Cisco Meeting Server

Abhishek Pal

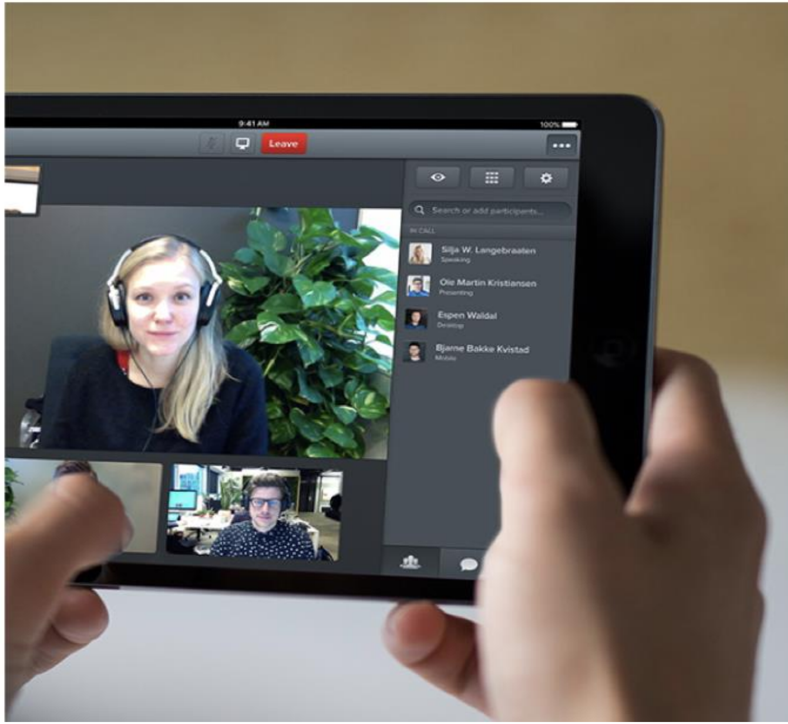
Agenda

- CMS Fundamentals
- CMS Components
- CMS Platform Option
- CMS Deployment options and configuration
- CMS Dial Plan
- CMS-UC Integration (CUCM, VCS)
- CMS Certificates

General Overview

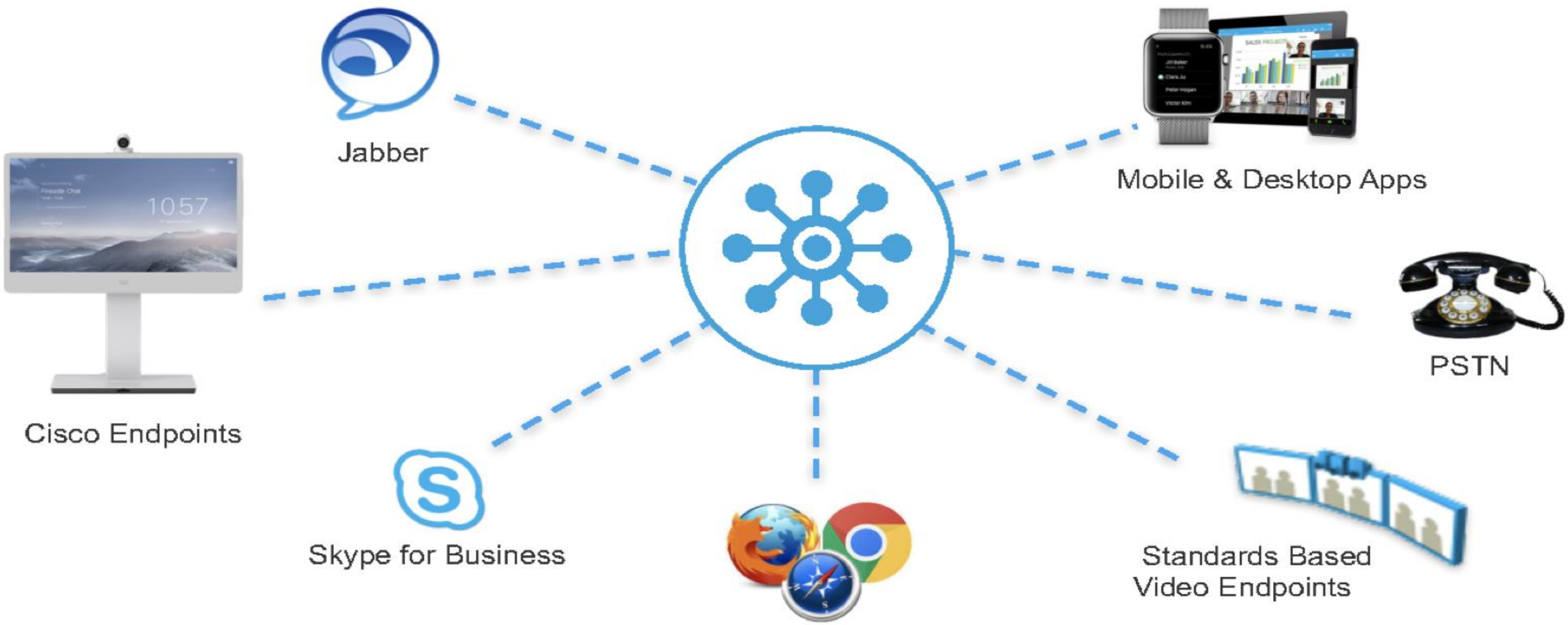
Cisco Meeting Server
Complete Conferencing Platform

- Video Conferencing
- Apps
- Interoperability
- Audio Conferencing
- Spaces
- Customisation
- Recording / Streaming
- Web Conferencing



Cisco Meeting Server

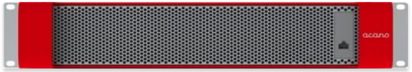
A Single Meetings Platform – Where everyone is invited



Cisco Meeting Server
Meet the Way You Want

- **Personal Spaces:**
  - Invite others to your personal space using your own join details
  - With Spaces – users are in control
- **Scheduled Spaces:**
  - Leverage Cisco TelePresence Management Suite (including Microsoft Outlook integration)
  - One-Button-to-Push support
- **Ad-hoc with UCM:**
  - Easily escalate your 1:1 calls to include more people
- **Interoperability Gateway:**
  - Enable native calling and content sharing between Skype for Business

CMS Platform Options



Acano Server [EOS]

CMS 1000/2000
MM410v, MM400v (legacy)

Spec Based Server

Cisco Meeting Server

Acano X Series Server – End of Support (Nov 2021)

- X3
  - **250** HD* calls
  - 500 SD calls
  - 600 Skype for Business video calls
  - 1,500 web calls (audio & content)
  - 3,000 audio calls
- X2
  - 125 HD* calls
  - 250 480p calls
- X1
  - 20 HD* calls
  - Typically used for Edge Services

X Series Platform

(2 RU)

Bare Metal – no Vmware

Multiple Interfaces – Dedicated MMP interface

Supports SIP Trunk
- CUCM
- VCS

Supports Trusted SIP Trunk (SIP TLS)
- Lync/Skype for Business

Cisco Meeting Server 1000

Released in August 2016

Supports :

- **96** HD* calls
- 192 SD calls
- 192 Skype for Business video calls

Hardware :

- UCS C Series server (1 RU)
- 70 Hyperthreaded Cores
- Co-residency not supported
- Vmware ESXi 6.0 and above
- Virtual machine version 11 and above

Cisco Meeting Server 1000

Supports SIP Trunk

- CUCM
- VCS

Supports Trusted SIP Trunk (SIP TLS)

- Lync/Skype for Business

Cisco Meeting Server 2000

New High Capacity Platform

High Capacity

- Up to **500*** HD calls

UCS Blade Server Chassis (6 RU)

- Based on UCS 5108 and B200 Blades
- Compatible with CMS 1000

Bare Metal platform

- No Vmware ESXi required

Cisco Meeting Server 2000-Cont’d

- CMS 2000 is an Appliance (no vmware)
- All 8 blades act as a single unit. No need to cluster or cascade the blades
- Is not available with single blade
- Can be clustered with another CMS 2000, CMS 1000 or spec based server deployment
- Server has 4 hot swappable power supplies
- CMS 2000 uses serial over LAN (SoL) connection to provide access to the MMP

CMS Components

Core Components

Call Bridge	Is the component that bridges conference connections together into a single conference
WebBridge	If you are using the CMA WebRTC Client you will need to enable and configure the WebBridge
Database	The CallBridge reads from and writes to the database storing the space and configuration information
WebAdmin	The WebAdmin is a web interface to administer the CallBridge, typically running on 443, unless the WebBridge is running on the same interface, then it should be moved to 445 or 8443
XMPP Server	The XMPP server handles the signalling and media to and from CMA clients, including the WebRTC client
Recorder	Enables automatic recording on meeting start, triggered via DTMF or administrator defined.

CMS Components

Core Components – continued..

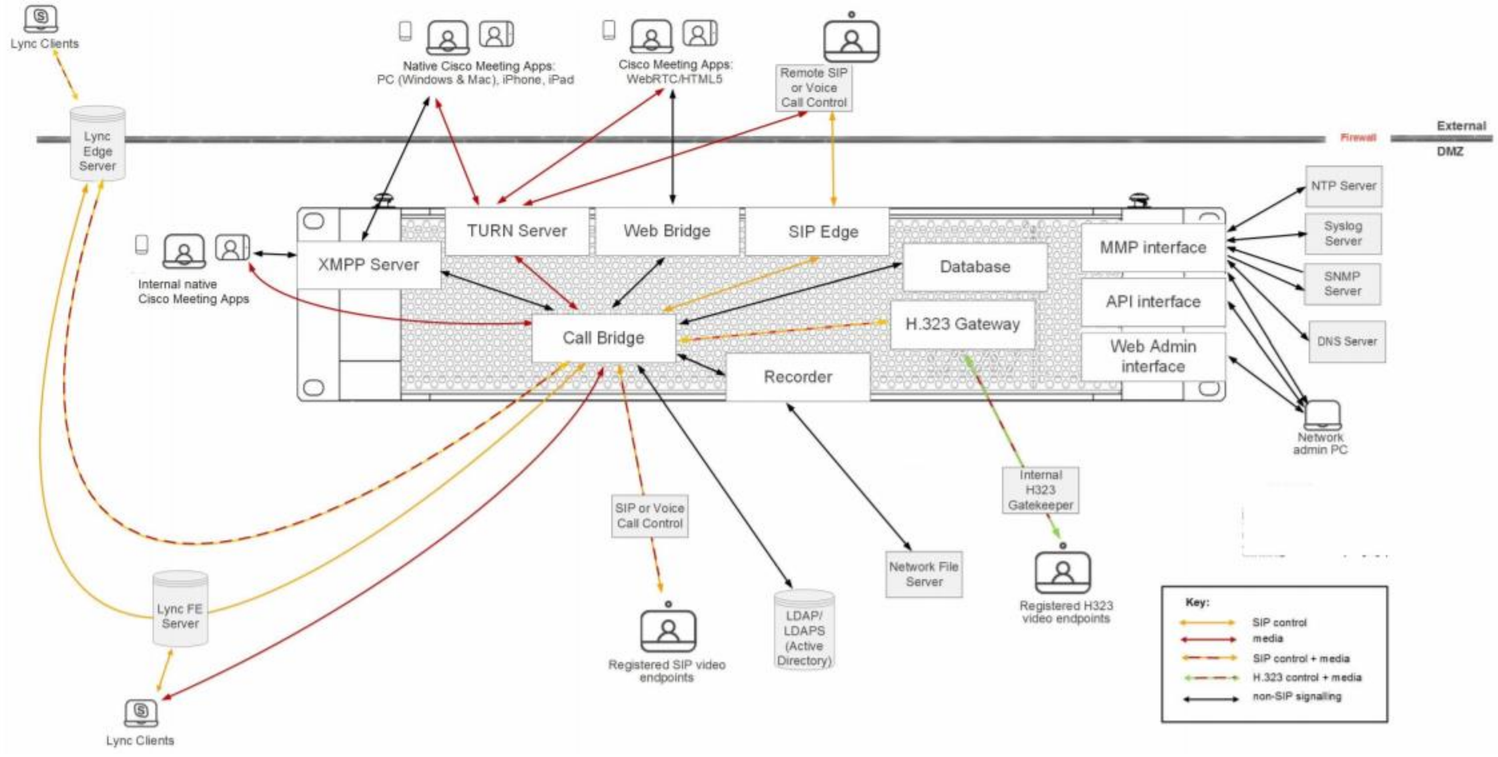
Streamer	Streamer destination Url defined by API. VBrick supported as external streaming server.
H.323 GW	H.323 Gateway enables a H.323 call to connect to the CMS CallBridge

Edge Components

Load Balancer	The Load Balancer (LB) acts as a proxy to the internal XMPP Server, providing secure firewall traversal for external CMA clients in split deployments
TURN Server	The TURN server provides firewall traversal technology, allowing the Meeting Server to be deployed behind a Firewall or NAT
SIP Edge	To support traversal of local firewalls for SIP endpoints and open and direct federation for O365, SfB and Lync calls

Single Combined Deployment

Single Combined Deployment



Server Components & Configuration

Web Admin

The first Component we configure is the WebAdmin. The WebAdmin is the Service to enable Web GUI for Meeting server

- WebAdmin is specifically to configure how the Call Bridge talks to other components
- Required for Call routing configuration
- Required for Callbridge Clustering configuration
- Required for viewing logs via web and set log to debug level

Web Admin Config

Configure Webadmin using MMP, We need to set Webadmin listen interface, add certificate and key, and enable the service.

- “webadmin listen <iface> <port>”
- “webadmin certs <key> <cert>”
- “webadmin http-redirect enable”
- “webadmin enable”

```
Configure webadmin

Usage:
    webadmin
    webadmin restart
    webadmin enable
    webadmin disable
    webadmin listen <interface> [<port>]
    webadmin certs <key-file> <crt-file> [<cert-bundle>]
    webadmin certs none
    webadmin http-redirect <enable/disable>
    webadmin status
Core1>
```

Call Bridge

The next Component is the Callbridge. CallBridge bridges the conference connections, enabling multiple participants to join meetings hosted on the Meeting Server

- Primary component of the solution
- Must exist somewhere in all deployments
- Media processing engine
- API integration point
- Supports clustering for distributed calls

Call Bridge Configuration

Configure a Callbridge listen interface, add certificate and key, and restart the service

- "callbridge listen <iface> "
- "callbridge certs <key> <cert>"
- "callbridge restart"

```
Configure Acano callbridge

Usage:

    callbridge listen <interface whitelist>
    callbridge prefer <interface>
    callbridge certs <key-file> <crt-file> [<cert-bundle>]
    callbridge certs none
    callbridge add edge <ip address>:<port>
    callbridge del edge
    callbridge trust edge <trusted edge certificate bundle>
    callbridge restart
Core1>
```

XMPP

The XMPP service to enable the Cisco Meeting Apps such as PC clients and iOS (iPhone and iPad) device to connect the Meeting Server. The XMPP service handles signaling to and from Cisco Meeting Apps.

- Registration point for PC and Mobile clients as well as Web Bridge
- Allows for calls, IM, and presence
- Traversal capable
- Can balance between multiple servers in large deployment
- Requires LDAP source to be configured on Call Bridge

XMPP Config

After configuring XMPP, add the callbridge (so that the callbridge later securely can connect to the XMPP service)

- “xmpp listen <iface>”
- “xmpp certs <key> <cert>”
- “xmpp domain <xmpp_login_domain>”
- “xmpp enable”
- “xmpp callbridge add <callbridge_name>”

```
Configure XMPP server

Usage:
    xmpp
    xmpp (enable|disable)
    xmpp restart
    xmpp reset
    xmpp domain <domain name>
    xmpp callbridge add <callbridge>
    xmpp callbridge add-secret <callbridge>
    xmpp callbridge del <callbridge>
    xmpp callbridge list
    xmpp listen <interface whitelist>
    xmpp certs <key-file> <crt-file> [<crt-bundle>]
    xmpp certs none
    xmpp motd add "<message>"
    xmpp motd del
    xmpp max_sessions <number>
    xmpp max_sessions none
    xmpp status
    xmpp multi_domain add <domain name> <key-file> <crt-file> [<crt-bundle>]
    xmpp multi_domain del <domain name>
    xmpp multi_domain list
    xmpp cluster (enable|disable)
    xmpp cluster trust none
    xmpp cluster trust <trust bundle>
    xmpp cluster status
    xmpp cluster initialize
                join <leader>
                remove [<node>]
```

General configuration

XMPP server settings

Unique Call Bridge name	core1
Domain	cmslab.com
Server address	127.0.0.1
Shared secret	•••••••••••••••••••• [cancel]
Confirm shared secret	••••••••••••••••••••

Web Bridge

The Webbridge service to enable WebRTC app. The WebRTC app works with browsers and uses the WebRTC standard for video and audio

- Allows for a guest to join via special link or full access to “web” version of desktop client
- Utilizes XMPP signaling (acts similar to desktop client between itself and Call Bridge)
- Chrome, Firefox, and Opera supported, Chrome is preferred
- Must use different port or IP from Web Admin if on the same server

Web Bridge Configuration

Set Webbridge with a listen interface, key and certificate, add a trust towards the callbridge, and enable the service

- "webbridge listen <iface> <port>"
- "webbridge certs <key> <cert>"
- "webbridge trust <callbridge_cert/ca_cert>"
- "webbridge http-redirect enable"
- "webbridge enable"

```
Configure webbridge

Usage:
    webbridge
    webbridge restart
    webbridge enable
    webbridge disable
    webbridge listen <interface[:port] whitelist>
    webbridge certs <key-file> <crt-file> [<crt-bundle>]
    webbridge certs none
    webbridge trust <crt-bundle>
    webbridge trust none
    webbridge http-redirect (enable|disable)
    webbridge clickonce <url>
    webbridge clickonce none
    webbridge msi <url>
    webbridge msi none
    webbridge dmg <url>
    webbridge dmg none
    webbridge ios <url>
    webbridge ios none
    webbridge status
Core1>
```

Web bridge settings

Guest account client URI: https://join.cmslab.com

Guest account JID domain: cmslab.com

External access

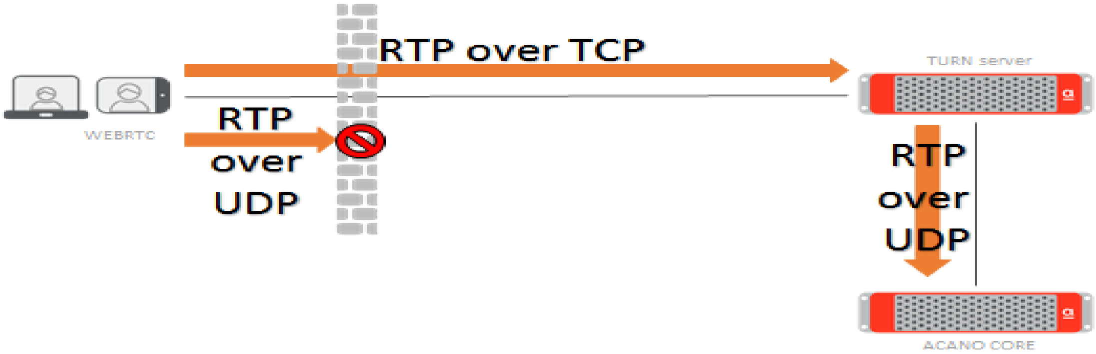
Web Bridge URI: https://join.cmslab.com

TURN Server

- Necessary for NAT traversal when clients are connecting externally
- Provides a media path in situations when direct media is not possible
- H.323 Capable
- Included as an option in all deployments (no additional licensing)

TURN server support for TCP to UDP interworking

- Allows TCP media from browser clients to be received
- TURN server converts this back to UDP media
- Useful when UDP traffic from browsers is blocked



TURN Server Config

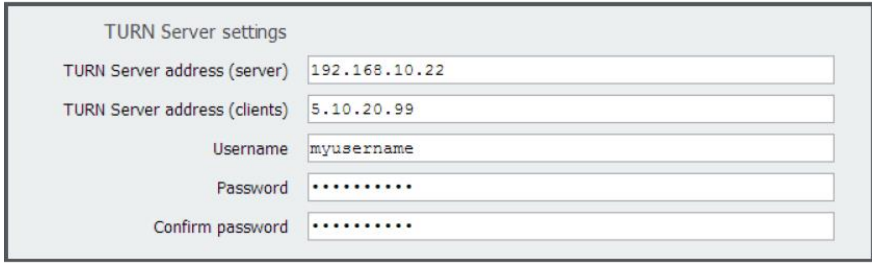
- Basic TURN Server setup
- "turn listen <iface>"
- "turn credentials <username> <password> <domain>"
- "turn enable"
- "turn public-ip <IP>"
- "turn tls 443"

```
Configure TURN server

Usage:
    turn enable
    turn disable
    turn restart
    turn credentials <username> <password> <realm>
    turn public-ip <ip address>
    turn del public-ip
    turn listen <interface whitelist>
    turn tls <port|none>
    turn certs <key-file> <crt-file> [<crt-bundle>]
    turn certs <none>
Core1>
```

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

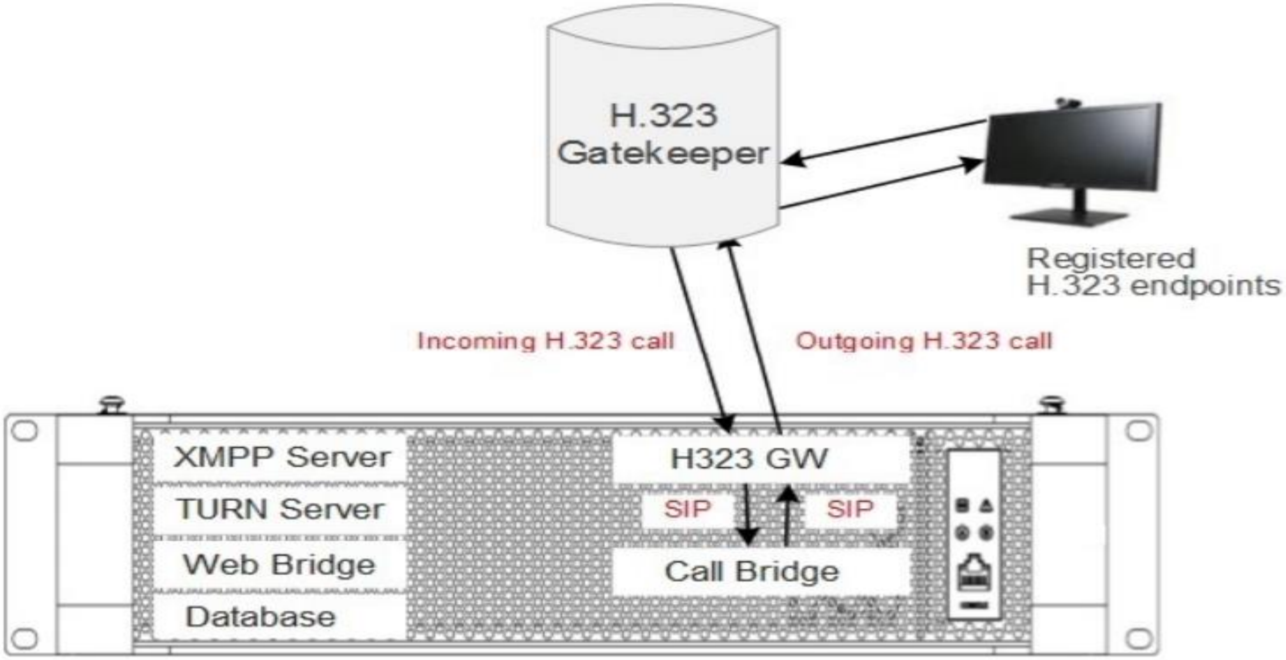
For maximum connectivity from external locations, Cisco recommends that port 443 is used for both the Web Bridge and TURN Server.



H.323 Gateway

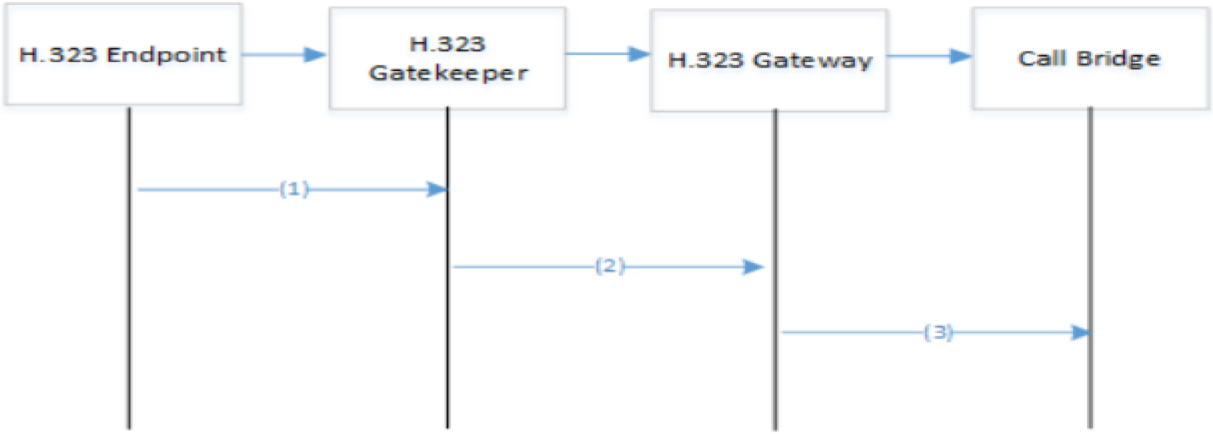
- Allows for external (b2b) calls via H.323 into the solution
- A requirement to provide h.323 support as the Call Bridge only operates in SIP
- Gateway will convert all inbound H.323 traffic to SIP for internal communication
- Generally positioned on the Edge server in multi-deployment

H.323 Gateway



H.323 Gateway

Figure 9: Call Flow for Inbound Call from Registered H.323 Endpoint



Where:

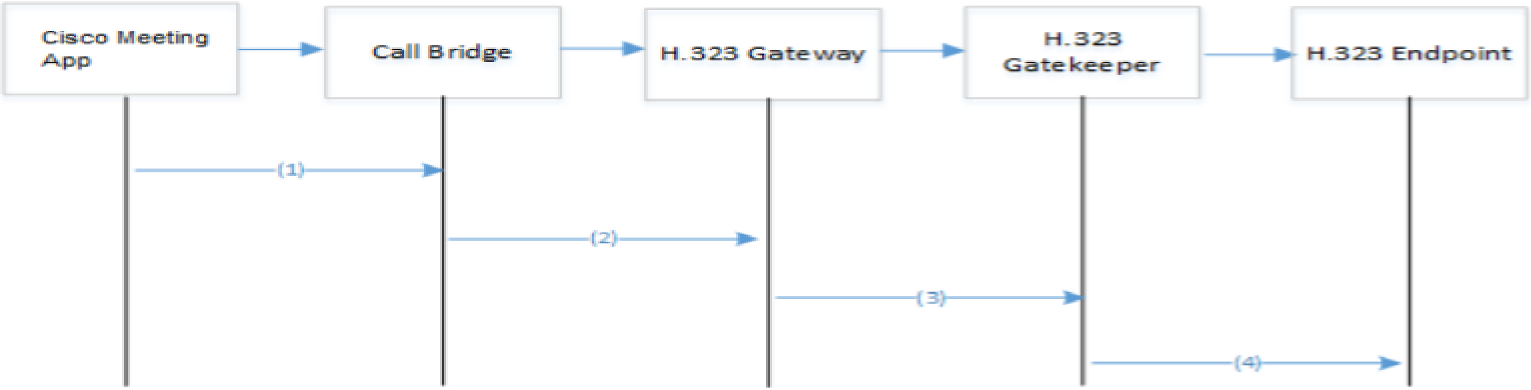
(1) is an H.323 call to example.cospace@example.com

(2) is an H.323 call to example.cospace@example.com

(3) is a SIP call to example.cospace@example.com

H.323 Gateway

Figure 16: Call flow for outbound call to a registered H.323 endpoint



Where:

(1) is a Cisco Meeting App call to h323@h323.com

(2) is a SIP call

(3) is an H323 call

(4) is an H323 call

H.323 Gateway Config

- “h323_gateway h323_interfaces <iface>”
- “h323_gateway sip_interfaces <iface>”
- “h323_gateway sip_port 6061”
- “h323_gateway sip_proxy 127.0.0.1”
- “h323_gateway certs <key-file> <crt-file>”
- “h323_gateway default_uri <IVR/Space>”
- “h323_gateway enable”

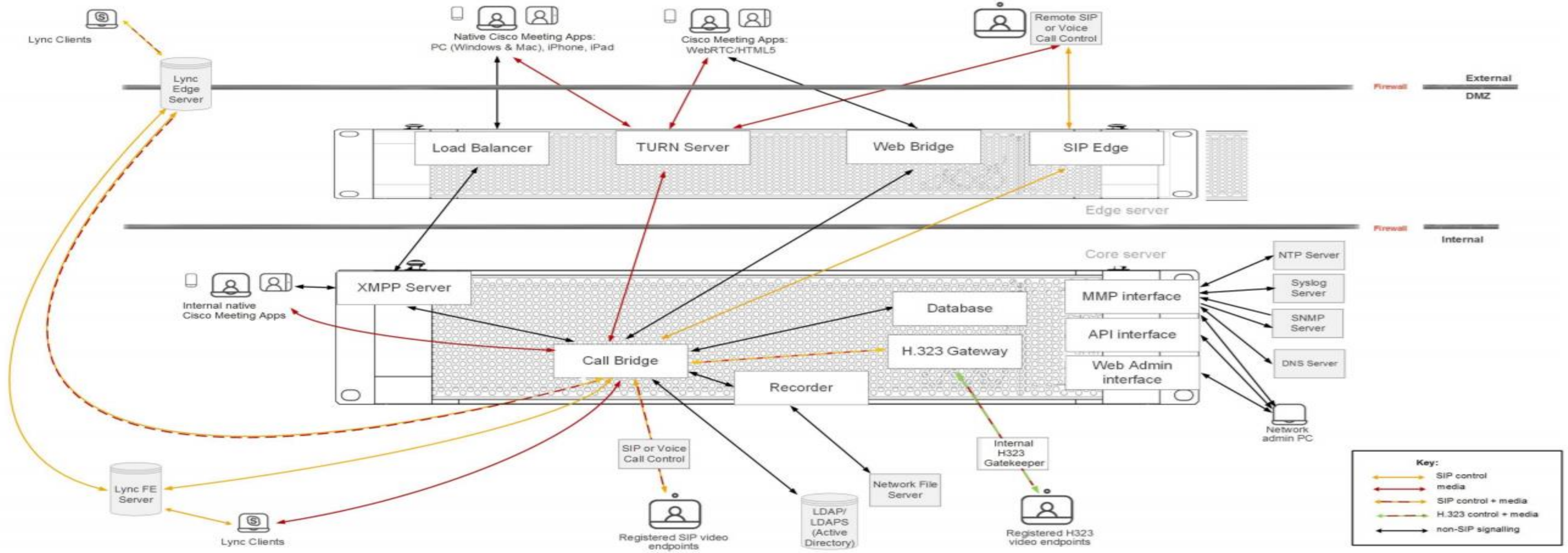
```
lab1> h323_gateway ?
Configure H.323 gateway

Usage:
    h323_gateway enable
    h323_gateway disable
    h323_gateway restart
    h323_gateway default_uri <uri>
    h323_gateway del default_uri
    h323_gateway sip_domain <domain>
    h323_gateway del sip_domain
    h323_gateway sip_domain_strip <yes/no>
    h323_gateway h323_domain <domain>
    h323_gateway del h323_domain
    h323_gateway h323_domain_strip <yes/no>
    h323_gateway h323_interfaces <interface whitelist>
    h323_gateway h323_nexthop <host/ip>
    h323_gateway del h323_nexthop
    h323_gateway sip_interfaces <interface whitelist>
    h323_gateway sip_port <port>
    h323_gateway sip_proxy <uri>
    h323_gateway certs <key-file> <crt-file> [<cert-bundle>]
    h323_gateway certs none
    h323_gateway restrict_codecs <yes/no>
    h323_gateway disable_content <yes/no>
    h323_gateway trace_level <level>
lab1>
```

cisco

Split Server Deployment

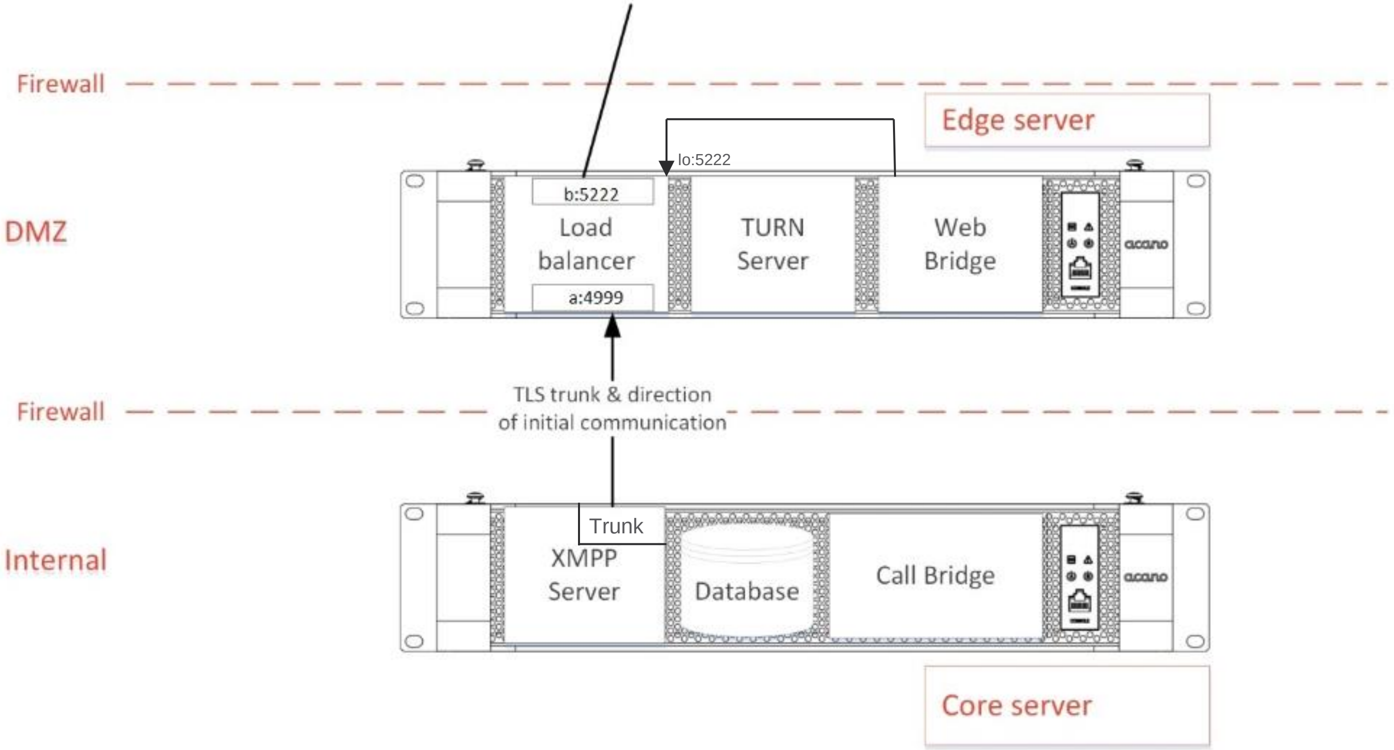
Split Server Deployment



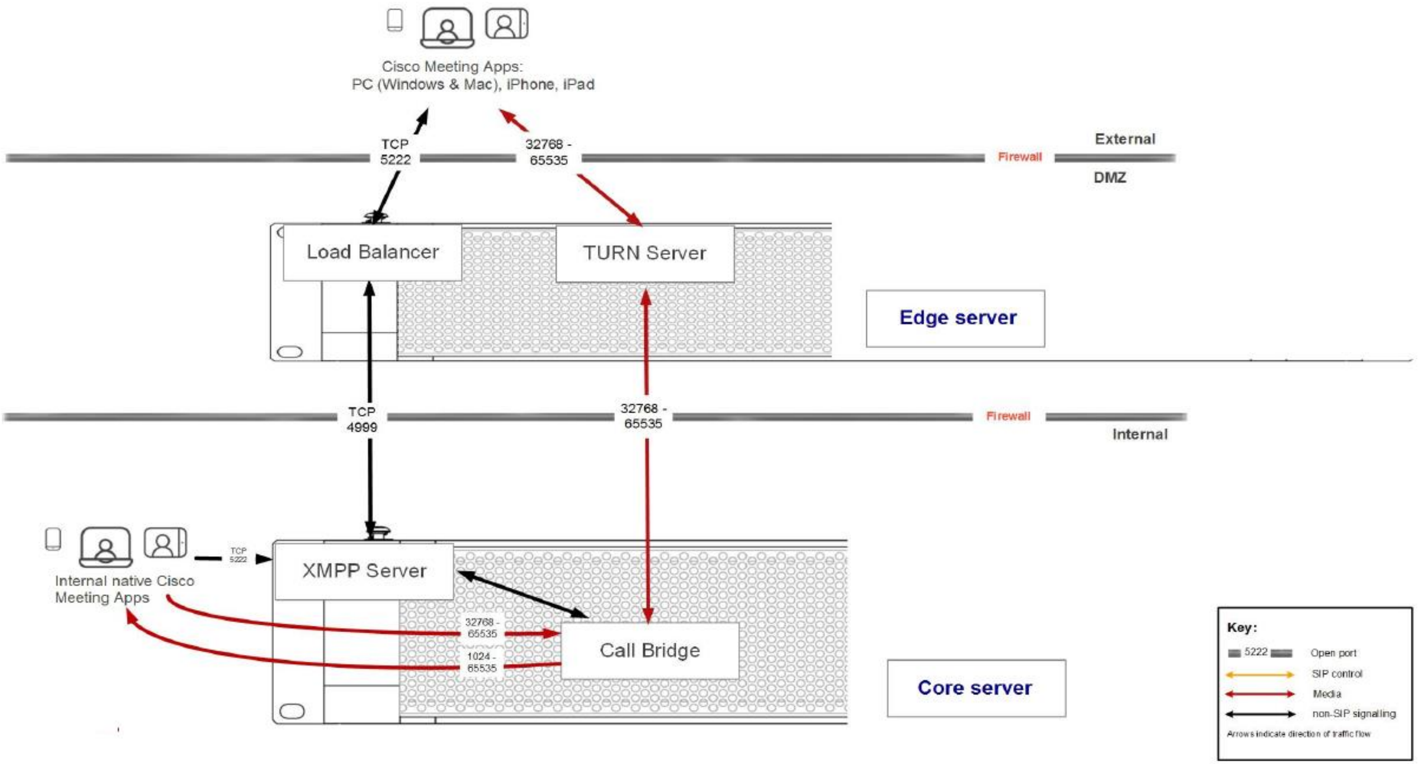
Load-balancers and Trunks

- In a split deployment, the XMPP server is located on the core and the “loadbalancer” is located on the edge.
- The trunk provides the connection to the load balancer on the core side to tunnel traffic internally to the xmpp server.
- The loadbalancer does not really distribute load, but rather provides one of potentially multiple points for traffic to be passed to the XMPP server.
- The load balancer never initiates connections, but listens both internally and externally. The associated ports and interfaces are customizable, but by default internal communication from the trunk is on port 4999 while external communication from clients is on 5222.
- The external side should also listen on the loopback interface if a webbridge is on the same server as the loadbalancer.

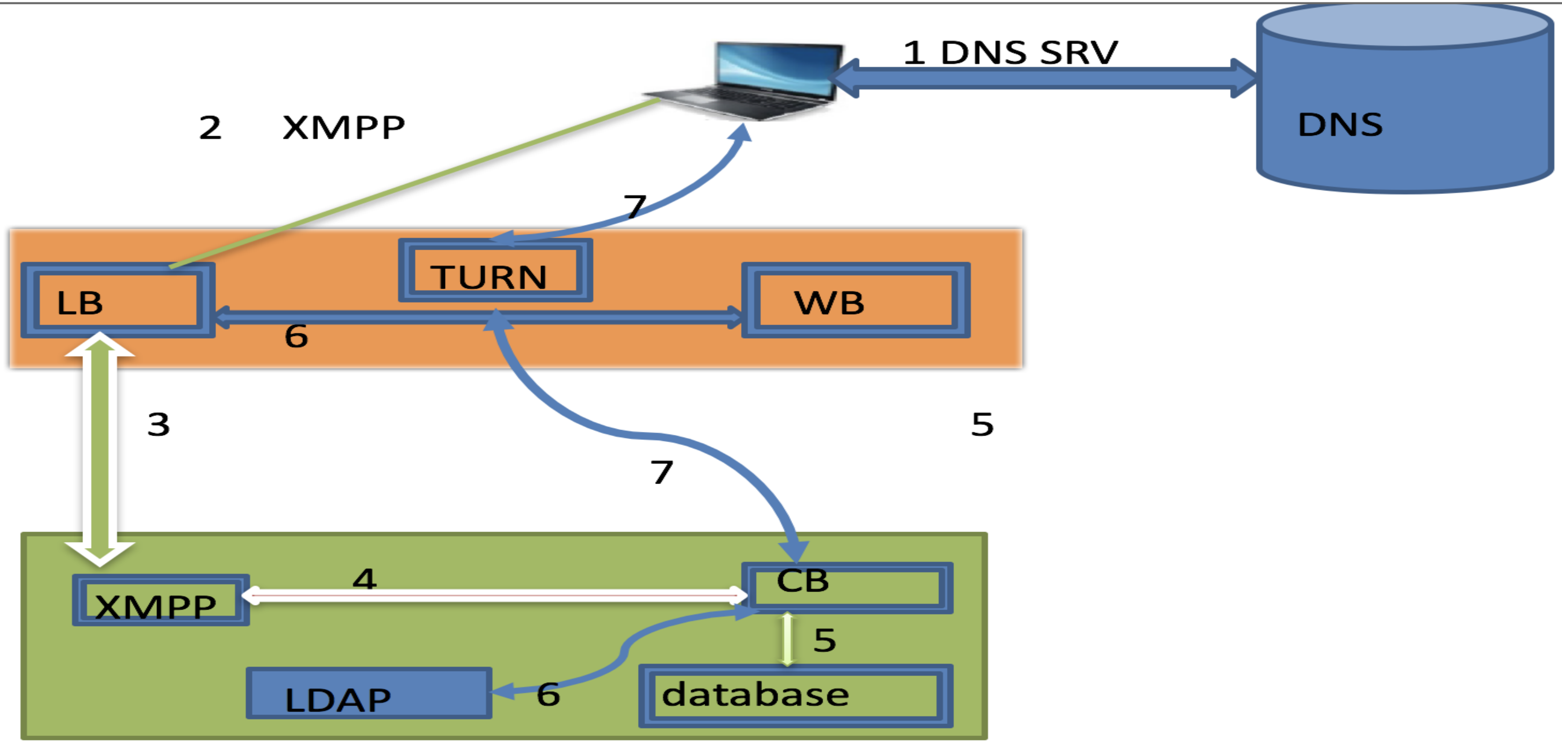
Load-balancers and Trunks (Cont'd)



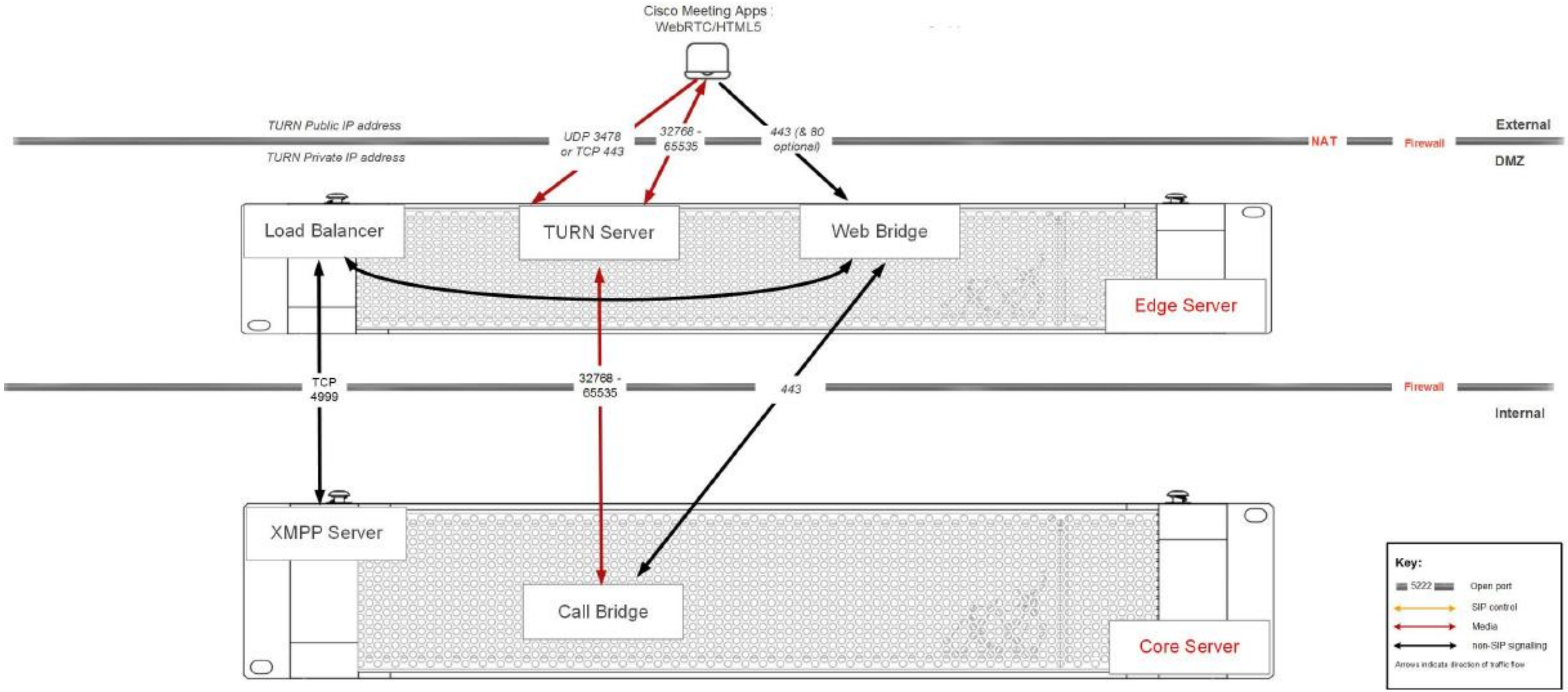
Split Server Deployment – PC client Authentication

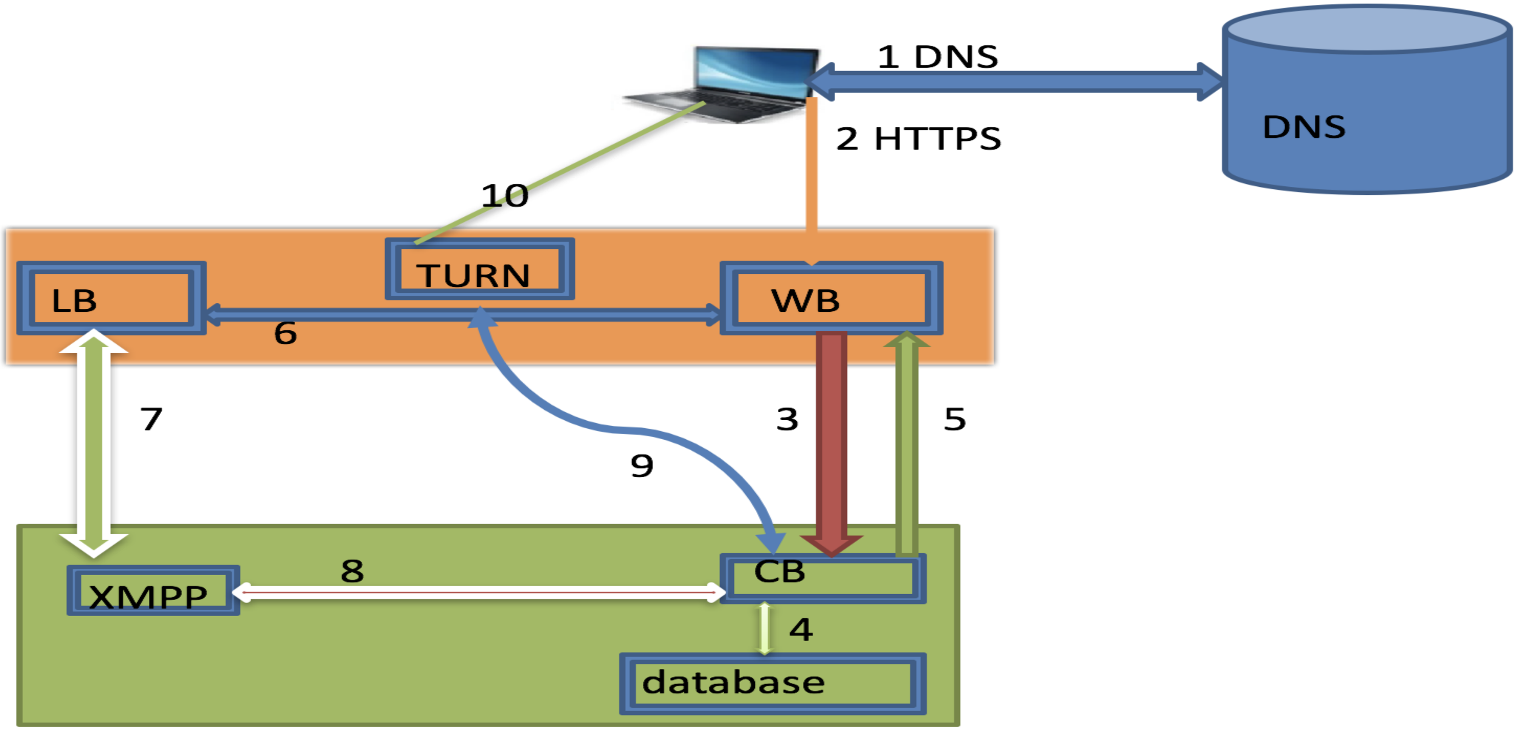


CMA Call Flow



Split Server Deployment – Web Bridge Connection





webRTC Call Flow

Load-balancer and Trunk Configuration

- On the edge server:
  - loadbalancer create edge
  - loadbalancer auth edge loadbal.key loadbal.crt trunk.crt
  - loadbalancer public edge a:5222 lo:5222
  - loadbalancer trunk edge a:4999
  - loadbalancer enable edge
- On the core server:
  - trunk create toedge xmpp
  - trunk auth toedge trunk.key trunk.crt loadbal.crt
  - trunk edge toedge edge.lab1.cmslab.com 4999
  - trunk enable toedge

```
Core1> trunk debug edge1
Trying to connect to trunk local service, port 5222
Success
Resolved name acano-edge1.tkratzke.local to the following:
14.80.82.33:4999
Trying to connect to 14.80.82.33:4999
Connection created [14.80.82.33:4999 -> 14.80.82.30:39999]
Diagnostics request written to edge
Reading diagnostics
{
    "0": {
        "core": {
            "connection": "[::ffff:14.80.82.30:34887 -> ::ffff:14.80.82.33:4999]"
        }
    },
    "process": {
        "memory": {
            "size": "11623",
            "resident": "1020",
            "share": "796",
            "text": "191",
            "lib": "0",
            "data": "303",
            "dt": "0"
        }
    }
}

Core1>
```

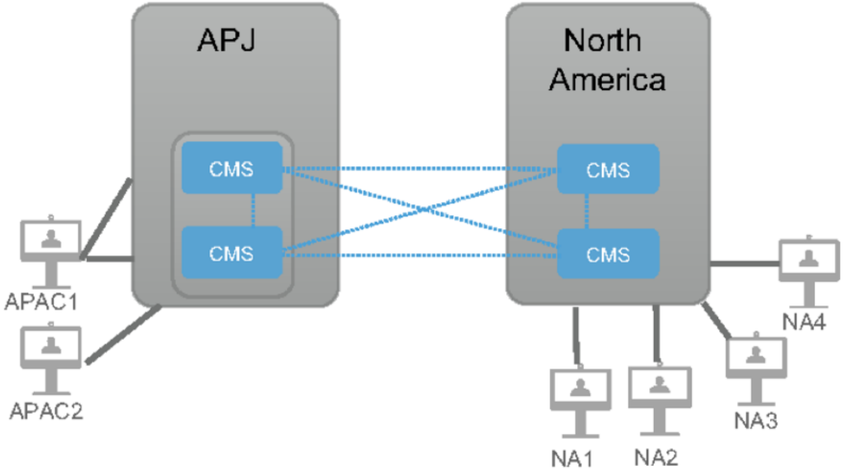
The command "trunk debug <trunk_name>" will show connection statistics to the associated load balancer.

Redundant Deployments and Clustering

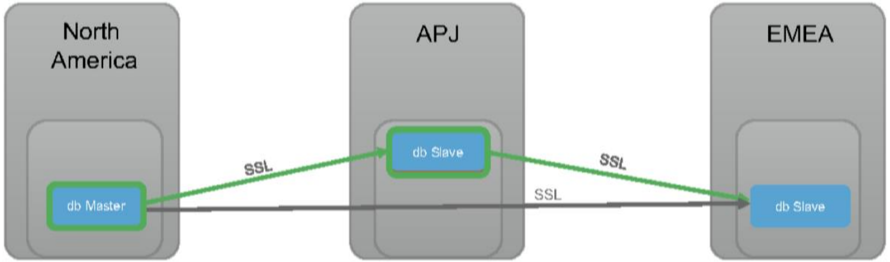
Scalable & Resilient Setup

- Increase capacity and resiliency
- Database are clustered, automatically establishing distributed links for same conference among servers
- Maximum Servers up to 8* per cluster

Picture showing Cluster of 4 nodes

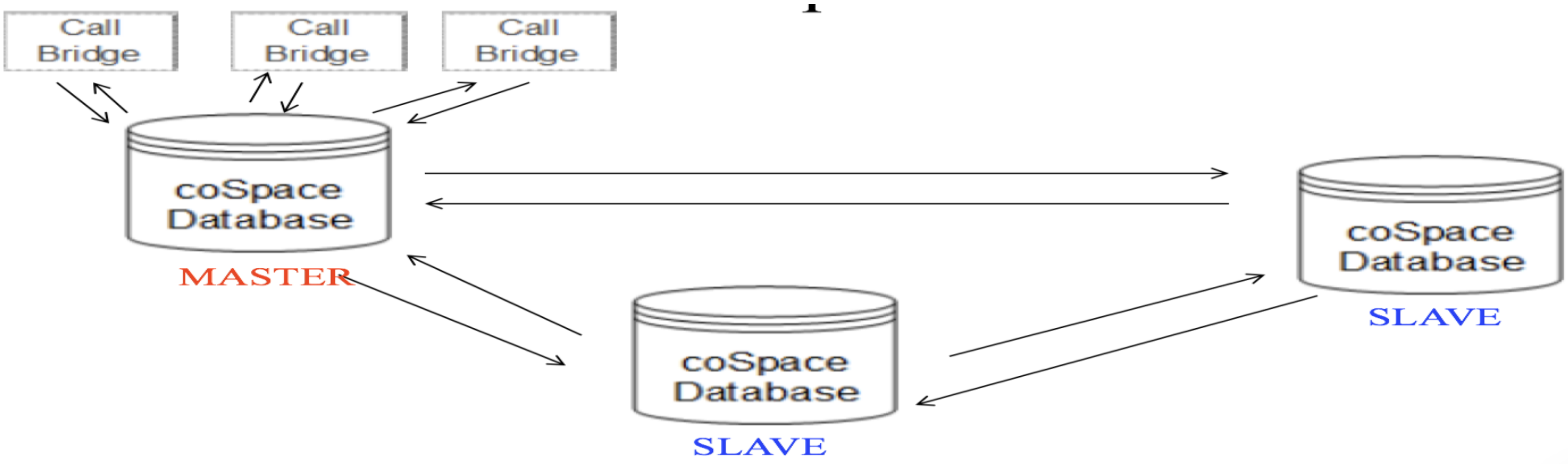


Database Cluster Concepts

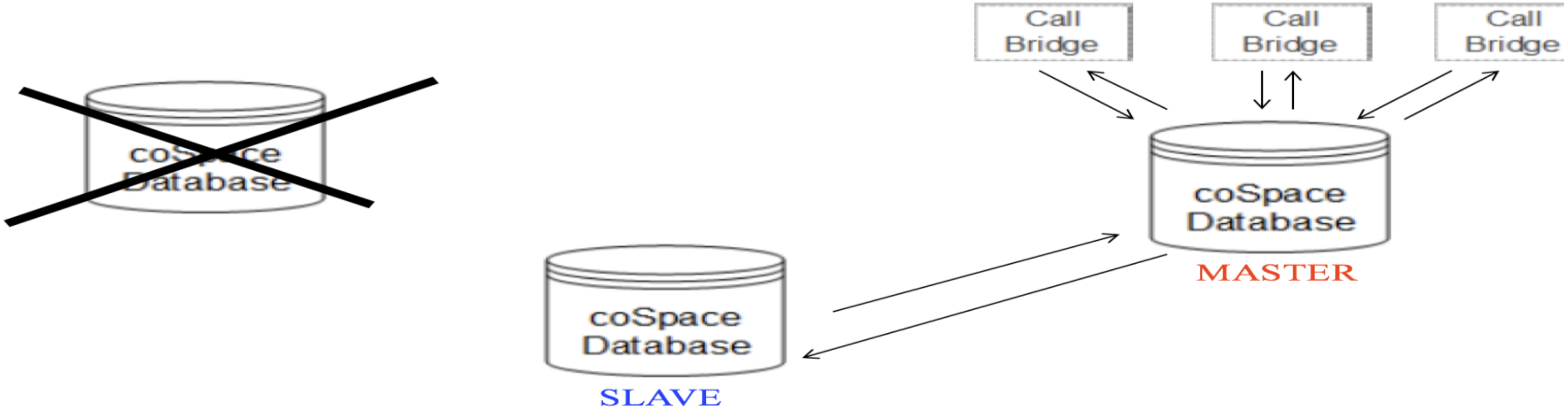


- It is highly recommended to use minimum of 3 nodes for database clustering and maximum can be upto 5 nodes
- Latency must be below than 200ms among servers
- All CMS uses “POSTGRES” for database
- With Certificates (do not use self signed) communication is over SSL using port 5432
- Call Bridge clustering required database cluster in place
- When clustering, keep alives are sent, and if they fail 5 times, one of the other peers will be promoted to master.
- Failover takes approx. 10-15 seconds to elect a new master. If the old master comes back online, it will remain a slave.
- When a node acting as a slave, the database reverted to read only access

- The master database is used by all of the call bridges for reading and writing
- The master database is replicated to the slaves



- If the master fails (due to power or network failure), a slave will become the new master.
- When the master recovers, it will be a slave



Database Clustering Certificates

- If a database cluster is created and secured, all certs must be signed by the same CA, therefor they CANNOT be self signed (private CA can be used)
- Two CSRs must be created, one for the "database client" cert pair and one for the "server" cert pair
- The CN for the client csr MUST be "postgres". The server will throw an error when initializing or joining a cluster if this is incorrect
- The dbclient and dbserver certificate and key pairs must be uploaded to all nodes in the cluster along with the CA certificate list
- It is possible to run a callbridge on a server without a local database in the cluster. If this is done, the client key and cert (and CA trust list) must be uploaded to the callbridge as well
- The standalone callbridges (without database) connect to the cluster with the "database cluster connect <IP/Hostname>" command

Database Clustering Commands

- “database cluster localnode <interface>”
- “database cluster certs dbserver.key dbserver.crt dbclient.key dbclient.crt dbcluster_ca.crt”
- If master database…
  - “database cluster initialize”
- If peer database…
  - “database cluster join <IP/Hostname>”
- “database cluster status”
- To add standalone callbridge to database
- “database cluster connect <IP/Hostname>”

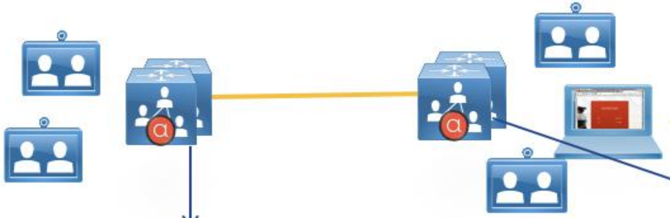
Database Cluster Upgrade Process

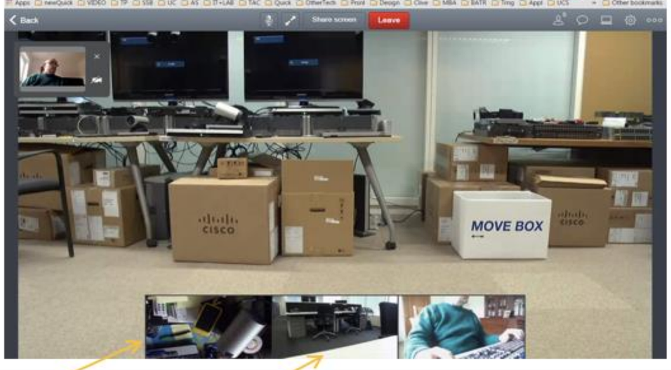
- When initiating an upgrade, first a backup should be taken with the “backup snapshot” command
- Upgrade each node of the cluster one by one starting with the peers (master should be last)
- Wait until each server has fully booted and the database has reconnected to the cluster before moving to the next.
- Once finished, wait for all database nodes to be in sync, then login to the master and issue the command “database cluster upgrade_schema”
- Confirm there are no errors and everything is normal with “database cluster status”
- The following command “database cluster upgrade_schema” is not required when building a new database cluster. It is only required after every subsequent upgrade of cluster

CallBridge Clustering

- Callbridges can be clustered to allow for distributed meetings, where clients connected to multiple bridges appear to be joined together.
- Since all clustered Callbridges require a database cluster, any space created on one bridge by a client or API call is visible to all others.
- All details of a space such as passcodes and URIs carry over as well.
- All Callbridges in the cluster access the same master database.
- If two users call into the same space on different Callbridges, a direct media connection between the two bridges is established.

Distributed Call





Active Calls

Filter [ ] Set   Show only calls with alarms Set

Conference: 8001 (3 active calls; 2 local participants; 3 remote participants)

SIP 14002@tpuc.com [less] (incoming, unencrypted)

call duration	24 minutes, 57 seconds
incoming media	AAC (64.0 Kb/s), H.264, 1280 x 720 29.9fps, 1.21 Mb/s
outgoing media	AAC, H.264, 1280 x 720 29.8fps, 314 Kb/s
remote address	14002@tpuc.com
SIP call ID	c09efa80-6fb17752-932a8-2773330a@10.51.115.39

SIP 14011@tpuc.com [less] (incoming, unencrypted)

call duration	6 minutes, 58 seconds
incoming media	AAC (64.0 Kb/s), H.264, 1280 x 720 30.0fps, 1.22 Mb/s
outgoing media	AAC, H.264, 1280 x 720 30.0fps, 287 Kb/s
remote address	14011@tpuc.com
SIP call ID	43c15000-6fb17b89-2c75c-8c62330a@10.51.98.140

distributed call from "acano-core-uk-143" [less] (incoming, encrypted)

call duration	24 minutes, 43 seconds
incoming media	OPUS, H.264, 1440 x 660 30.5fps, 163 Kb/s
outgoing media	OPUS, H.264, 1920 x 1080 29.9fps, 1.03 Mb/s
remote address	8000@10.51.115.143
SIP call ID	29679ea8-9e7a-46a3-ac24-5588760aa9f6

Conference: 8001 (4 active calls; 3 local participants; 2 remote participants)

SIP 15005@tpbru2.com [less] (incoming, unencrypted)

call duration	5 minutes, 48 seconds
incoming media	AAC (64.0 Kb/s), H.264, 1280 x 720 30.5fps, 1.22 Mb/s
outgoing media	AAC, H.264, 1920 x 1080 29.9fps, 1.22 Mb/s
remote address	15005@tpbru2.com
SIP call ID	723f2b00-6fb17bd7-2c769-8c62330a@10.51.98.140

SIP SX20-6260 (packet loss) [less] (incoming, unencrypted)

call duration	13 minutes, 6 seconds
incoming media	AAC (64.0 Kb/s), H.264, 1280 x 720 12.7fps, 1.16 Mb/s (4.6% packet loss)
outgoing media	AAC, H.264, 1920 x 1080 26.7fps, 1.33 Mb/s
remote address	6260@tpbru2.com
SIP call ID	6d2dac00-6fb17a21-9330f-2773330a@10.51.115.39

distributed call to "acano-core-bru-139" [less] (outgoing, encrypted)

call duration	24 minutes, 51 seconds
incoming media	OPUS, H.264, 640 x 180 30.7fps, 308 Kb/s
outgoing media	OPUS, H.264, 1920 x 1080 25.5fps, 1.31 Mb/s
remote address	f0bb9ce800000001@10.51.115.239
SIP call ID	29679ea8-9e7a-46a3-ac24-5588760aa9f6

Acano user4 user [less] (incoming, encrypted)

call duration	29 seconds
incoming media	OPUS, VP8, 1280 x 720 7.1fps, 1.16 Mb/s
outgoing media	OPUS, VP8, 886 x 474 28.3fps, 296 Kb/s
remote address	user4.acano@tpbru3.tpuc.com

CallBridge Clustering Configuration

- Before clustering Callbridges, a database cluster must be configured and any stand-alone call bridges must be joined to the database cluster.
- Next, in the webadmin page, under Configuration -> Clustering, assign a “**Unique Name**” to each Callbridge to be clustered.
- Then, on one of the peers, add the unique name and address to the webadmin interface of itself, followed by all other peers to be clustered.
- If the connections are successful, you should see this same info on the clustering page of the other peers automatically, along with “connection attempted” and a time since last heartbeat by each peer.

CallBridge Clustering Configuration (Cont'd)

Call Bridge identity

Unique name	core1
Peer link bit rate	
Participant limit	
	Submit

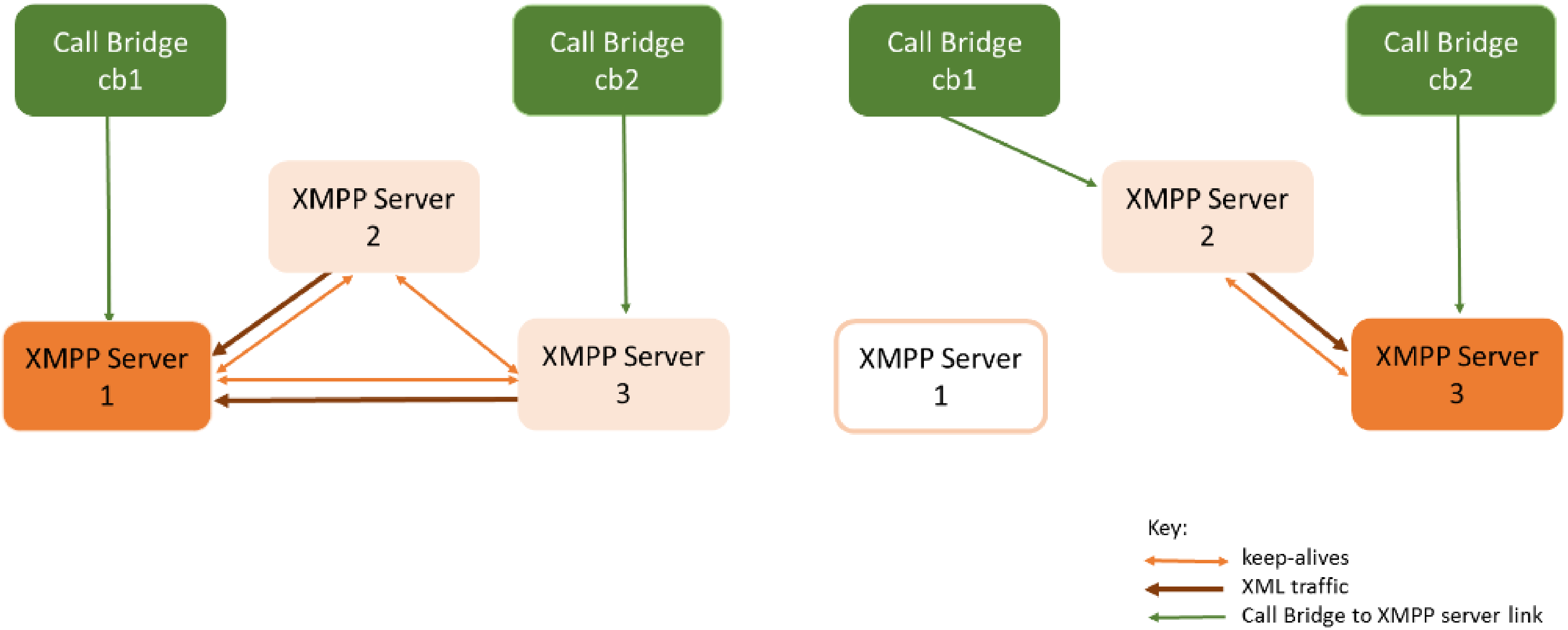
Clustered Call Bridges

	Unique name	Address	Peer link SIP domain	Status	
	core1	https://10.104.215.211:445		[this Call Bridge]	[edit]
	core2	https://10.104.215.212:445		connection active; time since last heartbeat: 8 seconds	[edit]
	core3	https://10.104.215.213:445		connection active; time since last heartbeat: 6 seconds	[edit]
					Add New

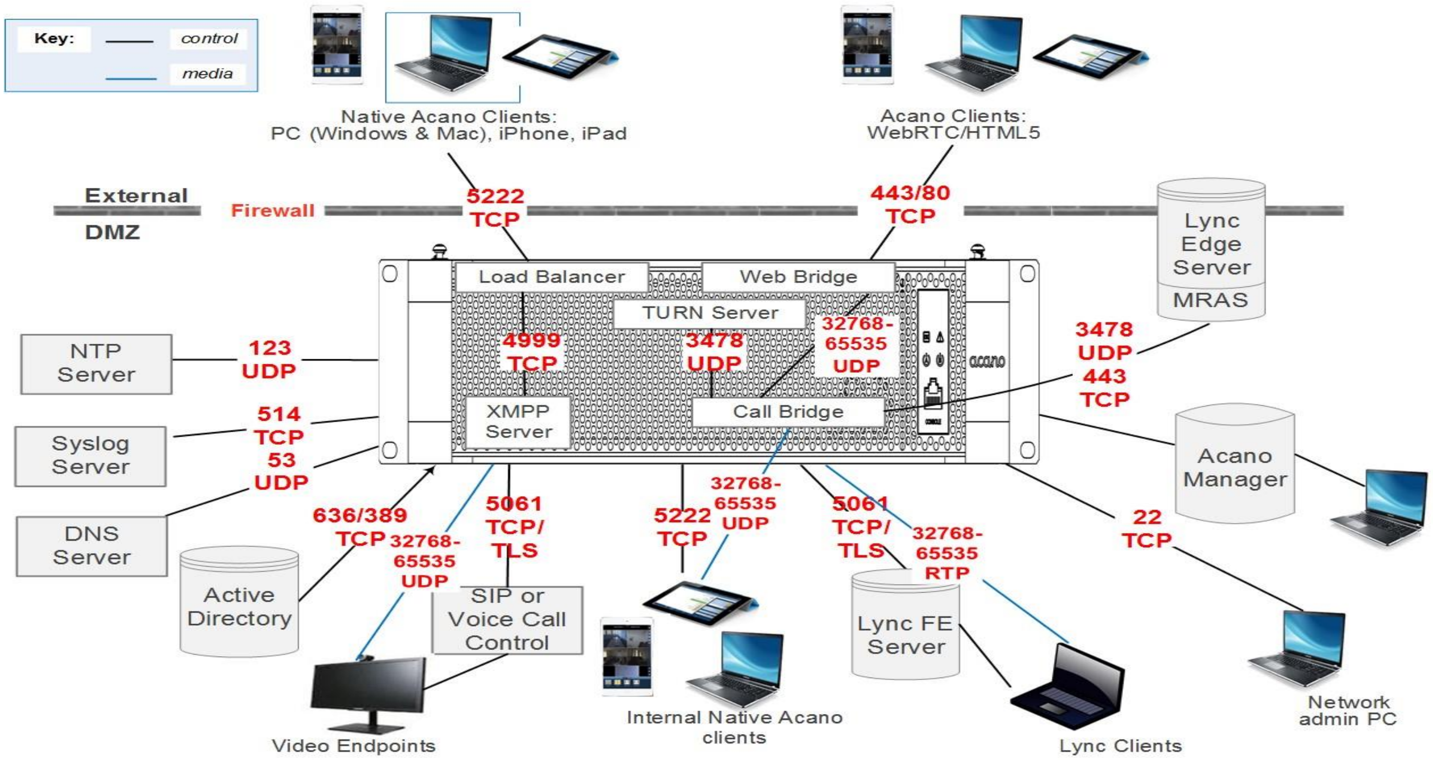
XMPP Resiliency

- Need at least three XMPP servers in the deployment. Those with only two will not benefit due to the algorithm for failover requiring over half of the nodes to be available. Having two servers effectively doubles the chance of an outage.
- All XMPP servers in the cluster know the location of all others and will elect a master.
- All communication will flow through the master XMPP server unless it goes down and a new master needs to be elected.
- The XMPP server that a callbridge connects to is controlled via DNS.
- While a callbridge only connects to one XMPP server at a time, it must be configured along with its shared secret on each XMPP server it could connect to in the event of a failover.
- Configuration example available in the scalability and resilience deployment guide.

XMPP Resiliency (Cont'd)



Reference Port usage



UC Infrastructure Integration

CUCM Integration - Rendezvous

- CUCM needs a trunk with the appropriate route patterns or SIP route patterns in place to send the traffic to the CMS server.

SIP Trunk

Trunks ***(1 - 1 of 1)*** ***Rows per Page*** 50

Find Trunks where Device Name ▾ begins with ▾ Find Clear Filter
Select item or enter search text ▾

		Name ▲	Description	Calling Search Space	Device Pool	Route Pattern	Partition	Route Group	Priority	Trunk Type	SIP Trunk Status	SIP Trunk Duration	SIP Trunk Security Profile
	SIP	CMS-Trunk			Default	900X				SIP Trunk	Full Service	Time In Full Service: 3 days 6 hours 28 minutes	Secure SIP Trunk Profile CMS

Add New | Select All | Clear All | Delete Selected | Reset Selected

Route Pattern

Route Patterns ***(1 - 1 of 1)*** ***Rows per Page*** 50

Find Route Patterns where Pattern ▾ begins with ▾ Find Clear Filter

	Pattern ▲	Description	Partition	Route Filter	Associated Device	Copy
	900X				CMS-Trunk	

Add New | Select All | Clear All | Delete Selected

CUCM Integration - Rendezvous (Cont'd)

- Ensure that the CMS inbound dial rules take into account the format the CUCM will be sending the URI in.

Incoming rule on CMS server

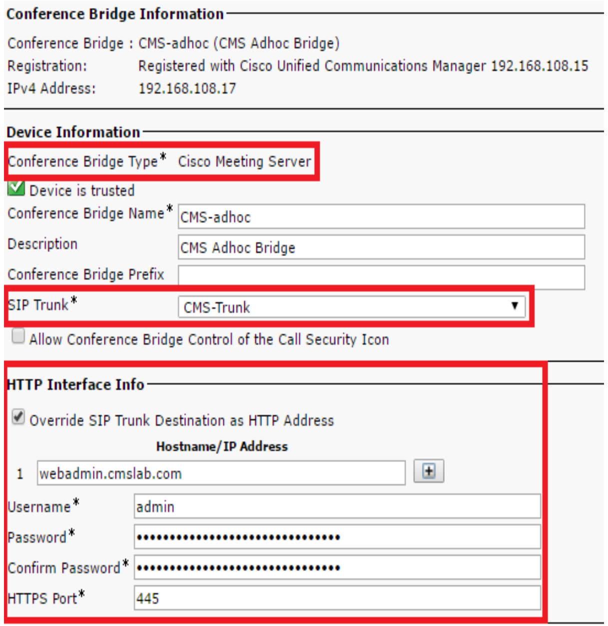
Incoming call handling

Call matching

■	Domain name	Priority	Targets spaces	Targets users	Targets IVRs	Targets Lync	Tenant	
	192.168.108.15	10	yes ▾	yes ▾	yes ▾	no ▾		Add New Reset

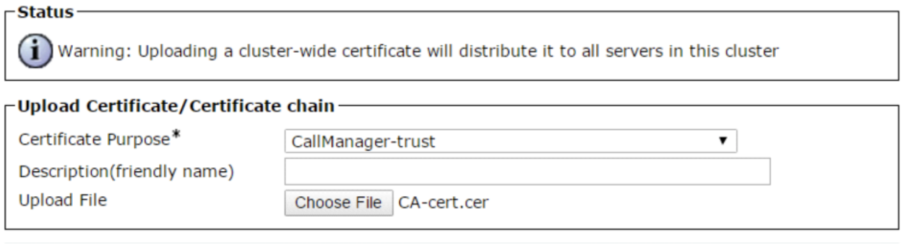
CUCM Integration - Adhoc

- CMS can be added as a conference bridge in CUCM
- Once CMS registered as conference bridge, create appropriate MRG and MRGL
- Assign the MRGL to Phones



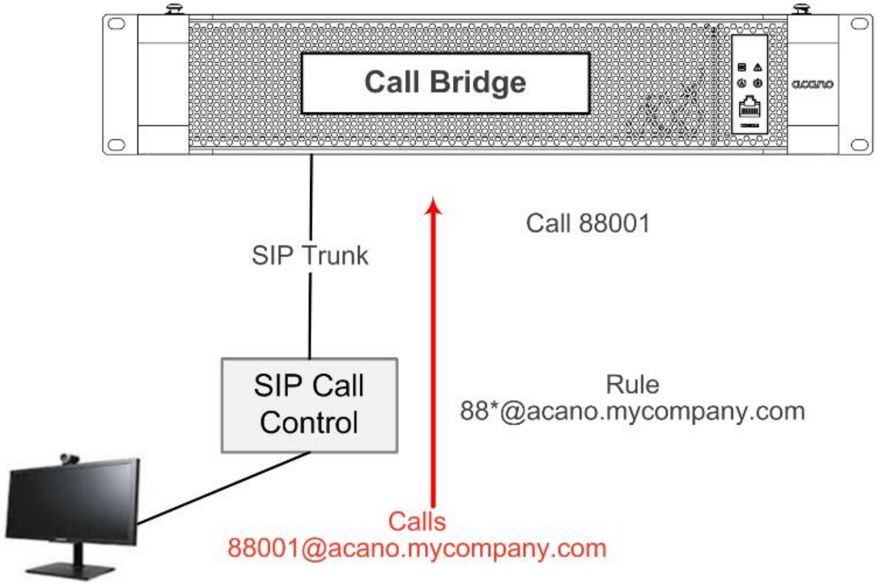
CUCM Adhoc-continued

- Upload the callbridge certificate and CA bundle in CallManager-Trust
- Upload the WebAdmin certificate and CA bundle in tomcat-trust
- Cisco Unified Communications Manager has some requirements on what TLS certificates it will accept. You should “csr" has the SSL client and SSL server purposes enabled. This is done during the certificate signing stage.



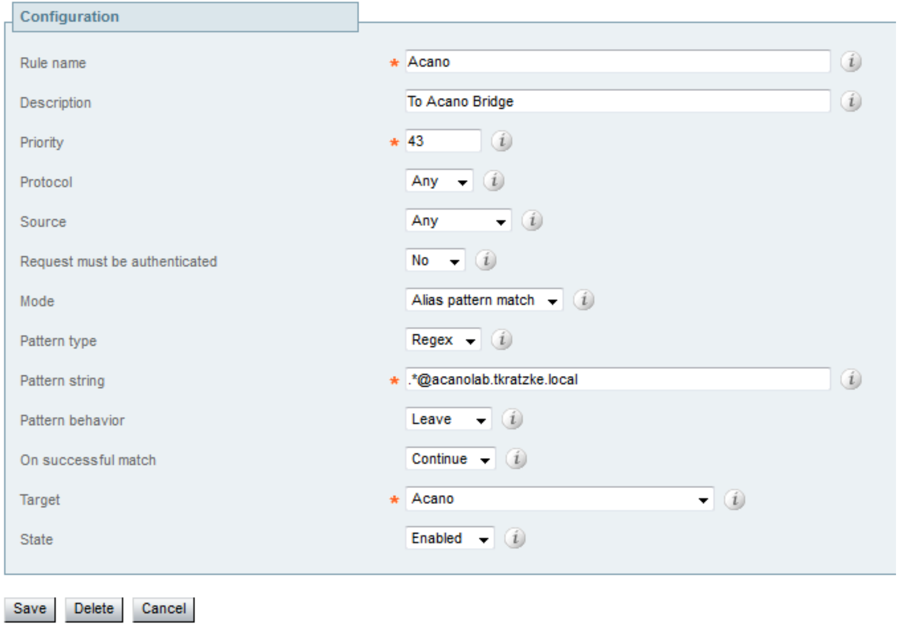
VCS Integration

- Implemented via a neighbor zone pointing to the CMS call-bridge.
- Outbound domain can be transformed via search rule if necessary for inbound matching.
- VCS should also be configured to handle incoming calls appropriately.

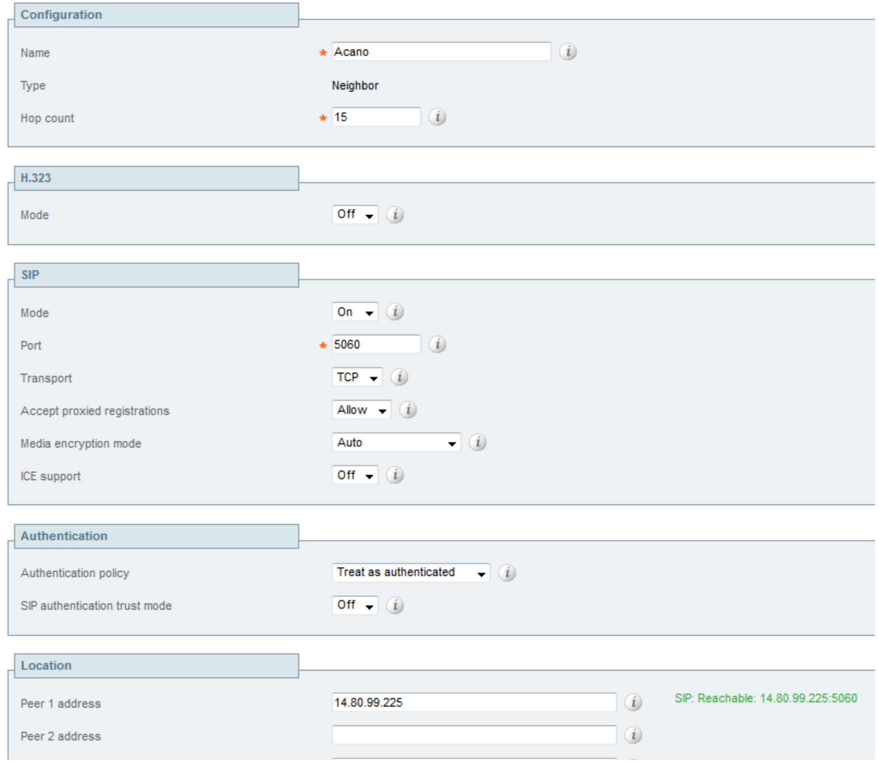


VCS Integration (Cont'd)

Search Rule



Neighbor Zone



Dial Plan Overview

Local Dial Plan

Outbound Call webadmin Configuration describes:

- Where CMS server send a call based on domain in URI
- Any transformation that need to be done on URI

Incoming Call webadmin configuration describes if incoming call need to be :

- Handled locally (spaces, users)
- Forwarded to an external destination (e.g. call from CUCM to Lync)
  via dial plan (Outbound Call) configuration

(forward behaviour can be summarized as below:
Incoming Call => *Incoming Call rule (no match)* => *Forwarded Call rule (match) => handle call according to Outbound Call rule*)

Local Dial Plan - Outbound (Cont’d)

Domain: Destination Domain of the outgoing call

SIP Proxy: which SIP trunk to use

Local Contact Domain: to be used only with Lync, FQDN of acano server
(applies to the domain of the “Contact” header in the outgoing SIP INVITE, if blank IP address of the server is used)

Local From Domain: domain used as “From”
(applies to the domain of the “From” header of the outgoing SIP INVITE, if blank domain of the server is used)

Trunk Type: Standard SIP or Lync or Avaya

Encryption Type: Auto or Encrypted or Unencrypted (use Encrypted for Lync trunks)

Priority: order of priority of the rule (high priority is tried first)

Outbound calls

Filter [ ] Submit Query

	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption	Tenant	
	video.tkratzke.local	14.80.99.237	acanolab.tkratzke.local	<use local contact domain>	Standard SIP	Stop	10	Auto	no	[edit]
					Standard SIP	Stop	0	Auto		Add New Reset

1

Delete

Dial Transforms

	Type	Match Expression	Transform Expression	Priority	Action	
	Raw			0	Accept	Add New Reset

Delete

Local Dial Plan – Inbound Call

- Incoming Call webadmin configuration describes if incoming call need to be :
  - Handled locally (spaces, users)
  - Forwarded to an external destination (e.g. call from CUCM to Lync) via dial plan (Outbound Call) configuration

- (forward behaviour can be summarized as below:
  Incoming Call => Incoming Call rule (no match) => Forwarded Call rule (match) => handle call according to Outbound Call rule)

Incoming call handling

Call matching

☐	Domain name	Priority	Targets spaces	Targets users	Targets IVRs	Targets Lync	Targets Lync Simplejoin	Tenant	
☐	tptac9.com	30	yes	yes	yes	no	no	no	[edit]

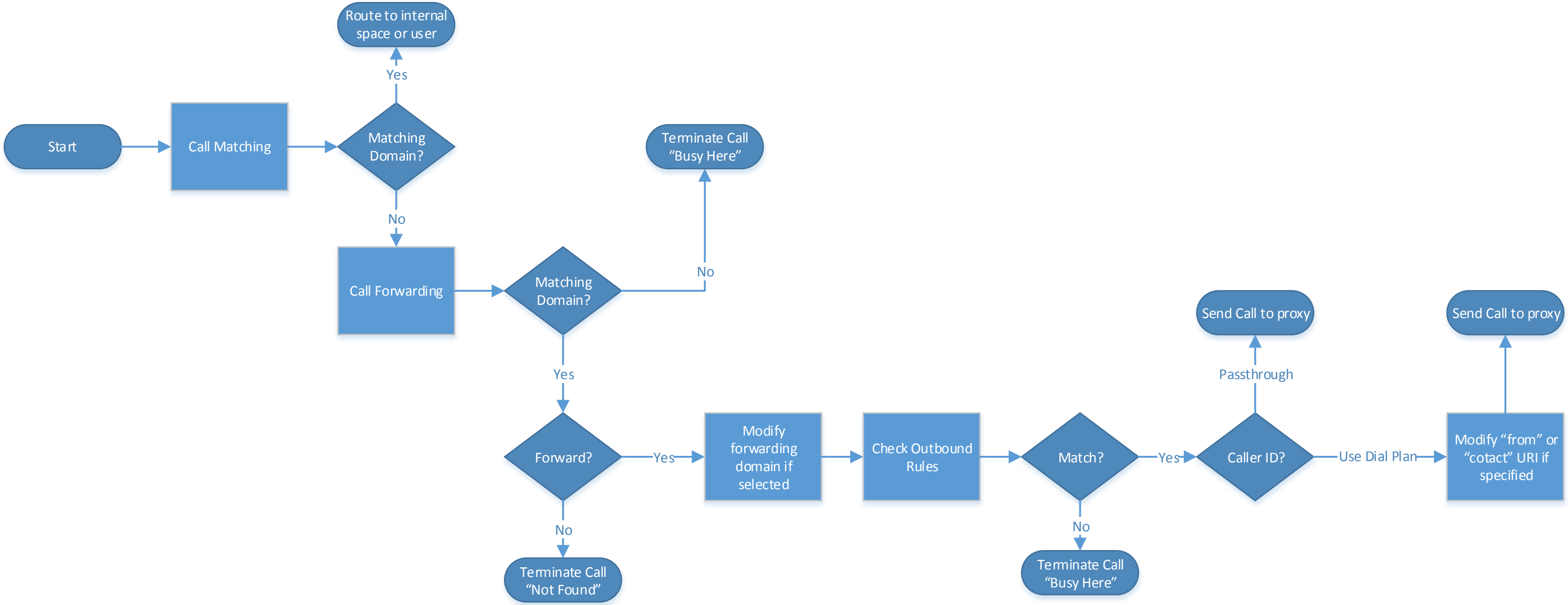
Call forwarding

☐	Domain matching pattern	Priority	Forward	Caller ID	Rewrite domain	Forwarding domain	
☐	cisco.com	0	reject	pass through	yes	ciscotac.in	[edit]
☐	tptac9.com	0	forward	use dial plan	no		[edit]
		0	reject	use dial plan	no		Add New Reset

Local Dial Plan

- Inbound call matching rules are specified on the “incoming call” page in the callbridge GUI.
- Domains can be specified under “call matching” and then routed to spaces or users (or both).
- If an inbound call does not match anything on this list, it will fall back to the “call forwarding” section. Here you can specify if you want to forward or reject a call based on domain.
- If forwarding, you can also transform the domain (this is useful if bridging a call to the Lync network).
- In the call forwarding table, wildcards can be used, and unlike VCS, **higher** numbered priority is tried first.
- Forwarded calls will then use the outbound call routing rules.
- If “use dialplan” is selected, all rules and outbound domains are respected. If “passthrough” is selected, no fields are changed from the source but the dial plan is still used for routing.
- If nothing matches, the call will be terminated.

Local Dial Plan (Cont'd)



Additional Features

Certificates

- Certificate are stored in the server root and they can be transferred with SFTP from any admin role user
- **pki** CLI/MMP is the main command
- selfsigned certificate are locally signed
- Recommend to use for lab environment
- selfsigned certs must not use for cluster deployment
- Certificate Authorities (CAs) are trustworthy authorities
- CA can be created locally to signed certificate
- Public CAs signed certificate can also be used

```
Usage:
    pki
    pki list
    pki inspect <file>
    pki match <key> <certificate>
    pki verify <cert> <CA file/cert bundle> [<CA file>]
    pki unlock <key>
    pki csr <key/cert basename> [<attribute>:<value>]
    pki selfsigned <key/cert basename>
    pki pkcs12-to-ssh <username>
```

Self signed certificates

- Generation:
- **pki selfsigned <key/cert basename>**

Example generation, list, verification
core1.pod6.tpbru3.tpuc.com> **pki selfsigned webadmin**
..........................+++
........+++
Created key file webadmin.key and selfsigned certificate webadmin.crt

core1.pod6.tpbru3.tpuc.com> **pki list** *(certificates are in server root)*
User supplied certificates and keys:
webadmin.key
webadmin.crt
core1.pod6.tpbru3.tpuc.com> **pki match webadmin.key webadmin.crt**
Matching certificate and private key

Certificate Authority signed Certificates

- Applications that interface internally within the Meeting Server only require certificates signed by an internal CA
- Internal CA signed certificates can be generated by a local or organizational Certificate Authority, such as an Active Directory server with the Active Directory Certificate Services Role installed
- The applications that require public CA signed certificates are
- Webbridge: If using webRTC, WebRTC clients require a public CA signed certificate from the Web Bridge in order to trust the connection.
- XMPP Server: Native Cisco Meeting App require a public CA signed certificate from the XMPP server in order to trust the connection.
- TURN server : If you configure TLS on your TURN server, then the TURN server will require a certificate/key pair similar to that created for the Web Bridge, so that the WebRTC client trusts the connection. The certificate should be signed by the same Certificate Authority as used for the Web Bridge certificate

CAs signed Certificates-continue

To generate the private key and Certificate Signing Request file:

- Type the “pki csr” command using this syntax
- pki csr <Key/Cert basename> <CN:value> <OU: value> [OU:<value>] [O:<value>] [ST:<value>] [C:<value>] [subjectAltName:<value>]
- <Key/Cert basename>: is a string identifying the new key and CSR. Can contain alphanumeric, hyphen or underscore characters.
- CN: This is the fully qualified domain name (FQDN) that specifies the server’s exact location in the Domain Name System (DNS)
- subjectAltName: From X509 Version 3 (RFC 2459), SSL certificates are allowed to specify multiple names that the certificate should match.
- If you plan to use the same certificate across multiple components, for example the Web Bridge, XMPP Server, Call Bridge and TURN server, then specify your domain name (DN) in the CN field, and in the SAN field specify your domain name (DN) and the FQDN for each of the components that will use the certificate.

CAs signed Certificates-continue

1. Generate a Certificate Sign Request
    1. Example *(note that Lync require CN == T rusted app pool name)*
    2. **pki csr cisco-global-cert CN:core1.pod6.tpbru3.tpuc.com O:"Cisco" OU :”TAC" L: :”BG" C :”IN" subjectAltName:core1.pod6.tpbru3.tpuc.com,edge1.pod6.tpbru3.tpuc.com,join.pod6.tpbru3.tpuc.com,pod6.tpbru3.tpuc.com**
2. Sign the certificate with CA
    1. Example with Window CA
    2. *DOS> **certreq -submit -attrib "CertificateTemplate:webserver***
3. Distribute (SFTP) signed certificate (and key if necessary) to servers
4. See Certificate Guide for detail

Questions?