Configure Cisco Meeting Server and Skype for Business

Contents

Introduction

This document describes how to configure Cisco Meeting Server (CMS) CallBridge Cluster with Skype for Business as a complement of the official guides. This document provides an example of a single CallBridge and another example of a three CallBridge cluster, but additional CallBridges can be added as necessary. A two CallBridge cluster is also supported.

Contributed by Rogelio Galindo and edited by Viridiana Fuentes, Cisco TAC Engineers.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

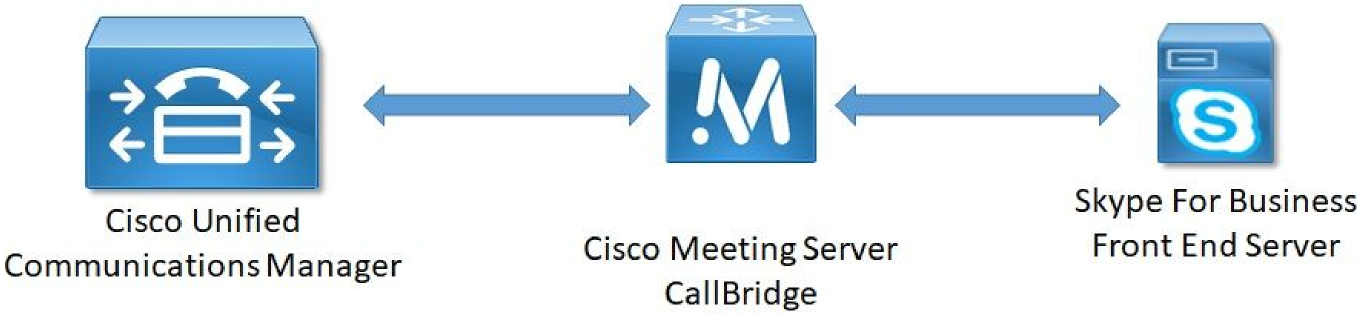
- Cisco Meeting Server (CMS)
- Domain Name Server (DNS)
- Skype For Business
- Application Programming Interface (API)

Note:The configuration guide can be found here: https://www.cisco.com/c/dam/en/us/td/docs/conferencing/ciscoMeetingServer/Deployment_Guide/Version-2-2/Cisco-Meeting-Server-2-2-Scalable-and-Resilient-Deployments.pdf

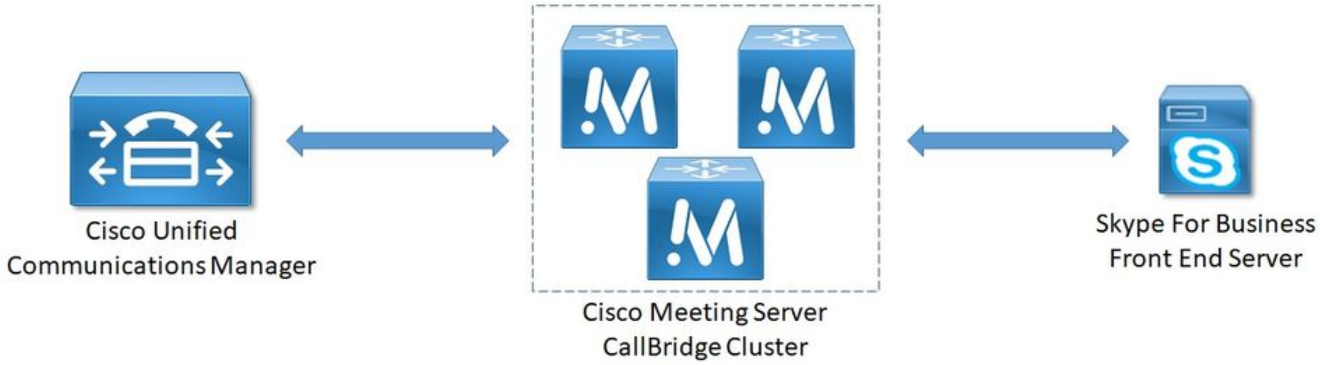
Components Used

- 3 CMS servers running a CallBridge cluster, software version 2.2.2.
- Skype for Business 2015
- Active Directory (AD) Windows Server 2012
- Secure Shell (SSH) client
- Secure File Transfer Protocol (SFTP) client such as WinSCP or similar
- API program such as Postman or similar
- Remote Desktop session for Active Directory, DNS and Skype server

Network Topology - Single CallBridge


Cisco Unified Communications Manager
Cisco Meeting Server CallBridge
Skype For Business Front End Server

Network Topology - Clustered CallBridges


Cisco Unified Communications Manager
Cisco Meeting Server CallBridge Cluster
Skype For Business Front End Server

Callbridge Certificate Requirements - Single CallBridge

Table 1a provides an example of the CallBridge certificate for a single CallBridge environment.

Table 1a

CallBridge Certificates	Description
Single CallBridge	
CN:cms.uc.local	CallBridge FQDN

Callbridge Certificate Requirements - Clustered CallBridges

Table 1b provides an example of the CallBridge certificates for a clustered CallBridge environment. A single certificate can be shared across the CallBridges in a cluster.

Table 1b

Callbridge Certificates	Description
Server 1: cms1.uc.local	
CN:cms.uc.local	CallBridge cluster FQDN. This record must resolve to all CallBridge cluster peers.
SAN:cms.uc.local	CallBridge cluster FQDN. This record must resolve to all CallBridge cluster peers.
SAN:cms1.uc.local	CallBridge 1 FQDN.
SAN:cms2.uc.local	CallBridge 2 FQDN.
SAN:cms3.uc.local	CallBridge 3 FQDN.
Server 2: cms2.uc.local	
CN:cms.uc.local	CallBridge cluster FQDN. This record must resolve to all CallBridge cluster peers.
SAN:cms.uc.local	CallBridge cluster FQDN. This record must resolve to all CallBridge cluster peers.
SAN:cms1.uc.local	CallBridge 1 FQDN.
SAN:cms2.uc.local	CallBridge 2 FQDN.
SAN:cms3.uc.local	CallBridge 3 FQDN.
Server 3: cms3.uc.local	
CN:cms.uc.local	CallBridge cluster FQDN. This record must resolve to all CallBridge cluster peers.
SAN:cms.uc.local	CallBridge cluster FQDN. This record must resolve to all CallBridge cluster peers.
SAN:cms1.uc.local	CallBridge 1 FQDN.

SAN:cms2.uc.local	CallBridge 2 FQDN.
SAN:cms3.uc.local	CallBridge 3 FQDN.

The CMS CLI can be used to view the contents of a certificate:

```
cms1> pki inspect cmsuccluster.cer
Checking ssh public keys...not found
Checking user configured certificates and keys...found
File contains a PEM encoded certificate
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            60:00:00:00:21:db:36:e8:b9:0d:96:44:41:00:00:00:00:00:21
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: DC=local, DC=uc, CN=DC-CA
        Validity
            Not Before: Mar 16 19:00:53 2018 GMT
            Not After : Mar 16 19:10:53 2020 GMT
        Subject: C=US, ST=NC, L=RTP, O=Systems, OU=Cisco, CN=CMS.UC.local
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b8:41:69:d9:1d:47:ef:b1:23:70:ae:69:da:e3:
                    ff:12:f8:97:2b:ee:1e:c0:6c:66:e4:95:3f:8a:74:
                    4d:ec:fc:1e:0d:38:56:1b:00:5c:ce:6d:d3:68:13:
                    e4:9d:b6:e7:7d:de:c4:a4:f3:00:02:11:e5:33:06:
                    b4:f6:64:29:c3:77:62:a9:dc:9d:ad:a2:e9:c1:0b:
                    72:f4:18:af:df:d3:e3:f4:4a:5d:66:e5:e8:4f:63:
                    09:15:5f:8e:ec:df:86:fb:35:47:99:db:18:d1:b7:
                    40:4e:b6:b3:b6:66:28:8e:89:15:8b:cc:0f:e6:5c:
                    e6:2d:de:83:6c:f8:e3:46:49:97:a6:a9:0e:6d:b1:
                    65:08:8e:aa:fc:f0:ae:2f:c1:c2:cd:b6:4f:a5:eb:
                    29:32:9a:48:8c:86:6d:1e:3a:c2:22:70:a3:56:e9:
                    17:01:ef:3a:ce:bb:9f:04:47:e5:24:e0:16:ba:c0:
                    85:df:92:4d:51:d2:95:bf:84:f7:9a:2e:c0:31:e9:
                    9f:91:4f:4a:ce:2c:27:17:f8:ae:3e:96:4e:3b:0a:
                    15:1a:66:cf:e9:12:96:e1:17:ee:65:3c:04:7a:c0:
                    a0:b3:09:fd:3e:16:08:c6:0b:36:51:57:cb:d8:09:
                    a3:40:d0:2c:ae:d6:06:e0:8c:06:de:b7:ce:24:83:
                    28:69
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                DNS:CMS.UC.local, DNS:CMS.UC.local, DNS:CMS1.UC.local, DNS:CMS2.UC.local,
DNS:CMS3.UC.local
            X509v3 Subject Key Identifier:
                FE:EF:64:D6:85:7A:62:C5:CA:7B:64:10:B7:F9:E7:18:1D:65:0B:70
            X509v3 Authority Key Identifier:
                keyid:B5:FC:2D:1E:7F:D9:3E:68:F4:B2:78:1F:F0:E8:B2:FC:80:7F:9C:E8

            X509v3 CRL Distribution Points:

                Full Name:
                  URI:ldap:///CN=DC-
CA,CN=DC,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=uc,DC=local?certifica
teRevocationList?base?objectClass=cRLDistributionPoint

            Authority Information Access:
                CA Issuers - URI:ldap:///CN=DC-
CA,CN=AIA,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=uc,DC=local?cACertificate?b
```

```
ase?objectClass=certificationAuthority

            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            1.3.6.1.4.1.311.21.7:
                0..&+.....7.....\...........A........N...O..d...
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            1.3.6.1.4.1.311.21.10:
                0.0
..+.......0
..+.......
    Signature Algorithm: sha256WithRSAEncryption
         83:31:16:15:74:41:98:e4:40:02:70:cc:6e:c0:53:15:8a:7a:
         8a:87:0a:aa:c8:99:ff:5b:23:e4:8b:ce:dd:c0:61:9c:06:b4:
         3d:22:91:b6:91:54:3a:99:8d:6e:db:18:27:ef:f7:5e:60:e6:
         48:a2:dd:d5:85:1d:85:55:79:e0:64:1a:55:22:9e:39:0c:27:
         53:a4:d8:3f:54:fd:bc:f9:d4:6e:e1:dd:91:49:05:3e:65:59:
         6e:d4:cd:f6:de:90:cb:3d:b3:15:03:4b:b8:9d:41:f1:78:f5:
         d9:42:33:62:b5:18:4f:47:54:c9:fa:58:4b:88:aa:0d:f6:26:
         9b:fb:8f:98:b4:82:96:97:24:fe:02:5b:03:04:67:c2:9e:63:
         3d:02:ae:ef:92:a7:be:ad:ca:7e:4e:d2:1e:54:e6:bf:75:3b:
         72:32:7c:d6:78:3f:5e:b9:e6:43:bd:1c:74:20:46:57:1b:81:
         c2:4b:b4:fc:9f:cc:c9:63:a8:2d:fd:dd:09:3f:24:d6:ac:f7:
         7c:bd:26:80:a5:b4:d1:a7:c8:fb:3d:d4:a7:93:70:d1:5c:77:
         06:9e:1c:f8:6a:81:a5:97:91:e9:21:e9:7a:df:a3:64:ab:ed:
         15:c7:be:89:5f:1e:53:a7:b5:01:55:ab:a2:cd:8f:67:8d:14:
         83:bc:29:a1
cms1>
```

Please take note of the Subject and X509v3 Subject Alternative Name fields. These will be extremely important later when we build our trust relationships in the Microsoft environment.

```
        Subject: C=US, ST=NC, L=RTP, O=Systems, OU=Cisco, CN=CMS.UC.local

            X509v3 Subject Alternative Name:
                DNS:CMS.UC.local, DNS:CMS.UC.local, DNS:CMS1.UC.local, DNS:CMS2.UC.local,
DNS:CMS3.UC.local
```

Note: The Certificate Configuration guide can be found here: https://www.cisco.com/c/dam/en/us/td/docs/conferencing/ciscoMeetingServer/Deployment_Guide/Version-2-2/Certificate-Guidelines-Single-Split_Server-Deployment-2-2.pdf

DNS Record Requirements - Single CallBridge

Table 2a provides an example of how to configure the DNS server. It provides an explanation of what does each field means.

Table 2a

A Record	IP Example	Description
cms.uc.local	10.10.10.1	CallBridge
fe.skype.local	10.10.10.5	Skype Front End Fully Qualified Domain Name (FQDN)

DNS Record Requirements - Clustered CallBridges

Table 2b provides an example of how to configure the DNS server. It provides an explanation of what does each field means.

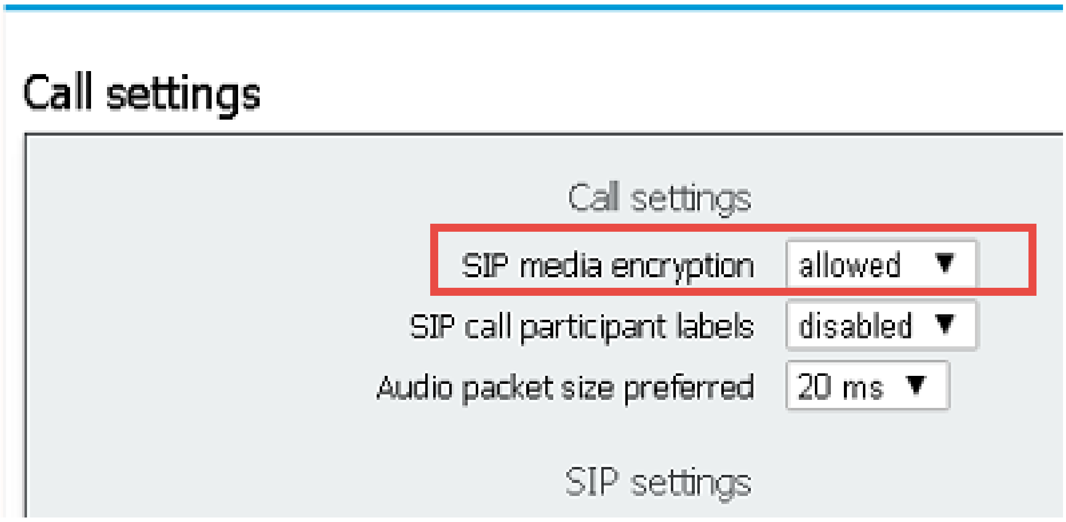
Table 2b

A Record	IP Example	Description
cms1.uc.local	10.10.10.1	CallBridge 1
cms2.uc.local	10.10.10.2	CallBridge 2
cms3.uc.local	10.10.10.3	CallBridge 3
cms.uc.local	10.10.10.1 10.10.10.2 10.10.10.3	An A record that resolves to all CallBridges in the cluster. This will be referred to the CallBridge Cluster Fully Qualified Domain Name (FQDN)
fe.skype.local	10.10.10.5	Skype Front End Fully Qualified Domain Name (FQDN)

Configuration

SIP Media Encryption

Navigate to **Configuration> Call Settings.** SIP media encrpytion must be set to allowed.



Call settings

Call settings

SIP media encryption: allowed ▼

SIP call participant labels: disabled ▼

Audio packet size preferred: 20 ms ▼

SIP settings

Inbound Rules

Table 3 describes what every field in the Incoming Calls - Call Matching configuration means.

Table 3

Incoming Call Matching Dial Plan Field	Description
Domain Name	If a call is received with this domain then use the user portion of the URI to look for matches in the enabled targets.
Priority	This determines the order in which the rules will be considered. Higher numbers wi checked first. Lower numbers will be checked last.
Targets Spaces	If set to yes: if the user portion of the URI matches a space the call will connect to t space.
Targets Users	If set to yes: if the user portion of the URI matches a CMA user the call will attempt call that user.
Targets IVR	If set to yes: if the user portion of the URI matches a configured IVR the call will co to that IVR.
Targets Lync	If set to yes: If the user portion of the URI matches a PSTN Dialin Number of a Sky for Business Meeting connect to that Meeting as a Dual Homed call.
Targets Lync	If set to yes: Convert the user portion of the URI into an HTTPS target and try to fir

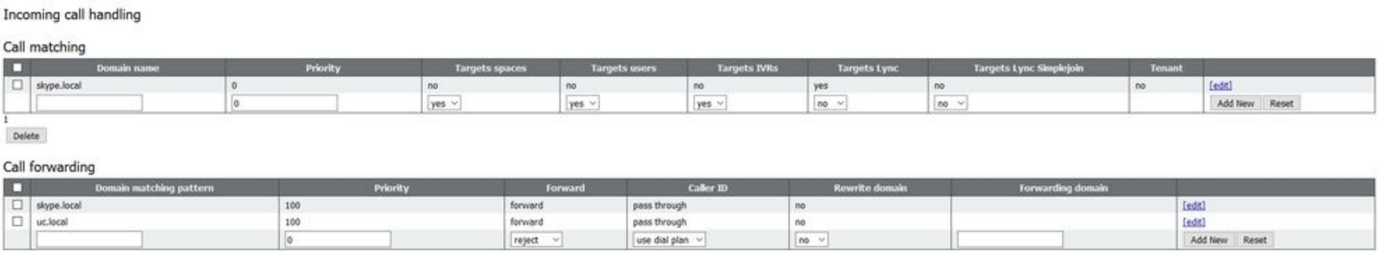
Simplejoin	Office365 meeting hosted at that URL.
Tenant	This determines which tenants this rule will be considered for.

Table 4 describes what every field in the Incoming Calls - Call Forwarding configuration means.

Table 4

Incoming Call Forwarding Dial Plan Field	Desciption
Domain Matching Pattern	If a call is received with this domain then forward or reject the domain as configure
Priority	This determines the order in which the rules will be considered. Higher numbers w checked first. Lower numbers will be checked last.
Forward	If set to forward the call will be handled by the outbound rules. If set to reject the ca be rejected and not forwarded.
Caller ID	If set to pass through the from portion of the domain will be preseved. If set to use plan the from portion will be rewritten as configured in the outbound rule. Note: Pass through cannot be used for rules that match a Lync/Skype domain if th CallBridge is in a cluster. This would break presentation on gateway calls.
Rewrite Domain	If enabled change the called domain to the value configured in the forwarding dom field.
Forwarding Domain	If rewrite domain is enabled the called domain will change to the value of this field.

Example Inbound Rules configuration - Single CallBridge



Incoming call handling

Call matching

■	Domain name	Priority	Targets spaces	Targets users	Targets IVRs	Targets Lync	Targets Lync Simplejoin	Tenant	
□	skype.local	0	no	no	no	yes	no	no	[edit]
		0	yes	yes	yes	no	no		Add New Reset

1

Delete

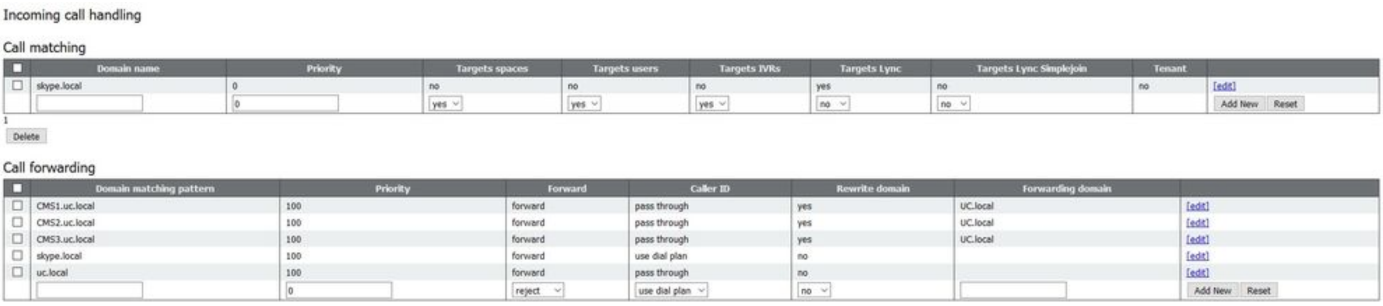
Call forwarding

■	Domain matching pattern	Priority	Forward	Caller ID	Rewrite domain	Forwarding domain	
□	skype.local	100	forward	pass through	no		[edit]
□	uc.local	100	forward	pass through	no		[edit]
		0	reject	use dial plan	no		Add New Reset

In this environment things are remarkably simple. Since we are not using clustered CallBridges we can set each domain to use pass through as their Caller ID. This cannot be done in a clustered environment as it will break presentation sharing.

Additionally there is a call matching rule for the domain Skype.local with "Targets Lync" set to true. This means if we call a Lync/Skype meeting by the PSTN dialin number, we should be able to connect as a Dual Home call.

Example Inbound Rules configuration - Clustered CallBridges



Incoming call handling

Call matching

■	Domain name	Priority	Targets spaces	Targets users	Targets IVRs	Targets Lync	Targets Lync Simplejoin	Tenant	
□	skype.local	0	no	no	no	yes	no	no	[edit]
		0	yes	yes	yes	no	no		Add New Reset

1

Delete

Call forwarding

■	Domain matching pattern	Priority	Forward	Caller ID	Rewrite domain	Forwarding domain	
□	CMS1.uc.local	100	forward	pass through	yes	UC.local	[edit]
□	CMS2.uc.local	100	forward	pass through	yes	UC.local	[edit]
□	CMS3.uc.local	100	forward	pass through	yes	UC.local	[edit]
□	skype.local	100	forward	use dial plan	no		[edit]
□	uc.local	100	forward	pass through	no		[edit]
		0	reject	use dial plan	no		Add New Reset

In this environment we are using a CallBridge cluster that consists of three CallBridges. Because of this we need one call forwarding rule for each CallBridge configured to rewrite the domain to uc.local. This is because when Lync/Skype users callback users from the UC environment they will actually be placing calls to the domain of cms1.uc.local, cms2.uc.local, or cms3.uc.local. Unfortunately this is a limitation of the configuration that is required to have content working in a clustered CallBridge environment. We need to convert this back to uc.local before forwarding the call to the uc.local sip proxy.

Additionally there is a call matching rule for the domain Skype.local with "Targets Lync" set to true. This means if we call a Lync/Skype meeting by the PSTN dialin number, we should be able to connect as a Dual Home call.

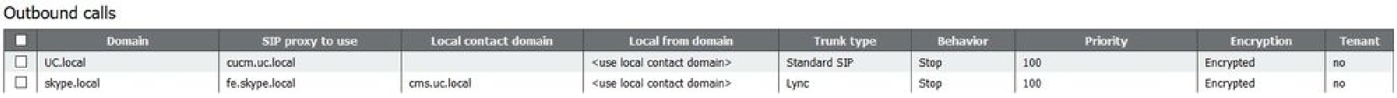
Outbound Rules

Table 5 describes what every field in the outbound calls configuration means.

Table 5

Outbound Dial Plan Field	Description
Domain	For calls out to this domain use this outbound rule
SIP proxy to use	The SIP proxy to send calls to for this domain
Local contact domain	This determines what value will be put in the contact header. For Lync/Skype integration this value must be set to the FQDN of the CallBridge. Note: For any outbound rules using a SIP proxy of Lync/Skype this field MUST be configured. For any outbound rules using a SIP proxy that is not Lync/Skype this field MUST NOT be configured.
Local from domain	This determines what value will be put in the from header. This will be the caller-ID address seen on the SIP proxy. If left blank this field will use the "Local contact domain" configured. Lync/Skype will use this as the destination URI for callbacks and presentation sharing. Note: This value is not used if the call is a gateway call and the inbound dial rule used has "Caller ID" set to passthrough.
Trunk type	This determines what variation of SIP will be used in communication with the SIP proxy.
Behavior	This determines whether or not we will continue checking lower priority rules or stop searching in the event of a match where we were unable to complete the call.
Priority	This determines the order in which the rules will be considered. Higher numbers will be checked first. Lower numbers will be checked last.
Encryption	This determines whether we will use encrypted or unencrypted SIP.
Tenant	This determines which tenants this rule will be considered for.
Call Bridge Scope	This determines which CallBridges this outbound dial rule will be considered for. In clustered CallBridges this is required to ensure the correct contact domain is sent from each CallBridge. Note: This value can only be set utilizing the API as explained below.

Example Outbound Calls configuration - Single CallBridge

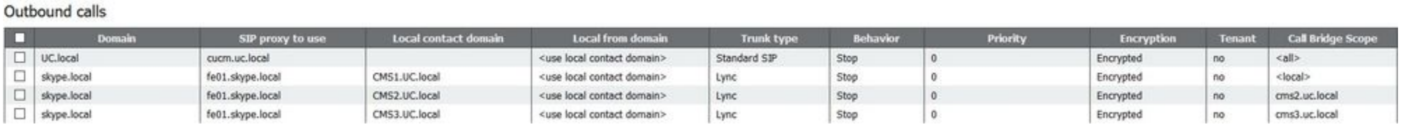


Outbound calls

■	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant
□	UC.local	cucm.uc.local		<use local contact domain>	Standard SIP	Stop	100	Encrypted	no
□	skype.local	fe.skype.local	cms.uc.local	<use local contact domain>	Lync	Stop	100	Encrypted	no

Again we see that the single CallBridge environment is considerably simpler than the clustered environment. One thing worth note above is that we have a contact domain specified. This is because if we do not specify the Fully Qualified Domain Name of our CallBridge as the local contact domain Lync/Skype will reject calls for security reasons. Since our incoming forwarding rules are set to use pass through, we will not actually be rewriting the from domain in this example.

Example Outbound Calls configuration - Clustered CallBridges



Outbound calls

■	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	Call Bridge Scope
□	UC.local	cucm.uc.local		<use local contact domain>	Standard SIP	Stop	0	Encrypted	no	<all>
□	skype.local	fe01.skype.local	CMS1.UC.local	<use local contact domain>	Lync	Stop	0	Encrypted	no	<local>
□	skype.local	fe01.skype.local	CMS2.UC.local	<use local contact domain>	Lync	Stop	0	Encrypted	no	cms2.uc.local
□	skype.local	fe01.skype.local	CMS3.UC.local	<use local contact domain>	Lync	Stop	0	Encrypted	no	cms3.uc.local

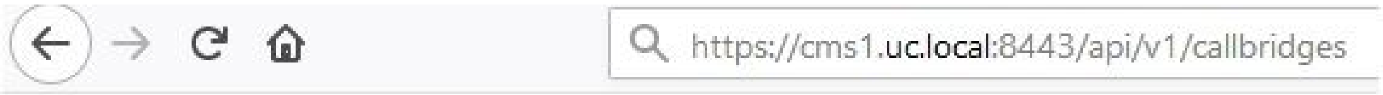
In this environment we are using a CallBridge cluster that consists of three CallBridges. Because of this we need one outbound rule for each CallBridge each with different local contact domains, local from domains, and scopes. Only one outbound rule is needed to route the calls from all CallBridges to the Cisco Unified Communications Manager. To set the scope we need to utilize the API.

Modifying Scope Utilizing the API - Clustered CallBridges only

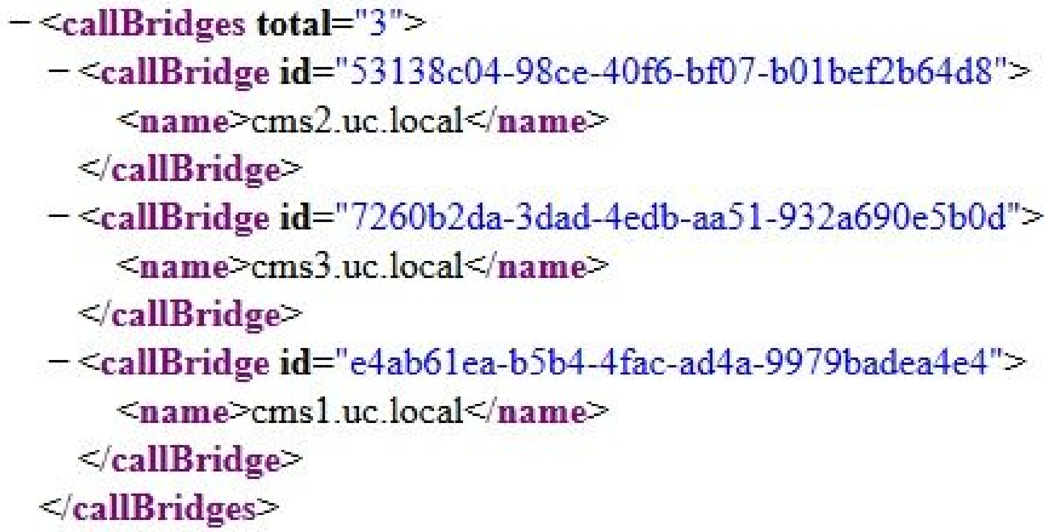
After creating an outbound call rule the scope will be set to <all> for that rule. This means that outbound rule will be used on all CallBridges in a cluster. For outbound rules that point towards Lync/Skype we need to use different contact and from headers depending on which CallBridge we are on. In order to do this we need to create a different outbound rule for each CallBridge where the contact/from fields match that CallBridge. Using the API we need to set the scope of these outbound dial rules so that they are only processed on the CallBridge that matches that rule.

GET a list of all CallBridges in the cluster

In a browser navigate to the /callbridges page of the CMS API. This will show all of the CallBridges in your cluster.



https://cms1.uc.local:8443/api/v1/callbridges



```
– <callBridges total="3">
  – <callBridge id="53138c04-98ce-40f6-bf07-b01bef2b64d8">
      <name>cms2.uc.local</name>
    </callBridge>
  – <callBridge id="7260b2da-3dad-4edb-aa51-932a690e5b0d">
      <name>cms3.uc.local</name>
    </callBridge>
  – <callBridge id="e4ab61ea-b5b4-4fac-ad4a-9979badea4e4">
      <name>cms1.uc.local</name>
    </callBridge>
  </callBridges>
```

Now I have the IDs for all of my CallBridges. Your IDs will be different in your environment. I can see that if I want to reference CallBridge cms1.uc.local I should use the ID of e4ab61ea-b5b4-4fac-ad4a-9979badea4e4.

GET a list of all outbound dial rules

Next, I need to look up my outbound rules and get their IDs. In a browser navigate to the /outbounddialplanrules page in the API.

```
<outboundDialPlanRules total="4">
  <outboundDialPlanRule id="7c76b6c7-4c42-45b0-af47-796cb6737e4e">
    <domain>UC.local</domain>
    <priority>0</priority>
  </outboundDialPlanRule>
  <outboundDialPlanRule id="b8cf4056-7f56-43a5-b67b-861253d5ca32">
    <domain>skype.local</domain>
    <priority>0</priority>
  </outboundDialPlanRule>
  <outboundDialPlanRule id="4ae1d777-48b7-423b-a646-a329e1e822af">
    <domain>skype.local</domain>
    <priority>0</priority>
  </outboundDialPlanRule>
  <outboundDialPlanRule id="05f00293-50fd-4c17-9452-dec224b43430">
    <domain>skype.local</domain>
    <priority>0</priority>
  </outboundDialPlanRule>
</outboundDialPlanRules>
```

Now I have the IDs for all of my rules, but I can't tell which is which. We don't care about the first rule since that one is to UC.local and we don't need to set a scope for that. We do need to know which rule is which for the remaining outbound rules to Skype.local. So starting one at a time I will match the IDs to the CallBridges.

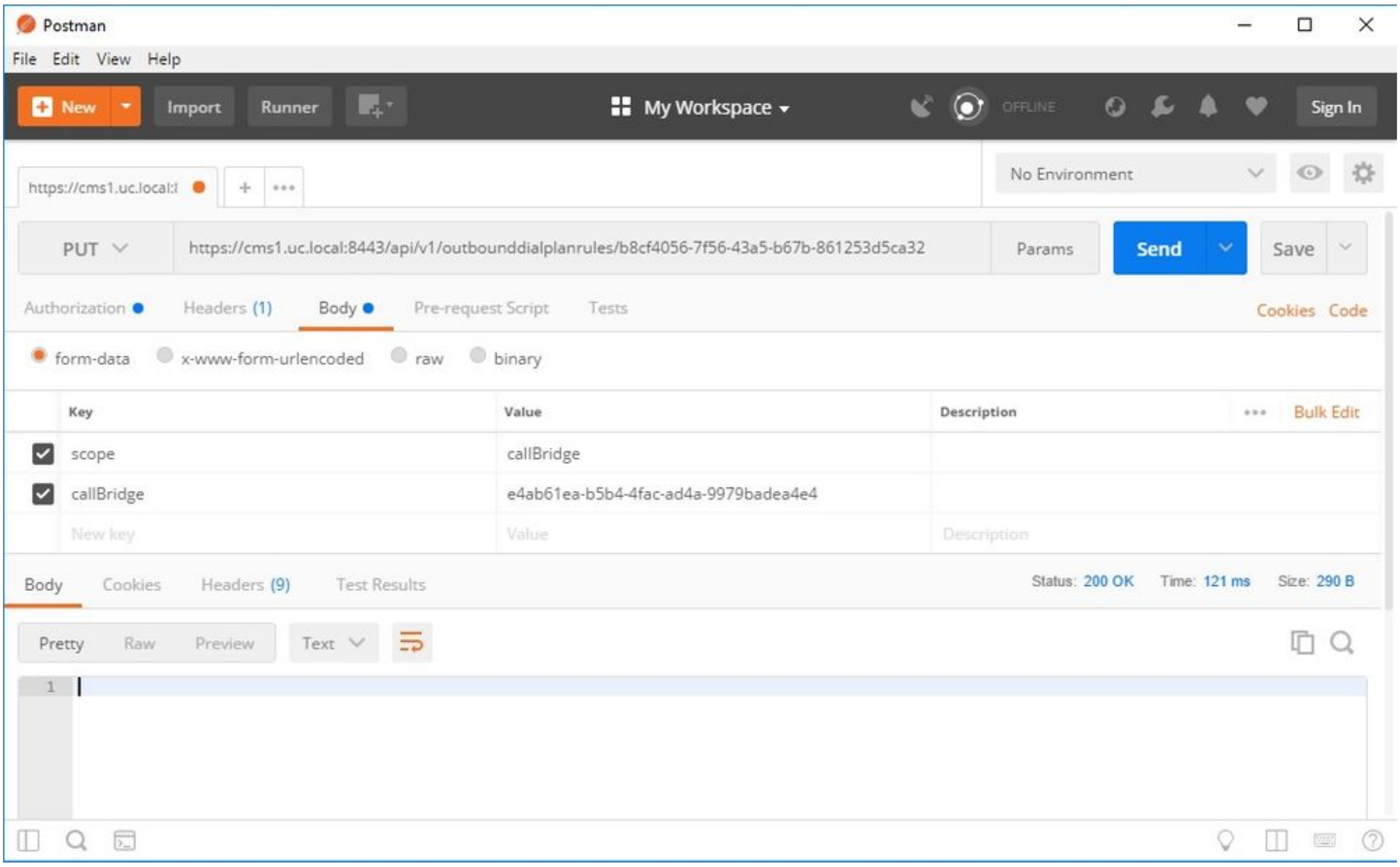
I will navigate to /outbounddialplanrules/b8cf4056-7f56-43a5-b67b-861253d5ca32 in my browser. Reading the contact header listed there I can tell this rule is for CMS1.UC.local. So we need to set the scope of this rule to CMS1.UC.local.

PUT the CallBridge Scope in

Using my favorite API tool I will send a PUT to the api on /outbounddialplanrules/b8cf4056-7f56-43a5-b67b-861253d5ca32 with the following body:

```
scope: callBridge
callBridge: e4ab61ea-b5b4-4fac-ad4a-9979badea4e4
```

In this screenshot I am using PostMan to send this request.



PUT https://cms1.uc.local:8443/api/v1/outbounddialplanrules/b8cf4056-7f56-43a5-b67b-861253d5ca32

Body: ● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	Key	Value	Description
☑	scope	callBridge	
☑	callBridge	e4ab61ea-b5b4-4fac-ad4a-9979badea4e4	
	New key	Value	Description

Status: 200 OK Time: 121 ms Size: 290 B

If this HTTP PUT was successful the outbound dial rules page in WebAdmin should now reflect a scope has been applied. If viewed from the Webadmin of the CallBridge that the scope was applied to it should show <local>. If the Webadmin of another CallBridge is used to view the outbound dial rules it should show the CallBridge FQDN in the scope field. A scope of <all> means the rule will be used on all CallBridges. A scope of <none> means that a scope has been enabled, but no CallBridges match the scope.

After setting the scope for one CallBridge it needs to be configured for each additional CallBridge. After this configuration has been completed every outbound rule for your Skype domain should have a scope.

CMS Service Accounts

In the general configuration page of the WebAdmin there is a Lync Edge settings section. In order to utilize TURN services or join Dual Home meetings via the PSTN Dialin number this must be configured.

Table 6 describes what every field in the Lync Edge settings configuration means.

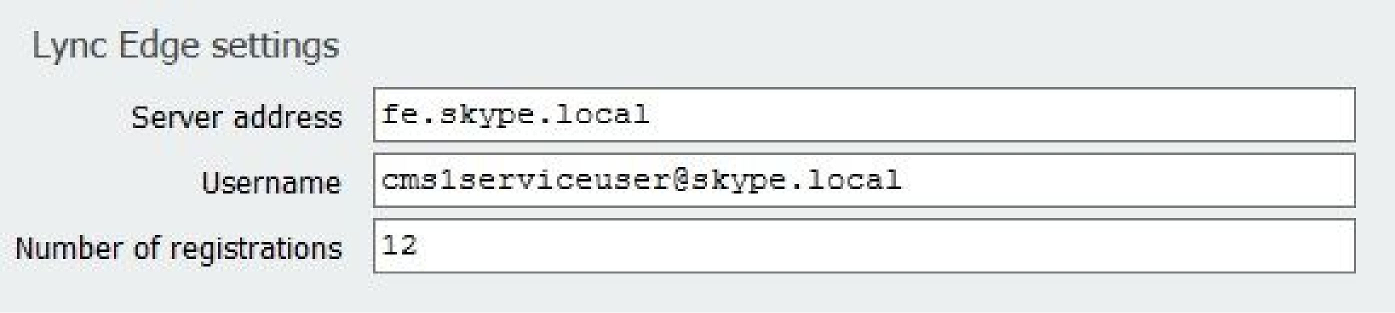
Table 6

Lync Edge settings field	Description
Server Address	Fully Qualified Domain Name (FQDN) of your Front End Pool
Username	The username of the service account you want to use for CMS
Number of Registrations	How many different user accounts you would like to register. If a value is not configured her then only the username as listed above will be registered. If a number is applied here the numbers 1-X will be applied as suffixes to the user portion of the URI where X is the numbe

configured in this field.

Example CMS Service Account Configuration

Configuration on CMS1:



Lync Edge settings

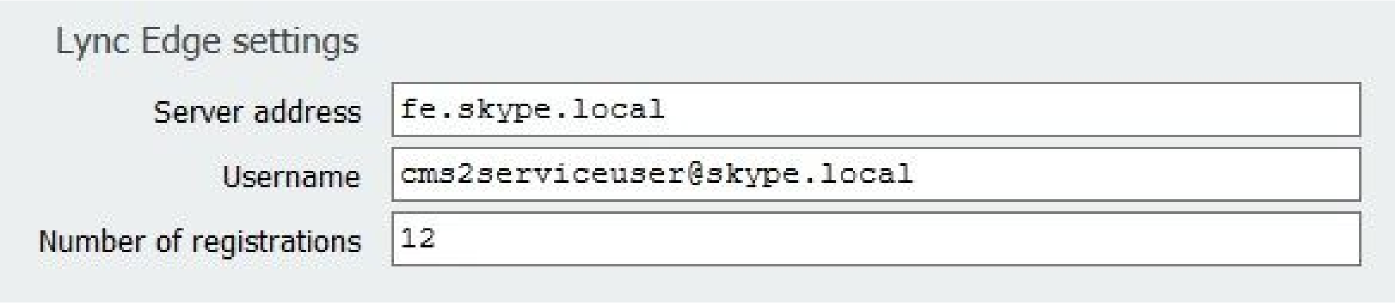
Server address: fe.skype.local

Username: cms1serviceuser@skype.local

Number of registrations: 12

This configuration would register cms1serviceuser1@skype.local, cms1serviceuser2@skype.local, cms1serviceuser3@skype.local, ... cms1serviceuser11@skype.local, and cms1serviceuser12@skype.local to fe.skype.local. Since in this example I'm in a clustered environment I would need to also create service accounts for my other CallBridges and configure them separately. Please note that the usernames in this example are different. On CMS1 the usernames are prefixed with cms1. On CMS2 the usernames are prefixed with cms2. On CMS3 the prefix is cms3. All of these accounts were made and enabled in the Skype for Business environment. Since our Trusted Application Pool is configured with "Treat as authenticated" we do not need to supply passwords to register.

Configuration on CMS2:



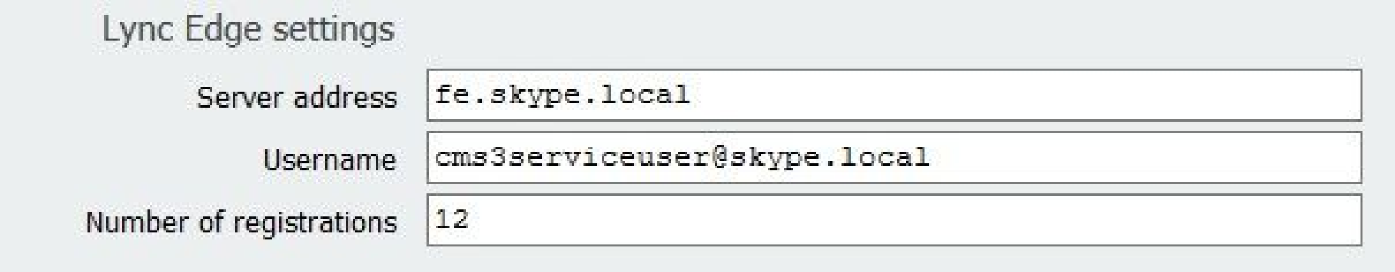
Lync Edge settings

Server address: fe.skype.local

Username: cms2serviceuser@skype.local

Number of registrations: 12

Configuration on CMS3:



Lync Edge settings

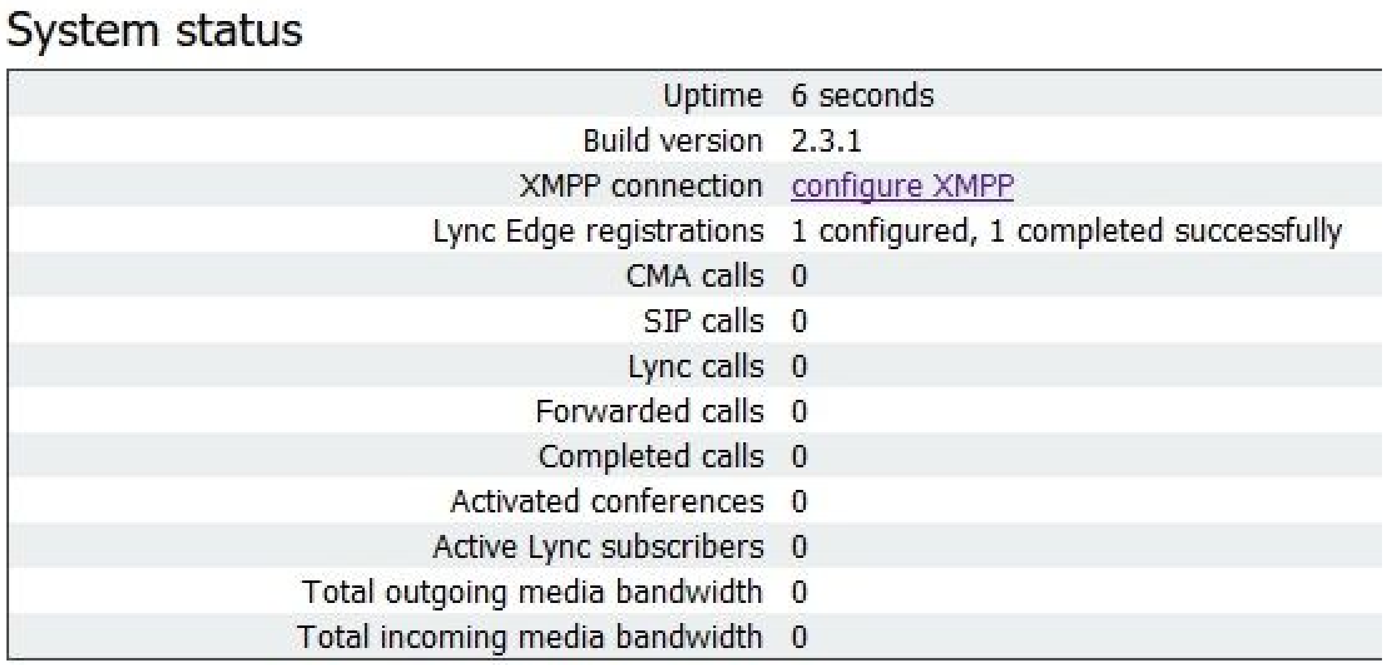
Server address: fe.skype.local

Username: cms3serviceuser@skype.local

Number of registrations: 12

Verifying CMS Service Accounts

The status page of the CMS WebAdmin will show if the Lync/Skype users have successfully registered. In the below example we are only configuring one registration and it has completed successfully. If you notice that the status shows registrations in progress for a long time collect SIP and DNS logs to determine why the failure is occurring.



System status

Uptime	6 seconds
Build version	2.3.1
XMPP connection	configure XMPP
Lync Edge registrations	1 configured, 1 completed successfully
CMA calls	0
SIP calls	0
Lync calls	0
Forwarded calls	0
Completed calls	0
Activated conferences	0
Active Lync subscribers	0
Total outgoing media bandwidth	0
Total incoming media bandwidth	0

Lync/Skype Configuration

Apply the below commands in the Lync/Skype Management Shell. Apply the commands on the Front End server.

> **Note**: The suggested commands are for guidence. In case you have doubts about the configuration on Skype server, you will need to contact your Lync/Skype administrator and/or support team.

Single CallBridge

First, we need to tell Skype to trust our CallBridge. To do this we add a Trusted Application Pool. In Microsoft terminology "Pool" just means "Cluster." In this scenario our cluster is just a cluster of one CallBridge. The Identity of our cluster MUST match the common name of the certificate in use on our CallBridge. Microsoft uses this as a security check. Having the Identity in a SAN is not enough. If the common name does not match Microsoft will tear down the TCP connection. When using this command the identity should be the CallBridge FQDN. The Registrar whould be the FQDN of the Front End Pool servicing these connections. The site should be the Lync/Skype site identifier. If you are unsure of the values that should be used for registrar or site please contact your Lync/Skype administrator.

```
New-CsTrustedApplicationPool -Identity CMS.UC.local -Registrar fe.skype.local -site 1 -RequiresReplication $false -
ThrottleAsServer $true -TreatAsAuthenticated $true
```

Next the Microsoft Environment must be configured to allow inbound communication from our CallBridge (Trusted Application Pool) on port 5061.

```
New-CsTrustedApplication -ApplicationId AcanoApplication -TrustedApplicationPoolFqdn CMS.UC.local -Port 5061
```

The Microsoft environment is currently configured to accept calls, but it cannot place back calls and cannot send presentation for gateway calls. To correct this we need to add a static route. In the single CallBridge scenario we only need a single route to allow all calls to our UC.local domain. In the below commands Destination is the FQDN of the CallBridge we want to send SIP requests to. The MatchURI field is the domain portion of the URI that should be used. Please note that in a Lync/Skype environment only one static route can be created per MatchURI.

```
$x1=New-CsStaticRoute -TLSRoute -Destination “CMS.UC.local" -MatchUri “UC.local" -Port 5061 -UseDefaultCertificate
$true Set-CsStaticRoutingConfiguration -Identity global -Route @{Add=$x1}
```

Finally, we need to tell Skype to implement all the changes we've just made.

```
Enable-CsTopology
```

Clustered CallBridges

First, we need to tell Skype to trust our CallBridge cluster. To do this we add a Trusted Application Pool. In Microsoft terminology "Pool" just means "Cluster."

The Identity of our cluster MUST match the common name of the certificate(s) in use on our CallBridge(s). Microsoft uses this as a security check. Having the Identity in a SAN is not enough. If the common name does not match Microsoft will tear down the TCP connection. When using this command the identity should be the CallBridge FQDN. ComputerFqdn should be the FQDN of the first CallBridge in your cluster. By specifying a ComputerFqdn you are indicating to the Lync/Skype environment that this is not a cluster with only a single server in it. The Registrar whould be the FQDN of the Front End Pool servicing these connections. The site should be the Lync/Skype site identifier. If you are unsure of the values that should be used for registrar or site please contact your Lync/Skype administrator.

```
New-CsTrustedApplicationPool -Identity CMS.UC.local -ComputerFqdn CMS1.UC.local -Registrar fe.skype.local -site 1 -
RequiresReplication $false -ThrottleAsServer $true -TreatAsAuthenticated $true
```

In this environment we need to add two CallBridges as Trusted Application Computers. The first CallBridge was already added when we created the Trusted Application Pool Above. When we add these computers we need to associate them with the pool we've just created. This tells Skype that we have additional Computers in our cluster that need to be trusted. All of the computer Identities here need to be listed as SAN's in our CallBridge certificate(s). These identities must also match the contact headers in the outbound dial rules in the CallBridges. If they do not match Microsoft will tear down the TCP connection.

```
New-CsTrustedApplicationComputer -Identity CMS2.UC.local -Pool CMS.UC.local New-CsTrustedApplicationComputer -
Identity CMS3.UC.local -Pool CMS.UC.local
```

Next the Microsoft Environment must be configured to allow inbound communication from our CallBridge cluster (Trusted Application Pool) on port 5061.

```
New-CsTrustedApplication -ApplicationId AcanoApplication -TrustedApplicationPoolFqdn CMS.UC.local -Port 5061
```

The Microsoft environment is currently configured to accept calls, but it cannot place back calls and cannot send presentation for gateway calls. To correct this we need to add static routes. First we need to add a static route to allow all calls to our UC.local domain. In the below commands Destination is the FQDN of the CallBridge we want to send SIP requests to. The MatchURI field is the domain portion of the URI that should be used. Please note that in a Lync/Skype environment only one static route can be created per MatchURI. Since the Destination is the FQDN of our CallBridge cluster and it has a DNS A record for every member of the cluster Lync/Skype can send traffic to all of our CallBridges. So if one goes down it can automatically route requests for our domain to another CallBridge in the cluster.

```
$x1=New-CsStaticRoute -TLSRoute -Destination "CMS.UC.local" -MatchUri "UC.local" -Port 5061 -UseDefaultCertificate
$true Set-CsStaticRoutingConfiguration -Identity global -Route @{Add=$x1}
```

Next, we need to create an additional static route for each CallBridge in the cluster. This is a requirement for callback and presentation to work.

```
$x2=New-CsStaticRoute -TLSRoute -Destination "CMS1.UC.local" -MatchUri "CMS1.UC.local" -Port 5061 -
UseDefaultCertificate $true Set-CsStaticRoutingConfiguration -Identity global -Route @{Add=$x2} $x3=New-
CsStaticRoute -TLSRoute -Destination "CMS2.UC.local" -MatchUri "CMS2.UC.local" -Port 5061 -UseDefaultCertificate
$true Set-CsStaticRoutingConfiguration -Identity global -Route @{Add=$x3} $x4=New-CsStaticRoute -TLSRoute -
Destination "CMS3.UC.local" -MatchUri "CMS3.UC.local" -Port 5061 -UseDefaultCertificate $true Set-
CsStaticRoutingConfiguration -Identity global -Route @{Add=$x4}
```

Finally, we need to tell Skype to implement all the changes we've just made.

```
Enable-CsTopology
```

Troubleshooting

Collecting logs from CMS

The first step in diagnosing any issue is determining where the issue is. To do this we need to analyze the logs from the Cisco Meeting Server, but first we need to collect them. Here are my personal recommendations on logs to collect.

First, enable SIP and DNS debugging for all CallBridges via the WebAdmin interface. To do this navigate to the WebAdmin and then to Logs > Detailed Tracing. From here enable SIP and DNS logging for the next thirty minutes. This should be more than enough time to catch and diagnose the issue. Keep in mind this needs to be done individually for all CallBridges as log enablement is not shared across a cluster.

Second, enable packet captures on all CallBridges. To do this connect via SSH to each CallBridge and run the command pcap <interface> where <interface> is the interface traffic should use. In most cases this will be interface a. So the command "pcap a" would start a packet capture on interface a for the CallBridge we are connected to.

Once the packet capture is running on all interfaces the next step is to produce the problem. Go ahead and attempt a call or do whatever it was that was failing. After this is completed terminate all of the packet captures. This can be done by entering Ctrl-C in all of the SSH windows. Once the packet capture is completed the name of the file generated will be written to the screen. Keep track of this filename as we will need to download it in the next step.

Finally we need to collect the logs from the CallBridges. To do this connect via SFTP to each CallBridge. Download the file logbundle.tar.gz and the packet capture file generated. This file is only available in CMS2.2+. In CMS versions 2.3+ it will include the full configuration of your CMS. If you are running version

2.2 it will not include your inbound/outbound rules, so it would be good to take screenshots of those pages as well as the Lync Edge settings for reference. Make sure to store the logs/screenshots collected in separate folders that has a name matching the CallBridge the logs were pulled from. This will help make sure the logs don't get mixed up.

Viewing Lync/Skype Configuration

These commands will come in extremely helpful when troubleshooting the Lync/Skype configuration. In this document commands are given to create and view configuration, but no commands are given to remove configuration. This is because removing configuration can be dangerous unless performed by administrators with a full understanding of the Lync/Skype environment. If you need to remove configuration please work with your Lync/Skype admin to do so.

Command	Description
Get-CsTrustedApplicationPool	This command lists clusters (pools) trusted by Lync/Skype. The identity of this pool MUST match the common name of the CallBridge certificate(s). Even in a single CallBridge environment a CallBridge cluster (pool) of one must be specified here.
Get-CsTrustedApplicationComputer	This command lists servers trusted by Lync/Skype and which Pool these servers are associated with. All computers here MUST be identified in the certificate sent by the CallBridges. In a single CallBridge environment this is typically the common name. In a clustered environment these computers MUST be listed as Subject Alternative Name (SAN) entries. Additionally, all computers here MUST be identified by local contact domain entries on the CallBridge outbound dial rules.
Get-CsTrustedApplication	This command lists which services trusted application pools are allowed to communicate with. For CMS communication with Lync/Skype we will use TCP port 5061 for TLS encrypted SIP.
Get-CsStaticRoutingConfiguration | Select-Object -ExpandProperty Route	This command lists the static routes that Lync/Skype uses for forwarding requests. The MatchURI field is the destination domain of the SIP message. The "TLS Fqdn" field in the XML should show the destination server for this traffic.

Example output of Lync/Skype Get commands

Below is the output of the above Lync/Skype Get commands issued in the three CallBridge cluster scenario covered in this document

```
PS C:\Users\administrator.SKYPE> Get-CsTrustedApplicationPool


Identity              : TrustedApplicationPool:CMS.UC.local
Registrar             : Registrar:lyncpoolfe01.skype.local
FileStore             :
ThrottleAsServer      : True
TreatAsAuthenticated  : True
OutboundOnly          : False
RequiresReplication   : False
AudioPortStart        :
AudioPortCount        : 0
AppSharingPortStart   :
AppSharingPortCount   : 0
VideoPortStart        :
VideoPortCount        : 0
Applications          : {urn:application:acanoapplication}
DependentServiceList  : {}
ServiceId             : 1-ExternalServer-1
SiteId                : Site:RTP
```

```
PoolFqdn             : CMS.UC.local
Version              : 7
Role                 : TrustedApplicationPool



PS C:\Users\administrator.SKYPE> Get-CsTrustedApplicationComputer


Identity : CMS1.UC.local
Pool     : CMS.UC.local
Fqdn     : CMS1.UC.local

Identity : CMS2.UC.local
Pool     : CMS.UC.local
Fqdn     : CMS2.UC.local

Identity : CMS3.UC.local
Pool     : CMS.UC.local
Fqdn     : CMS3.UC.local




PS C:\Users\administrator.SKYPE> Get-CsTrustedApplication


Identity                    : CMS.UC.local/urn:application:acanoapplication
ComputerGruus               : {CMS1.UC.local
sip:CMS1.UC.local@skype.local;gruu;opaque=srvr:acanoapplication:GMqDXW_1rVCEMQi4qS6ZxwAA,
CMS2.UC.local

sip:CMS2.UC.local@skype.local;gruu;opaque=srvr:acanoapplication:_Z9CnV49LFufGDXjnFFi4gAA,
CMS3.UC.local
sip:CMS3.UC.local@skype.local;gruu;opaque=srvr:acanoapplication:dt8XJKciSlGhEeT62tyNogAA}
ServiceGruu                 :
sip:CMS.UC.local@skype.local;gruu;opaque=srvr:acanoapplication:dQFM4E4YgV6J0rjuNgqxIgAA
Protocol                    : Mtls
ApplicationId               : urn:application:acanoapplication
TrustedApplicationPoolFqdn : CMS.UC.local
Port                        : 5061
LegacyApplicationName       : acanoapplication




PS C:\Users\administrator.SKYPE> Get-CsStaticRoutingConfiguration | Select-Object -
ExpandProperty Route


Transport                : 
TransportChoice=Certificate=Microsoft.Rtc.Management.WritableConfig.Settings.SipProxy.UseDefault
Cert;Fqdn=CMS.UC.local;Port=5061
MatchUri                 : UC.local
MatchOnlyPhoneUri        : False
Enabled                  : True
ReplaceHostInRequestUri : False
Element                  : <Route
xmlns="urn:schema:Microsoft.Rtc.Management.Settings.SipProxy.2008" MatchUri="UC.local"
MatchOnlyPhoneUri="false" Enabled="true" ReplaceHostInRequestUri="false">
                            <Transport Port="5061">
                              <TLS Fqdn="CMS.UC.local">
                                <UseDefaultCert />
                              </TLS>
                            </Transport>
                          </Route>
```

```
Transport               :
TransportChoice=Certificate=Microsoft.Rtc.Management.WritableConfig.Settings.SipProxy.UseDefault
Cert;Fqdn=CMS1.UC.local;Port=5061
MatchUri                : CMS1.UC.local
MatchOnlyPhoneUri       : False
Enabled                 : True
ReplaceHostInRequestUri : False
Element                 : <Route
xmlns="urn:schema:Microsoft.Rtc.Management.Settings.SipProxy.2008" MatchUri="CMS1.UC.local"
MatchOnlyPhoneUri="false" Enabled="true" ReplaceHostInRequestUri="false">
                            <Transport Port="5061">
                              <TLS Fqdn="CMS1.UC.local">
                                <UseDefaultCert />
                              </TLS>
                            </Transport>
                          </Route>

Transport               :
TransportChoice=Certificate=Microsoft.Rtc.Management.WritableConfig.Settings.SipProxy.UseDefault
Cert;Fqdn=CMS2.UC.local;Port=5061
MatchUri                : CMS2.UC.local
MatchOnlyPhoneUri       : False
Enabled                 : True
ReplaceHostInRequestUri : False
Element                 : <Route
xmlns="urn:schema:Microsoft.Rtc.Management.Settings.SipProxy.2008" MatchUri="CMS2.UC.local"
MatchOnlyPhoneUri="false" Enabled="true" ReplaceHostInRequestUri="false">
                            <Transport Port="5061">
                              <TLS Fqdn="CMS2.UC.local">
                                <UseDefaultCert />
                              </TLS>
                            </Transport>
                          </Route>

Transport               :
TransportChoice=Certificate=Microsoft.Rtc.Management.WritableConfig.Settings.SipProxy.UseDefault
Cert;Fqdn=CMS3.UC.local;Port=5061
MatchUri                : CMS3.UC.local
MatchOnlyPhoneUri       : False
Enabled                 : True
ReplaceHostInRequestUri : False
Element                 : <Route
xmlns="urn:schema:Microsoft.Rtc.Management.Settings.SipProxy.2008" MatchUri="CMS3.UC.local"
MatchOnlyPhoneUri="false" Enabled="true" ReplaceHostInRequestUri="false">
                            <Transport Port="5061">
                              <TLS Fqdn="CMS3.UC.local">
                                <UseDefaultCert />
                              </TLS>
                            </Transport>
                          </Route>



PS C:\Users\administrator.SKYPE>
```

Contacting TAC

If you encounter errors with this implementation please contact Cisco TAC. When opening the service request please include a link to this document. It will help the TAC engineers understand your configuration. Additionally, it would be extremely helpful if the Cisco Meeting Server logs are attached to the case as described above and the output of all of the Get commands from the Lync/Skype Front End are entered into the case notes. If you don't include this information it's sure to be one of the first things the TAC engineers ask for so please go ahead and collect it before opening your case.