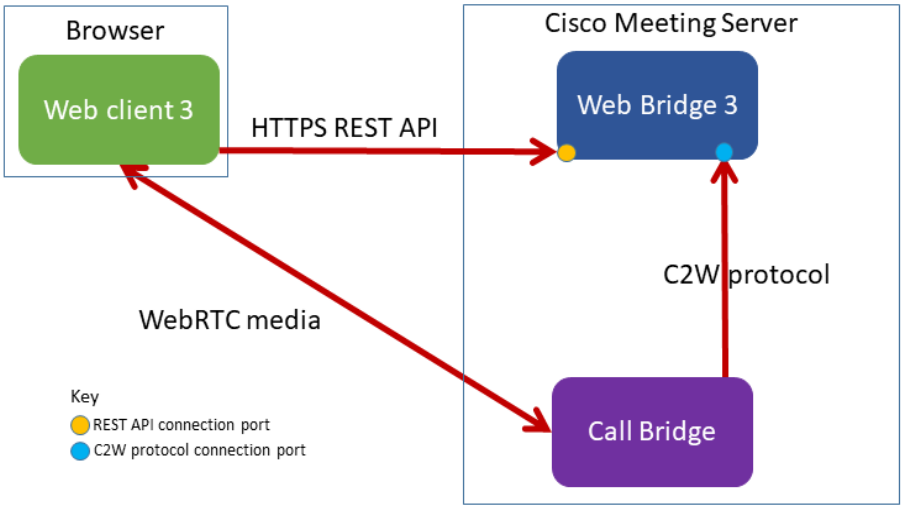


Figure 73: Web Bridge 3 setup flow diagram



K.2 Configuring Meeting Server to use Web Bridge 3

If upgrading your Meeting Server to 2.9, by default, this release will use your existing Web Bridge 2 configuration. However, you can configure Web Bridge 3 to operate at the same time as Web Bridge 2. Web Bridge 2 uses XMPP and Web Bridge 3 uses Call Bridge to Web Bridge (C2W)

protocol connections so they can work in parallel, however, they will need to be configured to use different ports as shown in Figure 73.

Note: You don't need to configure an XMPP Server for Web Bridge 3.

Web Bridge 3 is similar to Web Bridge 2 for configuration and setup which is done using MMP commands via SSH. The main difference is that Web Bridge 2 requires configuring an HTTPs port, whereas Web Bridge 3 requires configuring an HTTPS port and a C2W port.

To configure Meeting Server to use Web Bridge3:

1. SSH into the MMP and log in.
2. Use the **webbridge3** command in the MMP to configure webbridge3. To display the webbridge 3 usage, enter: **help webbridge3**

```
> help webbridge3

    Usage:
    webbridge3
    webbridge3 restart
    webbridge3 enable
    webbridge3 disable
    webbridge3 https listen <interface:port allowed list>
    webbridge3 https certs  <key-file> <crt-fullchain-file>
    webbridge3 https certs none
    webbridge3 http-redirect (enable [port]|disable)
    webbridge3 c2w listen <interface:port allowed list>
    webbridge3 c2w certs  <key-file> <crt-fullchain-file>
    webbridge3 c2w certs none
    webbridge3 c2w trust <crt-bundle>
    webbridge3 c2w trust none
    webbridge3 options <space-separated options>
    webbridge3 options none
    webbridge3 status
```

More detail can be found in the Cisco Meeting Server Release 2.9 MMP Command Line Reference.

3. (Optional) Set up a port for HTTP connections. This port will be opened for all Meeting Server interfaces on which the web app has been configured. Incoming HTTP connections will be automatically redirected to the matching HTTPS port for the interface they arrived on. The default port, if you don't specify one in **webbridge3 http-redirect enable [port]**, is 80.
4. Configure the port for the HTTPS service to listen to. To configure it to listen on port 443

of the a interface:

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

5. Set the HTTPS certificates. These are the certificates that will be presented to web browsers so they need to be signed by a certification authority and the hostname/purpose etc needs to match. (The certificate file is the full chain of certificates that starts with the end entity certificate and finishes with the root certificate.) Enter the command:

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

6. Configure the C2W connection. We recommend that you make this address/port accessible from the Call Bridge(s) only. The following command sets it in port 9999 of interface a:

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

   Note that here we use the example of port 9999, however, it can be any available port on your network. It's not a fixed port, unlike 443.

7. Configure the C2W connection certificates. You need to configure the SSL Server certificates used for the C2W connection. (See "Configuring Call bridge to use C2W connections" below for certificate requirements, and more information can be found in this FAQ.)

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

8. The Web Bridge 3 C2W server is expecting Call Bridges to present a client certificate – it will verify whether to trust them using the trust bundle provided by the following command:

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

9. Now enable Web Bridge 3:

   ```
   webbridge3 enable
   ```

K.3 Configuring Call bridge to use C2W connections

C2W certificates are used for the connection between Call Bridge and Web Bridge 3. For the Call Bridge to make a C2W connection to a Web Bridge 3, you need to specify a C2W trust store to verify certificates against, i.e. the ones presented by the Web Bridge 3 that were configured in step 7 above.

1. Use the **`callbridge`** command in the MMP to display the Call Bridge usage, enter: **`help callbridge`** to display:

   ```
   > help callbridge
   Configure CMS callbridge
   ```

```
Usage:

    callbridge listen <interface allowed list>
    callbridge prefer <interface>
    callbridge certs <key-file> <crt-file> [<cert-bundle>]
    callbridge certs none
    callbridge trust xmpp <bundle>
    callbridge trust xmpp none
    callbridge trust c2w <bundle>
    callbridge trust c2w none
    callbridge add edge <ip address>:<port>
    callbridge del edge
    callbridge trust edge <trusted edge certificate bundle>
    callbridge trust cluster none
    callbridge trust cluster <trusted cluster certificate bundle>
    callbridge restart
```

2. Set the certificates for the Call Bridge:

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

3. Set the C2W trust store that will be used to validate the SSL Server certificate presented by the Web Bridge 3. (For more information, see this FAQ.)

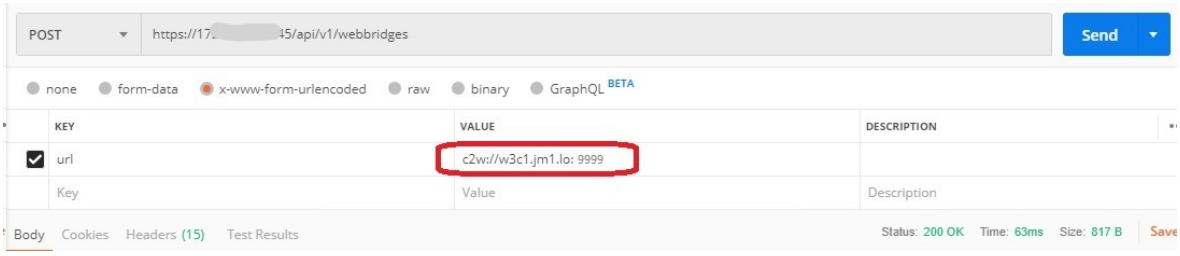
   ```
   callbridge trust c2w c2w-callbrige-trust-store.crt
   ```

4. Now restart Call Bridge:

   ```
   callbridge restart
   ```

5. Register the Web Bridge 3 URL to the running callbridge REST API in the same way as you would for Web Bridge 2, as shown below, i.e. POST to /api/v1/webbridges with the "url" parameter. The URL protocol indicates if it is webbridge2 or webbridge3. So, if the protocol is http:// or https:// then the webbridge is treated as webbridge2, and if you specify c2w:// protocol in the URL then it will be handled as a webbridge3 connection.

Figure 74: Registering Web Bridge 3 URL to the Call Bridge API



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.

Cisco Trademark