Unified Contact Center Enterprise Solution Design Considerations

Core Components Design Considerations

General Solution Requirements

Data Backup for Your Solution

Run data backup tools only during a scheduled maintenance window. If you use local SQL backups, make sure that the local machine has sufficient capacity. If not, back up to remote storage on the network.

NTP and Time Synchronization

Finesse Time Synchronization:

While time drift occurs naturally, it is critical to configure NTP to keep solution components synchronized. Cisco Finesse server and the Desktop client machines should be time synchronized with the same NTP server (Linux-based NTP v4) for the Duration fields within the Live Data reports to be updated correctly.

Live Data Time Synchronization:

Contact center enterprise solutions require that all parts of the solution have the same time. To prevent time drifts on Live Data reports, the NTP settings on the followinge VMs must be synchronized:

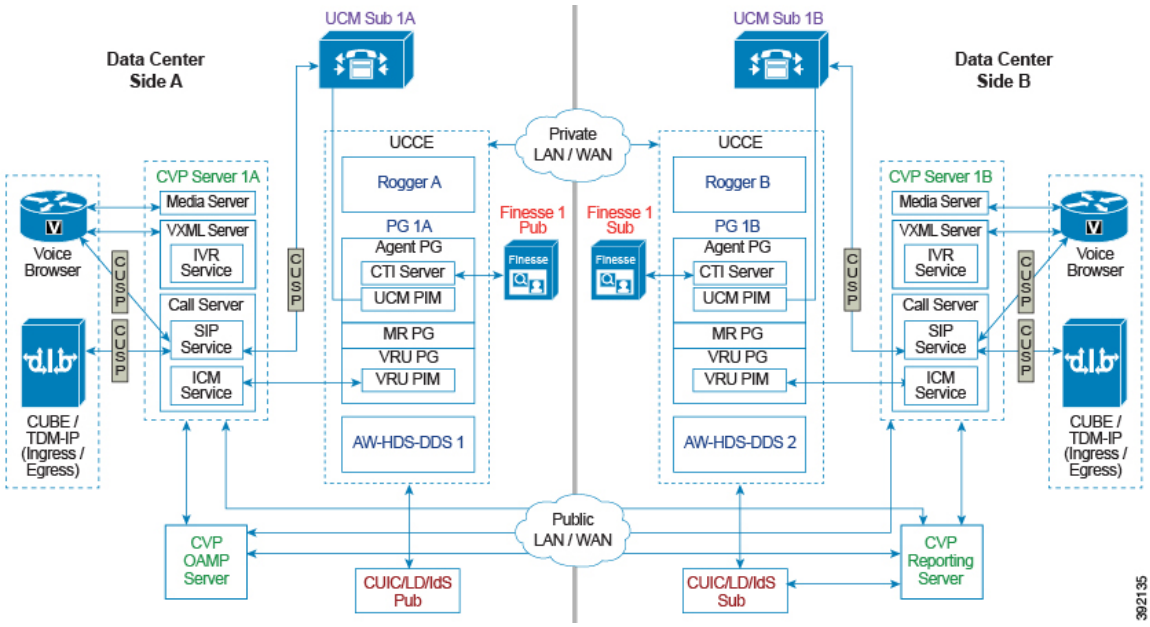
- Router
- Logger
- Administration & Data Server
- Unified Intelligence Center Publisher and Subscriber

Detailed Contact Center Enterprise Reference Design Topologies

Detailed 2000 Agent Reference Design

This figure shows the logical connections under normal operating conditions between the sides in a redundant data center.

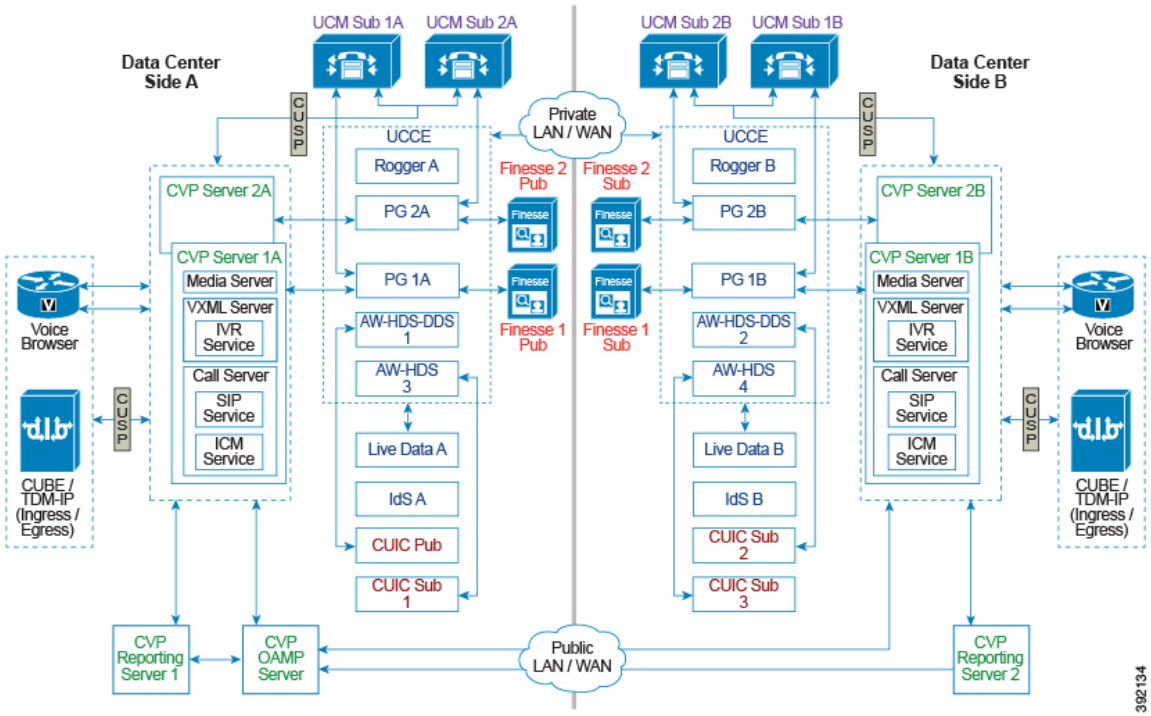
Figure 1: Detailed 2000 Agent Reference Design



Detailed 4000 Agent Reference Design

This figure shows the logical connections under normal operating conditions between the sides in a redundant data center.

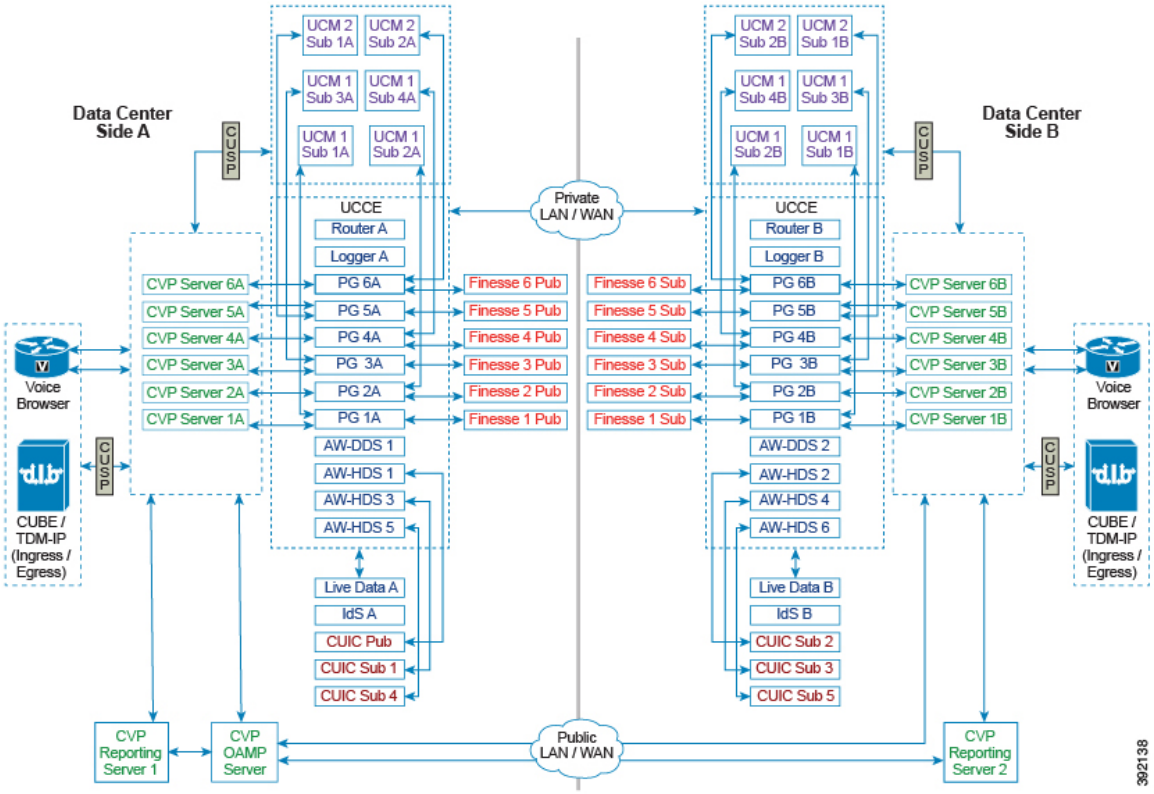
Figure 2: Detailed 4000 Agent Reference Design



Detailed 12000 Agent Reference Design

This figure shows the logical connections under normal operating conditions between the sides in a redundant data center.

Figure 3: Detailed 12000 Agent Reference Design



Ingress, Egress, and VXML Gateways Design Considerations

IOS Gateway Roles

The contact center enterprise solutions use IOS Gateways for TDM ingress and VXML rendering. You can usually use any Cisco gateway that the solutions support can for either purpose or both. This table lists which call flows use each function:

Table 1: IOS Gateway Function Use by Call Flows

Call Flows	TDM Ingress	VXML Rendering
Reference Design		
Comprehensive Using Unified ICM Micro-Apps	Yes	Some
Comprehensive Using Unified CVP VXML Server	Yes	Some
Non-Reference Design		
Standalone Self-Service	Yes	Yes

Call Flows	TDM Ingress	VXML Rendering
Call Director	Yes	No
VRU Only with NIC Controlled Routing	Yes	Yes

Note You can use Cisco Virtualized Voice Browser as an alternative for VXML gateways.

When both Ingress and VXML are required, you can run both functions on the same gateways or designate some gateways for ingress and others for VXML. Use the following guidelines to determine whether to combine or split the functions:

- In branch office deployments, where the call is queued at the branch where it arrived, always combine the ingress and VXML functions.
- Where many non-CVP PSTN connections share the gateways, use separate gateways for each function.
- VXML-only gateways are less costly because they do not require DSP farms or TDM cards.
- For low call volumes, you generally combine the functions for redundancy purposes. If one combined gateway fails, the other gateway can still process calls at a reduced capacity.

The next decision is whether to use Cisco Integrated Service Router (ISR) or ISR-G2 Gateways.

The classic branch office in which to use ISR Gateways, includes:

- One of several sites where TDM calls arrive from the PSTN
- A site that is separated from the main site where most of your solution's equipment resides
- Each site uses one gateway

Related Topics

Cisco Virtualized Voice Browser Design Considerations, on page 25

TDM-IP Gateway Design Considerations

For the most current information about the various digital (T1/E1) and analog interfaces supported by the various voice gateways, see the latest product documentation available at the following sites:

- **Routers**—http://www.cisco.com/cisco/web/psa/default.html?mode=prod&level0=268437899
- **Unified Communications Gateways**—http://www.cisco.com/cisco/web/psa/default.html?mode=prod&level0=278875240

Cisco Unified Border Element Design Considerations

The Cisco Unified Border Element (CUBE) is a session border controller (SBC) that provides connectivity between IP voice networks with SIP. Your solution can use physical CUBEs or virtual CUBEs. Your solution can only use CUBE in flow-through mode, where all calls are routed through CUBE.

Note Unlike flow-through mode, flow-around mode loses the ability to do DTMF interworking, transcoding, and other key functions, such as telephone and media capabilities.

Your solution needs a CUBE when replacing a TDM voice circuit with an IP voice trunk from a telephone company. CUBE serves as a feature demarcation point for connecting enterprises to service providers over IP voice trunks.

Note For outbound calls, physical CUBE supports Call Progress Analysis (CPA). Virtual CUBE does not support CPA.

Our testing shows that you can use CUBE in the following scenarios:

- SIP-to-SIP connectivity between a third-party SIP device and Unified CVP over Cisco-certified SIP trunks.
- SIP-to-SIP connectivity between Unified CM and Unified CVP.
- Coresidency of VoiceXML Gateway and CUBE for any of the above scenarios but with the limitation that the call flow does not work when the configurations listed here occur at the same time on the CUBE:
  - Survivability TCL script and incoming translation rules are configured under the same incoming dial-peer.
  - Header-passing between the call legs is enabled globally.

For more information about using the CUBE with contact center enterprise solutions, see *Cisco Unified Border Element for Contact Center Solutions* at http://cisco.com/en/US/docs/voice_ip_comm/unified_communications/cubecc.html.

Note For a listing of the maximum sessions that each CUBE supports, see the *Cisco Unified Border Element Configuration Guide* at http://www.cisco.com/c/en/us/support/routers/cloud-services-router-1000v-series/products-installation-and-configuration-guides-list.html.

Note Due to a limitation in Cisco IOS, the CUBE does not support midcall escalation or descalation from audio to video or the reverse.

Note Currently, CVP does not check the Allowed-Methods in the SIP message. As a result, it passes the `UPDATE` message from Ingress to Outbound leg although Outbound does not support `UPDATE` method.

Workaround: Disable the `UPDATE` message in SBC in the Ingress leg.

CUBE Deployment Restrictions

Note the following restrictions when deploying CUBE with SIP Trunks:

- Configure CUBE in media pass-through mode, the default mode on the dial-peer, in the Unified CVP deployment. Media flow-around mode is not supported or validated.
- CUBE does not support passing the Refer-To header URI designation from CVP when a REFER call flow is initiated. CUBE rewrites the destination address based on the dial peer configuration. Therefore, configure the dial plan on CVP and CUBE.
- You cannot use REFER passthrough with Survivability. The script does not let REFER messages be relayed to a SIP service provider regardless of other CUBE configuration.
- You cannot use REFER consume with Survivability and Router Requery. Survivability always accepts the REFER, even if the transfer does not complete. Unified CCE deems the transfer successful and does not attempt to requery.
- You cannot use survivability with service provider Alternate Destination Routing (ADR). Manipulation in the script does not let error messages (ring-no-answer or busy) reach the service provider. Use manipulation in the Remote-Party-ID header instead.
- If GTD is present on the incoming call or if Unified CCE sets a value for the UUI variable, Unified CVP sends a BYE immediately after outpulsing digits in a DTMF transfer. If you need a delay between the digits, use a comma at the end of the label.
- If GTD is not present on the incoming call, Unified CCE does not set a value for the UUI variable. Then, the service provider does not disconnect a call after receiving digits in a DTMF transfer. Unified CVP sends a BYE request after the SIP.ExternalTransferWait timer expires.
- Solutions with Courtesy Callback require survivability.

Note Call Survivability is supported on CUBE HA mode with the following restrictions:

- If there is a courtesy callback (CCB) registered with CVP, then post switchover CCB is not supported.
- Only call survivability TCL script is supported with CUBE high availability. Other TCL based services are not supported.
- Only the active calls will be check pointed. (Calls which are connected - 200OK / ACK transaction completed). Calls in transition state will not be check pointed.

Related Topics

Cisco Unified Border Element

Virtual CUBE for Contact Center Solutions

Cisco ASR 1000 Series as a Unified Border Element

Unified CVP supports Cisco IOS XE software with the following limitations:

- The ASR 1000 Series gateways do not support VXML. So, route the VRU leg of the call to a separate VXML Gateway. Do not use the `Send To Originator` setting on the CVP Call Server to route the VRU

leg of the callback to the originating ASR CUBE Gateway. Route standalone CVP calls to a separate VXML Gateway.

- Unified CVP does not support the global `Pass Thru SDP` setting on the ASR 1000 Series gateways.
- The Courtesy Callback call flow does not work if you configure the ASR as CUBE for the media flow-around, instead of the media flow-through.
- ASR 1000 Series gateways do not support TCP transport with SIP signaling when using the box-to-box hardware redundancy feature. UDP transport is supported when failing the active ASR chassis to the standby chassis. The default TCP setting does not work with failover. Because of this limitation, UCS VM deployments cannot support ASR box-to-box failover because CVP only supports TCP on the UCS Call Server.
- Typically, you position a proxy server behind the session border controller. If the proxy is in front of the ASR session border controller, use the proxy servers to perform UDP to TCP Up-Conversion when receiving large packet SIP messages. In this case, turn off the proxy servers to ensure that UDP transport is used for the connection on the inbound call.
- Do not use the following Survivability.tcl options on the ASR. These options are traditionally for POTS dial peers:
  - ani-dnis-split.
  - takeback-method.
  - -- *8.
  - -- hf.
  - icm-tbct.
  - digital-fxo.
- The following Survivability.tcl options are not supported:
  - aa-name—This option is not supported because ASR does not support the CME auto-attendant service.
  - standalone—This option is not supported because ASR does not support VXML.
  - standalone-isntime—This option is not supported because ASR does not support VXML.
- Due to ASR limitations, the following features are not supported:
  - Refer with Re-query
  - Legacy Transfer Connect using DTMF *8 label
- ASR 1000 does not terminate the TDM trunks. Therefore, the following TDM Gateway features do not apply to ASR 1000:
  - PSTN Gateway trunk and DS0 information for SIP calls to Unified CCE
  - Resource Availability Indication (RAI) of DS0 trunk resources through the SIP OPTIONS message to Unified CCE

Note If your solution uses ASR 1000 Series gateways, it requries an Assessment to Quality (A2Q) review. This review is required for new contact center enterprise solutions and existing solutions that are upgrading to the ASR 1000.

Cisco ISR as a Unified Border Element

Unified CVP supports ISR with the following limitations:

- The Courtesy Callback call flow does not work with ISR as CUBE configured for the media flow-around. Configure it for the media flow-through instead.

VXML Gateway Design Considerations

Note You can use Cisco Virtualized Voice Browser as an alternative for VXML gateways.

VXML Gateway with DTMF or ASR/TTS

The VXML Gateway allows customers to interact with the VXML browser through DTMF tones or ASR/TTS. Because the gateway does not have PSTN interfaces, voice traffic is sent using Real-Time Transport Protocol (RTP) to the VXML Gateway. The RFC 2833 uses in-band signaling in RTP packets to transmit DTMF tones. A VXML with DTMF or ASR and TTS allows you to increase the scale of the deployment and support hundreds of VXML sessions.

In a branch office topology, you can deploy a separate PSTN Gateway and a VXML Gateway to provide an extra layer of redundancy In addition, provide support for Survivable Remote Site Telephony (SRST) at the branch office.

VXML Over HTTP

The VXML Server and Voice browser communicate with request-response cycles using VXML over HTTP. Uniform Resource Identifiers (URI) link the VXML documents together. Users input information by web forms similar to HTML. The forms contain input fields that the user edited and sent back to a server.

Resources for the Voice browser are located on the VXML Server. These resources are VXML files, digital audio, instructions for speech recognition (Grammars), and scripts. The VXML browser begins every communication process with the Voice application as a request to the VXML Server. The VXML files contain grammars that specify expected words and phrases. A link contains the URL for the Voice application. The browser connects to that URL when it recovers a match between spoken input and one of the grammars.

Note The CVP installer installs the CVP Call Server, the CVP VXML Server, and the Media Server together.

The following points are key to determining the VXML Server performance:

- QoS and network bandwidth between the Web application server and the voice gateway
- Performance on the VXML Server
- Use of prerecorded audio versus Text-to-Speech (TTS)

Voice user-interface applications tend to use prerecorded audio files wherever possible. Recorded audio sounds better than TTS. Choose the quality of the prerecorded audio fileso that it does not impact download time and browser interpretation. Make the recordings in the 8-bit mu-law 8 kHz format.

- Audio file caching

  Ensure that the voice gateway is set to cache audio content. Caching prevents delays in downloading files from the media source.

- Use of Grammars

  You can discover problems in a voice application only through formal usability testing or observation of the application in use. Poor speech recognition accuracy is a common problem with voice applications. It is most often caused by poor grammar implementation. When users mispronounce words or say things that the grammar designer did not expect, the recognizer cannot match their input against the grammar. Another common problem for grammars is many difficult-to-distinguish entries. These entries result in many incorrectly recognized inputs and decrease the performance of the VXML Server. Improve the recognition accuracy by analysizing its performance and tuning the grammar appropriately.

Distributed Gateways

These sections discuss the types of voice gateways and their effects in a distributed deployment.

Related Topics

Enhanced Location Call Admission Control Feature

Ingress or Egress Voice Gateways at the Branch

Your solution can use Ingress Voice Gateways at a branch office to provide callers with access by local phone numbers, instead of by centralized or nongeographic numbers. This capability is important for solutions that span multiple countries.

Your solution can use Egress Gateways at branches either for a localized PSTN breakout or to integrated decentralized TDM platforms into your solution.

The other components of your solution are centrally located. The WAN links provide data connectivity from each branch location to the main site.

Ingress or VXML Gateway at the Branch

Other voice services that run at the branch can affect the Ingress or VXML Gateways. For example, if the branch is a remote UnifiedCM site, Unified CM can support both ACD agent lines and nonagent lines. This deployment uses the PSTN gateway for new contacts and traffic from the nonagent lines. When a branch has the VXML and Voice Gateway functions on separate devices, ensure that the dial plan sends the VRU leg to the local VXML resource. This is because the Unified CVP Call Server `settransferlabel` label applies only to coresident VXML and Voice Gateway configurations.

Sometimes, the Ingress Voice Gateway and the VXML Gateway at a branch do not reside on the same Gateway. You can handle contacts within the branch and avoid sending them across the WAN to a different VXML Gateway as follows:

- Configure Unified CCE with multiple customers, one Unified CCE configuration.

  The Unified CCE configuration differentiates between calls by the Dialed Number. The Dialed Number is associated with a customer representing the branch site. When a NetworkVRU is needed, you select the NetworkVRU for the customer in Unified CCE and send the caller to that NetworkVRU. This method

allows you to have multiple NetworkVRUs, each with a unique label. The disadvantage of this method is that each NetworkVRU requires its own VRU scripts in Unified CCE.

- Configure Unified CVP using the SigDigits feature.

  The SigDigits feature allows you to use the dial plan on the SIP Proxy to route calls to the correct site. When the call arrives at an Ingress Voice Gateway, the gateway prepends digits before sending the call to Unified CVP. Those prepended digits are unique to that site for a dial plan.

  When the call arrives at Unified CVP, Unified CVP strips and stores in memory the prepended digits. This results in the original DID on which the call arrived. Unified CVP then notifies Unified CCE of the call arrival using the original DID and matches a Dialed Number in Unified CCE.

  Unified CCE returns a label to Unified CVP to transfer the call to a VXML gateway for VRU treatment or to an agent phone. Unified CVP prepends the stored digits before the transfer. The dial plan in the SIP Proxy sends the calls with a certain prepended digit string to a specific VXML Gateway or Egress Gateway.

  When the VXML Gateway receives the call, the CVP bootstrap service strips the digits again. When the VRU leg of the call is set up, the original DN is used on the incoming VXML request.

Note You can prepend the digits to translation route DNs. The egress or receiving component (such as Unified CM) must strip the digits to see the original DN.

  You turn on this feature and specify the number of significant digits to strip with the **Prepend Digits for SIP** command in the operations console.

  This method involves the least amount of Unified CCE configuration overhead: a single NetworkVRU and a single set of VRU scripts and Unified CCE routing scripts. All of the Unified CVP Servers and VXML Gateways function as a single network-wide virtual VRU from the perspective of Unified CCE.

  You can also use the SigDigits feature to solve multicluster call admission control problems.

Colocated VXML Servers and VXML Gateways

Your solution can either have all gateways and servers centralized or have a set of colocated UnifiedCVP VXMLServers and VXML Gateways at each site.

Colocation has the following advantages:

- A WAN outage does not affect self-service applications.
- VXML uses no WAN bandwidth.

Colocation has the following disadvantages:

- Replicated branch offices require extra UnifiedCVP VXMLServers.
- Deploying applications to multiple UnifiedCVP VXMLServers creates more overhead.

Gateways at Branch with Centralized VXML Server

Advantages of centralized VXML:

- Administration and reporting are centralized.

- Branch offices can share UnifiedCVP VXMLServer capacity.

Disadvantages of centralized VXML:

- Branch survivability is limited.
- Requires more WAN bandwidth for VXML over HTTP traffic.

Local Trunks in Contact Center Enterprise Solutions

Contact center enterprise solutions have two options for local trunks at the customer premise:

- Cisco Unified Border Element—Enterprise at the customer premise
- TDM gateway at the customer premise

Note Transcoding resources are not deterministically picked from the local customer premise gateway.

CVP Design Considerations

CVP Call Server Design Considerations

Unified CVP Algorithm for Routing

When you set up a dial plan and call routing, you can combine Unified CVP features to achieve the required effect. For example, you can use Location Based CAC, SigDigits, SendToOriginator, LocalSRV, and Use Outbound Proxy.

CVP uses this process to formulate the destination SIP URI for the outbound calls from Unified CVP. This description covers CONNECT messages that include labels from the Unified CCE (for example, VXML Gateway, and Unified CM). It also applies for calls to the ringtone service, recording servers, and error message playback service.

Note This process only describes calls using the SIP subsystem, which includes audio only and basic video SIP calls.

CVP supports the `SendToOriginator` algorithm only for a colocated IOS VXML Gateway and Ingress Voice Gateway. Cisco Virtualized Voice Browser (VVB) does not support this algorithm because the gateways cannot be colocated when you use Cisco VVB.

The process for creating the destination SIP URI host portion for outbound calls, which includes the Unified CCE label, is as follows:

1. The process starts with the Unified CCE label. The Unified CCE subsystem might already have inserted the Location siteID. If you're using SigDigits, they are prepended. For network VRU labels, the Unified CCE subsystem passes in the entire prefix and correlation ID as the label.
2. If `SendtoOriginator` is matched for the Unified CCE label, the Unified CVP algorithm uses the IP or hostname of the caller (Ingress Voice Gateway). The gateway returns the SIP URI.

The setting for `SendtoOriginator` only applies to callers on Cisco Ingress Voice Gateways (the SIP UserAgent header is selected). Non-Cisco IOS Gateways do not have the CVP bootstrap service that the Cisco IOS VXML Gateway uses.

3. If `use outbound proxy` is set, then use the host of the proxy and return SIP URI.
4. If `local static route` is found for the label, return the SIP URI.

Note If `local static route` is not found, the algorithm throws a `RouteNotFoundException` exception.

Consider these points for calls using the SIP subsystem:

- To avoid complex Dialed Number strings, do not use the SigDigits feature with Locations CAC siteIDs.
- You can specify an Outbound Proxy FQDN as a Server Group FQDN (local SRV FQDN). You can also configure a local static route destination as a Server Group FQDN.
- Ringtone DN (91919191), Recording Server (93939393), and Error message services (92929292) follow the same process.
- `SendtoOriginator` can work with a REFER label.
- A REFER label can work with the SigDigits setting.

CVP VXML Server Design Considerations

The complexity of your VXML applications affect the performance of the VXML Server. Load test your application for memory leaks and application deadlocks to maintain acceptable VXML Server performance.

CVP Media Server Design Considerations

Voice Prompt Deployment and Management

You can deploy voice prompts with the following methods:

- Local File System

  Store the voice prompt files on a local system. Audio prompt retrieval uses no bandwidth. With this method, Voice Browsers do not have to retrieve audio files for playing prompts, so WAN bandwidth is not affected. However, to change a prompt, you change it on every Voice Browser.

  - **IOS VXML Gateway**—Prompts are deployed on flash memory.

    An IOS VXML Gateway is either a VXML Gateway or a PSTN Gateway, which has the Ingress Voice Gateway and VXML Gateway colocated. Store only critical prompts here, such as error messages or messages for when the WAN is down.

    When recorded in G.711 mu-law format, typical prompts are about 10 to 15 KB in size. For these gateways, size the flash memory by factoring in the number of prompts and their sizes. Also leave space for storing the Cisco IOS image.

  - **Cisco VVB**—Prompts are uploaded on the local file system.

    Cisco VVB includes built-in CVP prompts. You can change the `Error` tone default prompt through the **Cisco VVB Administrator console**.

- Media Server

  Each local Voice Browser, if configured properly, can cache many prompts, depending on the size of the prompts. Cisco VVB can cache up to 512 MB and Cisco IOS can cache up to 100 MB. To test whether your Media Server is appropriately serving the media files, specify the URL of a prompt on the Media Server in a browser. Your web browser downloads and plays the `.wav` file without any authentication.

The design of a Media Server deployment depends on the following factors:

- The number of media files that each gateway plays
- The network connectivity between the gateway and the Media Server
- How often you change the media files

Design Considerations for Large Numbers of Media Files

If your gateway plays many different media files to your customers, the gateway might not have space to cache all the media files.

For example, consider an enterprise with many agents. Each agent has their own agent greeting file. You cannot cache all those files in the gateway flash memory.

Colocated Media Server with Voice Browser

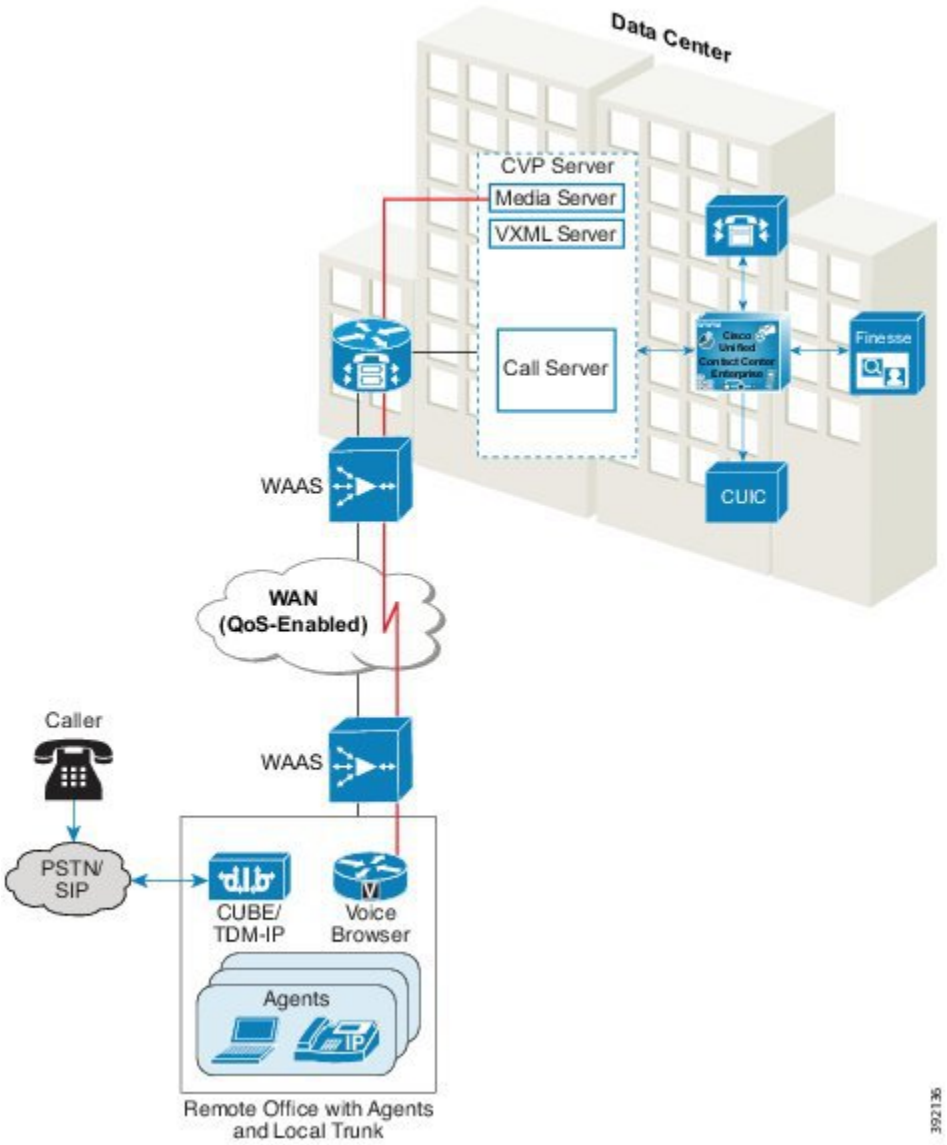
One approach to this problem is to colocate the Media Server with the Voice Browser. When a Media Server and Voice Browser coexist on a LAN with plenty of bandwidth, the download of prompts does not add noticeable delay.

Media Server and Voice Browser Distributed Over a WAN

Your solution can have a Media Server separated from a Voice Browser across a WAN.

This figure shows a distributed deployment over WAN.

Figure 4: Distributed Deployment Over WAN



The download of the media files across a high latency WAN to the Voice Browser can add noticeable delays. The delay can greatly affect the user experience. The delay is proportional to the size and number of media files that are transported across the WAN. You can optimize the delay with Cisco Wide Area Application Services (WAAS).

Design Considerations for Media Streaming

Consider the following factors for both the LAN deployment and the WAN accelerator deployment:

- Maximum network round-trip time (RTT) delays of 200 ms.

  An example is the transfer of files from the CVP Operations Console to the Ingress or the VXML Gateway using Bulk Administration File Transfer (BAFT).

- Maximum number of streaming sessions supported on each gateway with no additional overhead of video with media forking.

The following table lists the preferred media streaming method for various deployments:

Scenario	Frequency of Change	Over LAN	Over WAN
Small number of files	Rare	Cached	Cached
Small number of files	Often	Streamed or Cached	Streaming with WAAS
Large number of files	Rare	Streamed	Streaming with WAAS
Large number of files	Often	Streamed	Streaming with WAAS

Note Cisco VVB does not support the Media Streaming feature.

Design Considerations for Media File Deployment

No Support for TCP Socket Persistence

Unified CVP does not support TCP socket persistence.

WAN Acceleration Support

The Cisco Wide Area Application Services (WAAS) system is a set of devices called Wide Area Application Engines (WAEs). The WAEs work together to optimize TCP traffic over your network. Cisco WAAS uses TCP optimization techniques and application acceleration features to overcome the most common challenges in transporting traffic over a WAN. When deployed at the periphery of the network on the VXML Gateway side, Cisco WAAS performs these functions:

- Changes the TCP header to optimize the traffic.
- Acts as a large HTTP cache located locally.
- Uses compression algorithms to further reduce the traffic.
- Reduces traffic with Data Redundancy Elimination (DRE) techniques.

Cisco WAAS is deployed in inline mode where whole data is forced to pass through the Cisco WAAS.

Media File Deployment on IOS Gateway

Nonstreaming and Streaming Modes

In nonstreaming mode, the VXML gateway downloads the entire audio file from the HTTP server before the Media Player can start playing the prompt. This can cause a delay for the caller. For small files, the delay is only a few milliseconds. You can avoid the delay for larger files by using either caching or streaming mode.

In streaming mode, the Media Player streams the audio in media chunks from the HTTP server to the caller. The Media Player can start playing a prompt when it recieves the first chunk. In streaming mode, the size of the audio prompt does not add any delay for the caller. However, the back-and-forth interactions to fetch the media file in chunks can degrade performance.

Caching the audio files in memory reduces the advantage of streaming large files directly from the HTTP server.

Media File Cache Types

There are two types of cache for storing media files:

HTTP Client cache
In nonstreaming mode, the HTTP Client cache stores the entire media file. In streaming mode, the HTTP Client cache stores the first chunk of the media file. The HTTP Client cache stores 100 MB of prompts, in either mode. Any file that is larger than the configured HTTP Client memory file size is not cached.

VRU Media Player cache
Nonstreaming mode never uses the VRU Media Player cache. In streaming role, the VRU Media Player cache stores all chunks of the file. In nonstreaming mode, the VRU Media Player cache stores 16MB. In streaming mode, it can store 32 MB.

Query URL Caching

A query is a URL that has a question mark (?) followed by one or more **name=value** attribute pairs in it. The Unified CVP VXML Server uses query URLs when generating its dynamic VXML pages. Because each call is unique, data retrieved from a query URL can waste the cache memory. The data is also a possible security risk, because the query URL can contain information such as account numbers or PINs.

Cisco IOS disables Query URL caching by default. To ensure that it is disabled, enter a **show run** command in Cisco IOS and ensure that the following Cisco IOS command does not appear:

```
Gateway configuration: http client cache query
```

Media File Deployment on Cisco VVB

Cisco VVB includes an HTTP client. The client fetches VXML documents, audio files, and other file resources and stores them in flash memory.

A caching property is associated with VXML resources, audio prompts, grammar files, and script files.

By default, Query URLs are not cached. A query is a URL that has a question mark (?) followed by one or more **name=value** attribute pairs in it.

Cache Aging

The HTTP Client manages its cache by the freshness of each cached entry. Whether a cached entry is fresh or stale depends on its `Age` and `FreshTime`. `Age` is the elapsed time since the file was last downloaded from the server. `FreshTime` is the expected time for the file to stay in the HTTP Client cache since the file was last downloaded.

Several variables affect the `FreshTime` of a file, such as HTTP message headers from the server and the cache refresh value.

The `FreshTime` of a file is determined in the following sequence:

1. When downloaded, if the file has an HTTP message header with the `Cache Control: max-age` header, the `FreshTime` is the `max-age`.
2. If Step1 does not apply, the `FreshTime` is the `Expires` header minus the `Date` header.

Note The HTTP/1.1 specification, *RFC 2616 (HyperText Transport Protocol)*, recommends the use of either the `Cache Control: max-age` header or the `Expires` header.

3. If the previous headers are not present, the `FreshTime` is 10% of the `Date` header minus the `Last-Modified` header.

For the Cisco IOS VXML Gateway, you can assign a `FreshTime` value to the files with the `http client cache refresh` command. But, that value only applies if the previous sequence fails to set a value.

Stale files are refreshed only when needed. A stale cached entry stays in the cache until it is removed to make room for a new file, based on these conditions:

- The cached entry becomes stale.
- Its refresh count is zero (0); that is, the cached entry is not being used.
- The cache needs the memory space to make room for other entries.

When the Age exceeds the FreshTime and the file needs to be played, the HTTP Client checks with the media server to determine whether or not the file has been updated. When the HTTP Client sends a GET request to the server, it uses a conditional GET to minimize its impact on network traffic. The GET request includes an If-Modified-Since in the headers sent to the server. With this header, the server returns a 304 response code (Not Modified) or returns the entire file if the file was updated recently. When the Age exceeds the FreshTime and the file needs to be played, the HTTP Client checks with the media server to determine whether or not the file has been updated. When the HTTP Client sends a GET request to the server, it uses a conditional GET to minimize its impact on network traffic. The GET request includes an If-Modified-Since in the headers sent to the server. With this header, the server returns a 304 response code (Not Modified) or returns the entire file if the file was updated recently.

This conditional GET applies only to nonstreaming mode. In streaming mode, the HTTP Client always issues an unconditional GET. There is no If-Modified-Since header included in the GET request that results in an unconditional reload for each GET in streaming mode.

CVP Reporting Server Design Considerations

The CVP Reporting Server houses the Reporting Service and hosts an IBM Informix Dynamic Server (IDS) database management system. The database's schema is available to enable you to write custom reports for the database. The Reporting Service does not itself perform the database administrative and maintenance activities, such as backups or purges. However, Unified CVP provides access to such maintenance tasks through the Operations Console.

The Reporting Service:

- Provides historical reporting of self-service activity in your contact center. The service summarizes call activity for the contact center managers.
- Can also provide operational analysis of various VRU applications.
- Receives reporting data from the IVR Service of VXML server and the SIP Service. The Reporting Service transforms and writes this data into the Informix database.

Your solution can use either a single of multiple CVP Reporting Servers. A single Reporting Server does not necessarily represent a single point of failure. The database management system provides data safety and

security. Your solution can tolerate temporary outages due to persistent buffering of information on the source components.

If your solution uses multiple Reporting Servers, you can associate each CVP Call Server with only one Reporting Server. Also, your reports cannot span multiple Informix databases.

Note Unified CVP subcomponents cannot synchronize the machine time themselves. Provide a cross-component time synchronization feature, such as NTP, to assure accurate time stamps for logging and reporting.

CVP Reporting Server Features

Consider the following points when designing your solution with the CVP Reporting Server:

- You can size the Informix database up to 100 GB. You cannot use a 2 GB or smaller database in a production environment.
- The Reporting Server supports the Analysis Manager tool. The Analysis Manager can query the Reporting Server with an authenticated user's credentials.
- The Reporting Server aggregates Unified CVP data in 15-minute increments. Cisco Unified Intelligence Center provides templates to display call data and dominant path information at 15-minute, daily, and weekly intervals.
- All metadata for administrative processes is in the `Ciscoadmin` database. This locationlocation removes the tables from the normal view of reporting users.
- All database backup files are compressed and stored on the Reporting Server. The backup file is called **cvp_backup_data.gz** and is stored on the %INFORMIXBACKUP% drive in the **cvp_db_backup** folder.
- Using the system CLI, you can make the request to list log files on the Reporting Server (**show log**). This request includes the Informix Database Server Engine logs. The **show tech-support** command also includes these files.
- With the debug level 3 (or 0) command from within the System CLI, you can turn on and turn off the debug. When turned on, this command generates trace files for all administrative procedures, Purge, Statistics, and Aggregator.

Note After the command is turned on, trace files place an elevated burden on the database.

- Log data for administrative procedures are written on a nightly basis to the **%CVP_HOME%\logs** folder.
- All the **StartDateTime**, **EndDateTime**, and **EventDateTime** values are stored as UTC in Reporting Server tables.
- Transfer Type data and Transfer Labels for SIP call events are stored in the call event table.
- Summary purge results are logged in the log table.
- Three new scheduled tasks have been added to the Reporting Server scheduler:
  - **CVPSummary**, which builds summary tables.

- **CVPCallArchive**, which archives Callback data to maintain callback database performance.
- **CVPLogDump**, which extracts the administrative logs on a nightly basis.

CVP Backup and Restore

Using the Operations Console, you can schedule daily database backups or run database backups on-demand. In a major failure, you can restore the database manually to the last backup time. This limits the loss of data to 24 hours at most.

CVP Operations Console Server Design Considerations

Operations Console is a web-based interface to configure and monitor Unified CVP subcomponents. Your solution can have only one Operations Console. You can manage the following Unified CVP subcomponents from the Operations Console:

- Call Server
- VXMLServer
- Reporting Server

Note Operations Console is also referred to as the OAMP (Operate, Administer, Maintain, and Provision). The Operations Console manages individual components through the Unified CVP Resource Manager, which is collocated with each managed Unified CVP component. The Resource Manager is invisible to the user.

For details on Operations Console, see the *Configuration Guide for Cisco Unified Customer Voice Portal* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-and-configuration-guides-list.html.

CVP Call Studio Design Considerations

When you design applications in CVP Call Studio, keep the applications small and closely mapped to your business flows. Large applications are harder to maintain and work with. Maintain a balance between subflows and independent applications.

Unified CVP Coresidency

To calculate the number of servers required with SIP call control, use this formula:

```
(Self Service + Queue and Collect + Talking) / 3000, rounded up
```

Where:

Self Service is the number of calls that require SIP call control and run an application on the VXML Server.

Queue and Collect is the number of calls that require SIP call control and run an application using Microapps only on the Call Server.

Talking is the number of calls at agents.

The following example applies for VXML and HTTP sessions only.

```
((3000) + (500) + 3700) / 3000 = 3 servers
```

If you use CUBE as a Session Border Controller (SBC) for flow-through calls to handle VXML requirements, use the sizing information provided in the example.

If you use CUBE as a Session Border Controller (SBC) to handle flow-through calls only (no VXML), then consider Voice Activity Detection (VAD) and see the sizing information in the *Cisco Unified Border Element Ordering Guide*, available at http://www.cisco.com/c/en/us/products/collateral/unified-communications/unified-border-element/order_guide_c07_462222.html.

Contact Center Enterprise Design Considerations

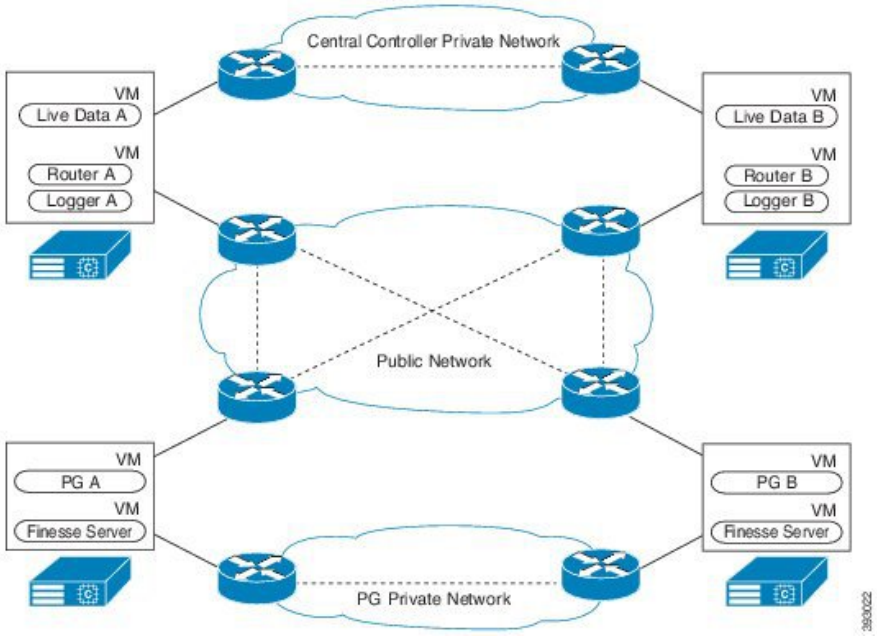
The contact center enterprise software provides enterprise-wide distribution of multichannel contacts. It can support inbound and outbound phone calls, web collaboration requests, email messages, and chat requests. It can also support geographically separated contact centers. The contact center enterprise software is an open standards-based solution that includes routing, queuing, monitoring, and fault tolerance capabilities.

Note Unified CCE is the name for both one of the contact center enterprise solutions and one of the core components for all of the solutions.

Router Design Considerations

You can geographically distribute redundant Unified CCE servers or locate them at the same physical site. In a production deployment, the Router and Logger must connect over a private network that is isolated from the public network.

Figure 5: High Availability Design for Central Controller



Note You can use the same private network path for the Central Controller and PGs.

Logger Design Considerations

The design of the Logger database holds two weeks of data usually. This period allows enough time for the data to replicate to the AW-HDS-DDS.

The Logger uses the same private network path as its Router.

Peripheral Gateway Design Considerations

Agent Peripheral Gateway Design Considerations

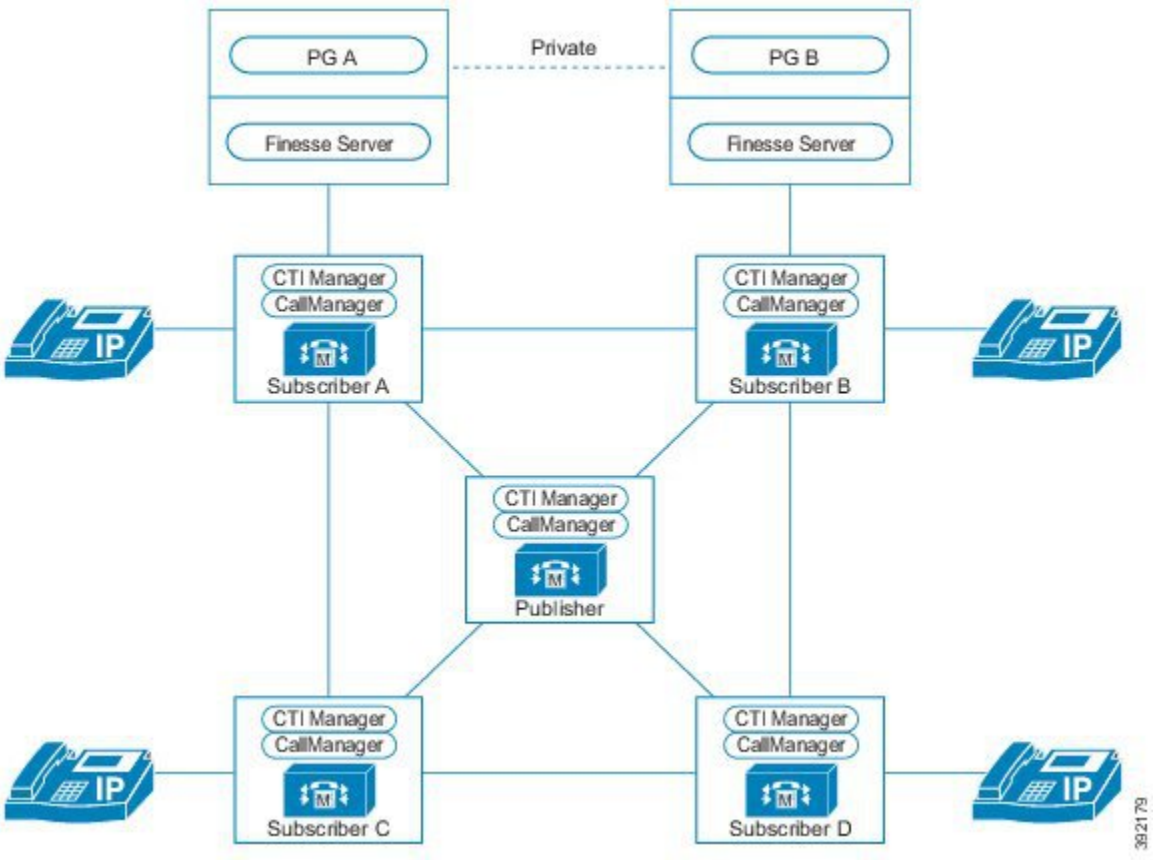
The Agent PG communicates with the Unified CM cluster through the CTI Manager. An Agent PG can control agent phones and CTI route points anywhere in the cluster. The Agent PG registers with the CTI Manager on a Unified CM subscriber in the cluster. The CTI Manager accepts all JTAPI requests from the PG for the cluster. When the PG requests a phone or route point on another subscriber, the CTI Manager forwards the request to the other subscriber.

Note The *Agent PG* is the PG that includes the Unified CM PIM. It is sometimes called a Unified CM PG. In Non-Reference Designs only, the Agent PG might be a Generic PG.

A fault-tolerant design deploys Agent PGs in a redundant configuration, because a PG only connects to the cluster through a single CTI Manager. If that CTI Manager fails, the PG cannot communicate with the cluster. A redundant PG provides a secondary pathway through a different CTI Manager on a different subscriber in the cluster.

The minimum design for a high-availability cluster is one publisher and two subscribers. If the primary subscriber fails, the devices rehome to the secondary subscriber and not to the publisher for the cluster.

Figure 6: High Availability Design for Unified CM Cluster



The redundant PGs keep in synchronization through a private network that is isolated from the public network. If the two PG servers are geographically distributed, use a separate WAN connection for the private network. To avoid a single point of failure in the network, do not use the same circuits or network gear as for the public network.

Within the Agent PG, the JTAPI Gateway and Unified CM PIM manage the connectivity to the cluster. The JTAPI Gateway handles the JTAPI socket connection protocol and messaging between the PIM and the CTI Manager. The PIM manages the interface between Unified CCE, the JTAPI Gateway, and the cluster. It requests specific objects to monitor and handle route requests from the cluster. The PG starts the JTAPI Gateway and PIM automatically as node-managed processes. The PG monitors the processes and automatically restarts them if they fail.

The JTAPI services from both redundant Agent PGs sign in to the CTI Manager after initialization. Agent PG-A signs in to the primary CTI Manager; Agent PG-B signs in to the secondary CTI Manager. Only one PG in each pair actively registers and monitors phones and CTI route points. The redundant PG runs in hot-standby mode. The redundant PG signs into the secondary CTI Manager only to initialize the interface and make it available for a failover. This arrangement significantly decreases the time for the failover.

When the system starts, the PG that first connects to the Router server and requests configuration information is the active PG. The Router ensures that the PG with the best connection becomes active. The nominal designations of “Side A” and “Side B” do not affect which PG becomes active. During a PG failover caused by a private link failure, a weighting mechanism chooses which PG is active to minimize the impact on the contact center.

If calls arrive at the CTI Route Points before the PIM is operational, the calls fail unless you set up a recovery number. Place the recovery number in the route points' `Call Forward on Unregistered` or `Call Forward on Failure` setting. For example, you can set the recovery numbers to the Cisco Unity voicemail system for the Auto Attendant.

Note You cannot use the DN for a CTI Route Point on a different CTI Route Point in another partition. Ensure that DNs are unique across all CTI Route Points on all partitions.

Active PG Shutdowns

Avoid shutting down an active peripheral gateway service in your production environment. This causes a service interruption of a minute or more while the other side connects and activates. The length of the interruption depends on the size of the configuration and the type of peripheral. For example, the VRU peripheral usually takes less time. The other side for the VRU might take 30 seconds or less to reactivate.

Voice Response Unit Peripheral Gateway Design Considerations

In the standard three PG model, the VRU PG includes two PIMs with a 1:1 pairing to a CVP servers on Side A and Side B.

Media Resource Peripheral Gateway Design Considerations

In the standard three PG model, the MR PG includes PIMs to support these functions:

- Outbound Option
- Enterprise Chat and Email
- SocialMiner—This PIM handles Task Routing and Agent Request.
- Third-party integrations

Administration & Data Server Design Considerations

Administration & Data Server Limits by Reference Design

You can deploy only so many Administration & Data Servers for each Logger.

Table 2: Administration & Data Server Deployment Limits Per Logger

Component on each Logger side	2000 Agent	4000 Agent	12,000 Agent
AW-HDS-DDS	1 per side, installed on the same server with the core components.	2 per side	NA
HDS-DDS	NA	NA	1 per side
AW-HDS	Optionally, either 1 AW-HDS or AW-HDS-DDS per side, installed on a separate server from the core components	NA	3 per side

Component on each Logger side	2000 Agent	4000 Agent	12,000 Agent
Real-Time Distributors only[1]	2 per side	2 per side	5 per side

[1] These AWs are for configuration only. You install them off the servers shown in the Reference Design layouts.

Note Each Real-Time Distributor can support 64 users.

Live Data Server Design Considerations

Reporting Clients by Live Data Configuration

Use the Live Data coresident configuration for 2000 Agent Reference Designs. For solutions with a standalone Live Data server, you typically use the small Live Data deployment configuration with a Unified CCE Rogger deployment. You typically use the large Live Data deployment configuration with a separate Router and Logger.

Note The standard Cisco Finesse agent desktop includes the Live Data gadget.

Cisco Virtualized Voice Browser Design Considerations

The number of Virtualized Voice Browsers that your solution requires depends on the VRU ports that your solution needs on the VXML Gateway. Install Cisco VVB depending on the number of SIP sessions required for your solution. This table lists feature support by Cisco VVB:

Platform or Feature	Cisco Virtualized Voice Browser Considerations
Voice Codec	G711
Call Flows	Standalone and Comprehensive with call survivability are supported.
ASR/TTS	Supported
Courtesy Callback	Supported
HTTP	Supported
HTTPS	Supported
Local Prompts	Supported
Local Hostname Resolution	Supported
MRCP v1 and v2	Supported
VXML 2.0 and 2.1	Supported

Platform or Feature	Cisco Virtualized Voice Browser Considerations
RTSP Streaming	Not Supported
Video	Not Supported

Unified Communications Manager Design Considerations

Note The Reference Design layouts use a 7500 user Unified CM OVA which supports 2000 contact center enterprise agents on each redundant pair of subscribers. If you use a different Unified CM OVA, move the cluster off the servers in the Reference Design layout or comply with the specification-based hardware policy. See the *Cisco Collaboration Virtualization* page for your solution at http://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/cisco-collaboration-virtualization.html for details on the specification-based policy.

Cisco Unified Communications Manager (Unified CM) connects calls passed from Unified CVP to the agent that Unified CCE chose. Unified CVP transfers callers to Unified CCE agent phones or desktops using SIP. The Unified CVP Call Server receives an agent label from Unified CCE and routes the call using SIP proxy. The call is then sent to the appropriate Unified CM subscriber in the cluster, which connects the caller to the agent. The Call Server proxies the call signaling, so it remains in the call signaling path after the transfer is completed. However, the RTP stream flows directly from the originating gateway to the phone.

All contact center enterprise solutions use redundant Unified CM, Unified CCE, and Unified CVP components. Because of the redundancy, your solution can lose half of its core systems and be still operational. In that state, a solution handles calls by rerouting them through Unified CVP to a VRU session or an agent on the still-operational components. Where possible, deploy Unified CCE so that no devices, call processing, or CTI Manager services run on the Unified CM publisher.

To enable automatic failover and recovery, pairs of redundant components interconnect over private network paths. The components use TCP heartbeat messages for failure detection. Unified CM uses a cluster design for failover and recovery. Each cluster contains a Unified CM publisher and multiple UM subscribers. Agent phones and computers register with a primary target but automatically reregister with a backup target if the primary fails.

To set up a Unified CM cluster for your contact center enterprise solution:

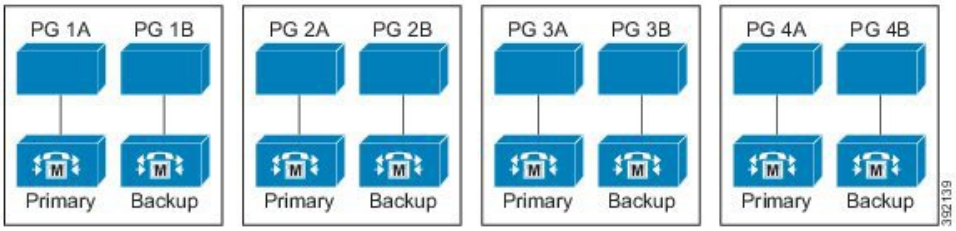
- Configure a SIP trunk in Unified CM.
- When configuring agent labels, consider which device is the routing client. When the label is returned directly to Unified CM, Unified CM is the routing client. When the label is sent to Unified CVP, associate the labels with each of the Unified CVP Switch leg Call Servers.

Unified CM Connection to the Agent PG

You can deploy Agent PGs to connect to a Unified Communications Manager cluster in the following ways:

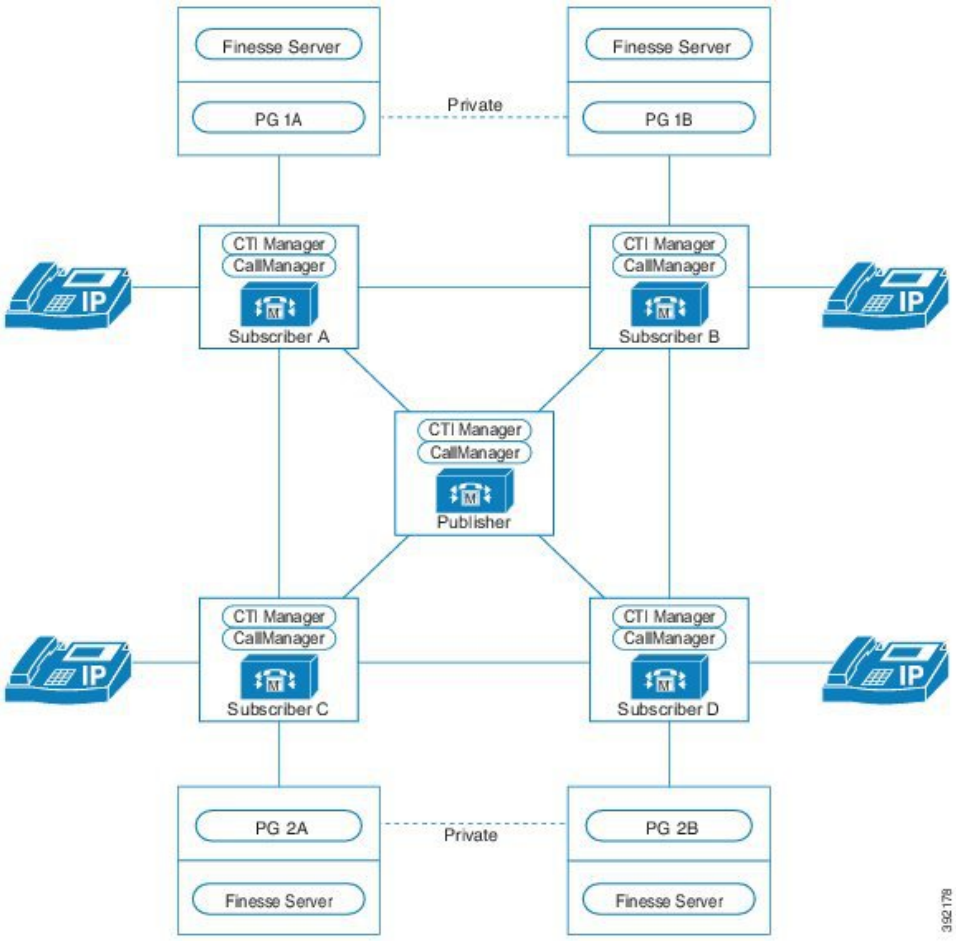
- Deploy an Agent PG for each pair of subscribers. Each subscriber runs the CTI Manager service. Each Agent PG connects to a CTI Manager running on its corresponding subscriber pair. This figure shows an example where four primary subscribers are required and four backup subscribers are deployed to provide 1:1 redundancy.

Figure 7: Deploy Agent PG for Each Pair of Subscribers in a Cluster



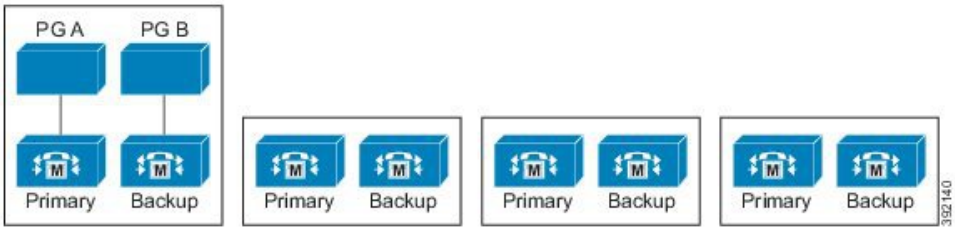
This figure shows the connections between the components in a solution with two Agent PG pairs and a cluster with four subscribers.

Figure 8: 2 Agent PG Pairs for Unified CM Cluster



- Deploy a single Agent PG for the entire cluster. This type of deployment requires a single pair of subscribers running CTI Manager. Spread agent phone registration among all the subscribers, including the subscribers running the CTI Manager service. The following diagram shows an example where four primary subscribers are required and four backup subscribers are deployed to provide 1:1 redundancy.

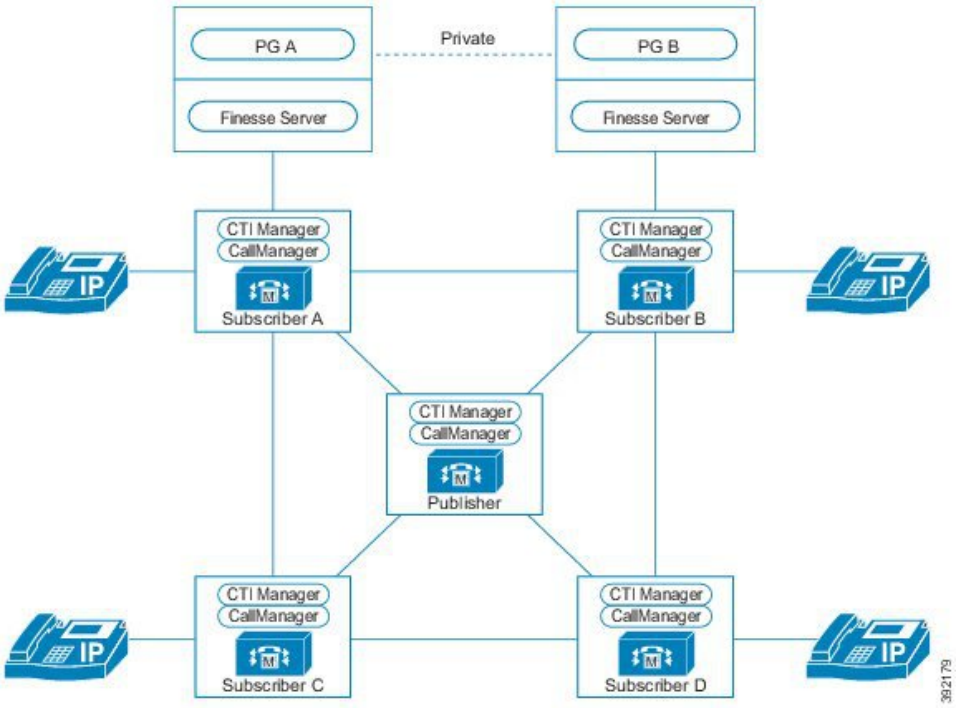
Figure 9: Deploy Single Agent PG for Entire Cluster



Note Use this option when your cluster supports both back-office phones and agent phones.

This figure shows the connections between the components in a solution with two Agent PG pairs and a cluster with four subscribers.

Figure 10: Single Agent PG Pair for Entire Unified CM Cluster



This model reduces the server count for the PG. Another benefit is that there is a single PIM for the entire cluster. So, you can create teams that span across many subscribers. This allows supervisors, for example, to monitor agent phones registered to any subscriber in the cluster. However, this deployment can have slightly higher resource usage on the cluster. Use the *Cisco Unified Communications Manager Capacity Tool* to size the Unified CM servers for your solution.

MTP Usage on the Unified CM Trunk

If your solution uses the Unified CM SIP Trunk, certain call flows, such as Cisco Unity Voice Mail or Mobile Agent, might require an MTP resource.

This is necessary when the negotiated media capabilities of the endpoints do not match, such as with the DTMF in-band versus out-of-band capability. In this case, Unified CM dynamically allocates an MTP due to the DTMF media capabilities mismatch.

Your solution might also require MTPs when interoperating with third–party devices.

Mobile and Remote Access

The Cisco Collaboration Edge architecture includes Unified Communications Mobile and Remote Access (MRA) to enable access by devices that are not in the enterprise network. MRA uses Expressway to provide secure firewall traversal and support for Unified CM registrations. Unified CM can then provide supported devices with call control, provisioning, messaging, and presence services.

For details on Collaboration Edge, see the documentation at http://www.cisco.com/c/en/us/support/unified-communications/unified-communications-system/tsd-products-support-series-home.html. For details on Expressway deployment and configuration, see the documentation at http://www.cisco.com/c/en/us/support/unified-communications/expressway-series/tsd-products-support-series-home.html. See the *Compatability Matrix* for your solution for details of device support for MRA.

If your solution uses MRA, consider these points:

- The connection between the Cisco Finesse client and server is over a VPN, not over the MRA connection.
- Certain phones do not support Extension Mobility over MRA.
- Contact center enterprise video deployments do not support MRA.
- If you have VPN split-tunneling configured, you can use Jabber with MRA and the Finesse desktop on the same client machine. See https://www.cisco.com/c/en/us/support/security/anyconnect-secure-mobility-client/products-installation-and-configuration-guides-list.html for Cisco AnyConnect Mobility Client Split-Tunneling configuration.
- If VPN split-tunneling is not available, you can run after splitting them onto two clients.
    - A remote agent who runs Jabber with MRA on one client machine and the Finesse desktop with a VPN connection on a second client machine.
    - A remote agent who runs a Jabber softphone on a laptop that is connected over MRA and runs the Finesse desktop as a Xenapp thin client.

Cisco Finesse Design Considerations

Cisco Finesse is a supervisor and agent desktop for use with contact center enterprise solutions. You install the Cisco Finesse server on a VM. Clients then use a web browser to point to the Cisco Finesse server. No Cisco Finesse software is installed on the client, which speeds and simplifies installation and upgrade.

See the *Contact Center Enterprise Compatibility Matrix* for supported browsers and operating systems for Cisco Finesse clients (administration console, agent desktop, and supervisor desktop).

The Cisco Finesse desktop application consists of the client and server components. The client uses standard web programming elements (HTML, JavaScript) that are distributed as gadgets using the OpenSocial 1.0

specification. You can configure the agent desktop to use Cisco and third-party gadgets through a layout management mechanism.

Cisco Finesse is part of a class of applications called Enterprise Mashups. An Enterprise Mashup is a web-centric method of combining applications on the client side. The gadget-based architecture of Cisco Finesse enables client-side mashup and easier integration. There are fewer version compatibility dependencies because gadget upgrades are handled independently.

You can customize the agent and supervisor desktops through the Cisco Finesse administration console. Administrators can define the tab names that appear on the desktops and configure which gadgets appear on each tab.

This table summarizes the capabilities of Cisco Finesse:

Table 3: Desktop Features

Desktop Functionality	Cisco Finesse
Browser-based desktops	Yes
Custom development	Yes (using standard web components such as HTML, JavaScript)
Desktop security	Yes **Note** Cisco Finesse supports HTTPS for up to 2000 agents on each PG pair.
Workflow automation	Yes
Mobile (remote) agents	Yes
Silent monitoring	Yes
Monitor mode applications	NA
Outbound calls	Yes
Microsoft Terminal Services support	NA
Citrix presentation server support	NA
Agent mobility	Yes
Agent Greeting	Yes

Cisco Finesse REST API

Cisco Finesse provides a REST API for client applications to access the supported server features. The REST API transports XML payloads over HTTP or HTTPS.

Cisco Finesse also provides a JavaScript library and sample gadget code to aid third-party integration. You can find the developer documentation for the REST API, the JavaScript library, and sample gadgets on the Cisco Developer Network at https://developer.cisco.com/site/finesse/.

Cisco Finesse Agent Desktop

Out of the box, the agent desktop provides the following features:

- Basic call control (answer, hold, retrieve, end, and make a call)
- Advanced call control (consultation, transfer after consult, conference after consult)
- Single-step transfer (agents can transfer a call without first initiating a consultation call)
- Queue statistics gadget (to view information about the queues to which the agent is assigned)
- View of the agent's Call History and State History
- Not Ready and Sign Out reason codes
- Contact lists
- Workflows
- Mobile agent support
- Progressive, Predictive, Preview Outbound, and Direct Preview Outbound

Cisco Finesse Supervisor Desktop

The Cisco Finesse supervisor features extend the agent desktop with more supervisor-only gadgets. These features include the following:

- Team performance gadget to view the agent status
- Queue statistics gadget to view queue (skill group) statistics for the supervisor's queues
- View of the supervisor's Call History and State History
- View the Call History and State History of any agent in the supervisor's team
-
- Unified CM Silent Monitoring
- Barge-in
- Intercept
- Change agent state (A supervisor can sign out an agent, force an agent into Not Ready state, or force an agent into Ready state.)

Cisco Finesse IP Phone Agent

With Cisco Finesse IP Phone Agent (IPPA), agents can access Cisco Finesse capabilities on their Cisco IP Phone instead of through the browser. Cisco Finesse IPPA only provides a subset of Cisco Finesse features that are available on the browser. It enables agents and supervisors to receive and manage Cisco Finesse calls without access to a PC.

Note Supervisors can only perform agent tasks on their IP Phones. Cisco Finesse IPPA does not support supervisor tasks, such as monitor, barge, and intercept.

Cisco Finesse IPPA supports the following functionality:

- Sign in and out
- Call variables display
- Pending state
- Wrap-up reasons
- Optional wrap-up
- Not Ready reasons
- State change using reason codes
- One Button Sign In

Cisco Finesse Administration Console

Cisco Finesse includes an administrative application that allows administrators to configure the following:

- Connections to the CTI server and the Administration & Data Server database
- Cluster settings for VOS replication
- Not ready and sign out reason codes
- Wrap-up reasons
- Contact lists
- Workflows and workflow actions
- Call variable and ECC variable layouts
- Desktop layout
- Team resources
- Cisco Finesse IP Phone Agent (IPPA)
- Context Service

Reason codes, wrap-up reasons, contact lists, workflows, and desktop layouts can be global (apply to all agents) or assigned to specific teams.

Cisco Finesse Deployment Considerations

Cisco Finesse and the Multiline Feature

Cisco Finesse supports the configuration of multiple lines on agent phones if Unified CCE is configured for multiline. You can configure several secondary lines on an agent phone. However, the Cisco Finesse server blocks any events that the CTI server sends in response to operations on secondary lines. The Cisco Finesse server does not publish these events to the Cisco Finesse clients. Information about calls on secondary lines does not appear on the Cisco Finesse desktop.

If your agents use 8900 Series or 9900 Series phones, enable Multi-Line on the Unified CM peripheral. This configuration option is a peripheral-wide option. If you enable Multi-Line for even one agent with an 8900 Series or 9900 Series phone, enable it for all agents.

To support multiline, configure all phones with the following settings:

- Set Maximum number of calls to 2.
- Set Busy trigger to 1.

Cisco Finesse with Citrix

Contact center enterprise solutions support running the Cisco Finesse desktop within a Citrix environment. Cisco Finesse supports Citrix XenApp and XenDesktop.

Note AWs, Configuration-Only Administration Servers, and Administration Clients can operate only as a single remote instance on a given VM.

For more information about supported versions, see the *Compatibility Matrix* for your solution at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-device-support-tables-list.html.

Cisco Finesse with NAT and Firewalls

Some solutions have two or more disjointed networks that interconnect with Network Address Translation (NAT).

Cisco Finesse provides limited support for NAT. Cisco Finesse supports basic NAT (one-to-one IP address mapping) between the Cisco Finesse servers and clients.

The following caveats apply to Cisco Finesse and NAT:

- You cannot use PAT/NPAT (one-to-many address mapping that uses ports) between the Cisco Finesse servers and clients.
- You cannot use NAT between the Cisco Finesse servers and any of the servers to which they connect (such as Unified CCE or Unified CM servers).
- Cisco Finesse IP Phone Agent (IPPA) does not support NAT.

Note For more information about NAT and firewalls, see the chapter on solution security.

IP Phone and IP Communicator Support

Cisco Finesse supports the use of Cisco IP hardware phones and the Cisco IP Communicator software phone.

For more information about the supported phone models and IP Communicator versions, see the *Compatibility Matrix* for your solution at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-device-support-tables-list.html.

IP Phones and Silent Monitoring

Silent monitoring supports both IP hardware phones and Cisco IP Communicator.

IP Phones and Mobile Agent

The Mobile Agent feature does not require any specific type of phone. You can even use analog phones with this feature.

IP Phones and Citrix or MTS

Cisco Finesse supports both IP hardware phones and Cisco IP Communicator when using Citrix or MTS.

In these environments, install Cisco IP Communicator on the agent desktop PC. You cannot deploy Cisco IP Communicator on the Citrix or MTS server.

Cisco Finesse and Cisco Jabber

Cisco Finesse supports Cisco Jabber for Windows as a contact center enterprise voice endpoint. Cisco Finesse supports the following Jabber functionality:

- Voice and Video
- Built-In Bridge (BIB) for silent monitoring
- IM and Presence

Note Agents cannot use Jabber to transfer or conference calls. Agents must use the Cisco Finesse desktop for transfer and conference.

To use Jabber with Cisco Finesse, change the default Jabber configuration as follows:

- Change Maximum number of calls from 6 to 2.
- Change Busy trigger from 2 to 1.

For more information on support for Jabber, see the *Compatibility Matrix* for your solution at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-device-support-tables-list.html.

Cisco Unified Intelligence Center Design Considerations

Unified Intelligence Center Deployments

A Unified Intelligence Center deployment consists of the following:

- One or more Unified Intelligence Center reporting (member) nodes in a cluster
- Real-time and historical data sources
- Live Data sources
- Other optional data sources

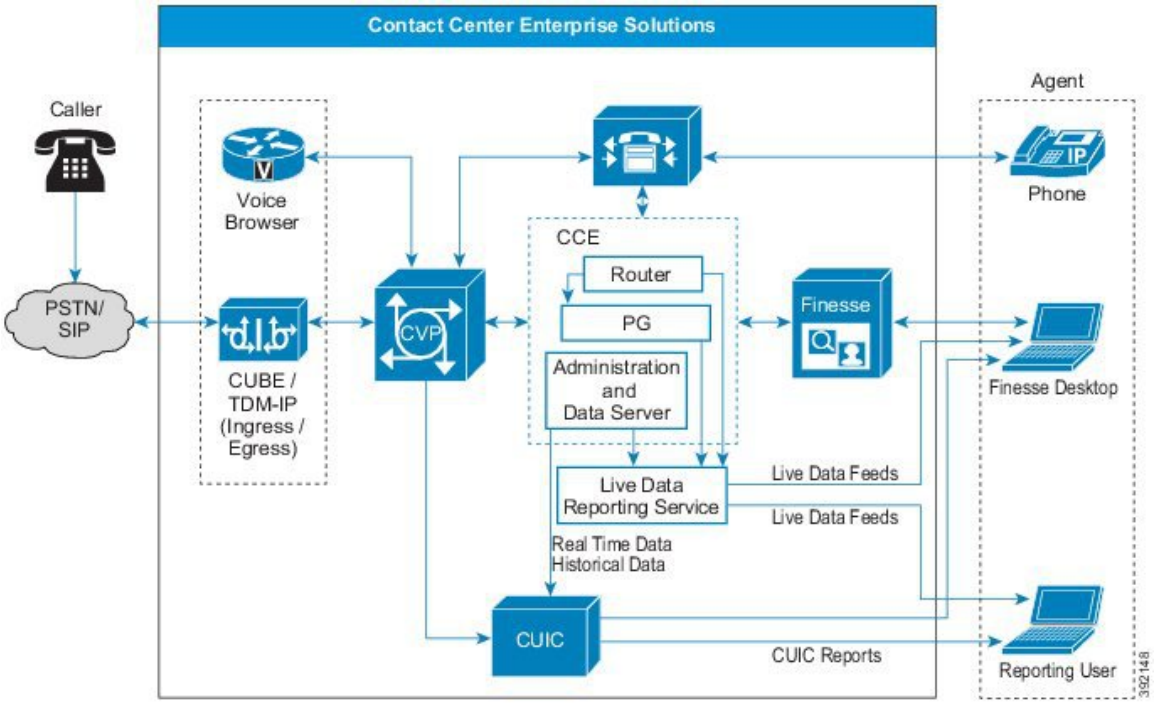
Note Ensure that the Unified Intelligence Center and the data source servers are in synch with the same NTP server.

Unified Intelligence Center nodes are deployed in standalone VMs in contact center enterprise solutions. Unified Intelligence Center supports Historical, Real-time, and Live Data reports.

The data flow for a historical or real-time report runs like this:

1. The web client makes an HTTPS request for a Unified Intelligence Center historical or real-time report.
2. The web server on the Unified Intelligence Center reporting node receives the request.
3. The reporting node pulls the data for the report from the data source server.
4. The reporting node sends the report to the web client through the web server.

Figure 11: Unified Intelligence Center Deployment



The client updates Live Data reports from a Live Data event stream from the Live Data service. For more information on Live Data control and data flows, see the *Serviceability Guide for Cisco Unified ICM/Contact Center Enterprise* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-installation-and-configuration-guides-list.html.

Unified Intelligence Center Reporting Node

The reporting node is the core of Unified Intelligence Center and contains all the features for reporting. You can deploy reporting nodes in the following configurations:

- A standalone reporting node on a controller node

- A controller node with up to seven reporting member nodes in a cluster

A reporting node includes the following applications:

- A firewall
- A web server that runs the Unified Intelligence Center application
- JAVA services and JSP pages that translate the web requests into HTML
- A Unified Intelligence Center Database (Informix) with replication support within the cluster
- The Administration (OAMP) application (on the publisher node)

Unified Intelligence Center Database (Informix)

Each reporting node includes the Unified Intelligence Center application database. The Unified Intelligence Center database is the main data store for the Unified Intelligence Center reporting web application. It holds configuration information relating to users, reports, and user access rights for each node in the cluster.

In a Unified Intelligence Center cluster, each database server connects to all other database servers. Data immediately replicates from any server to all other servers.

An automated daily purge runs at midnight and handless database maintenance activities. You can change the purge schedule as needed. Purge and backup are the only local database maintenance tasks for local Unified Intelligence Center databases.

Unified Intelligence Center Data Sources

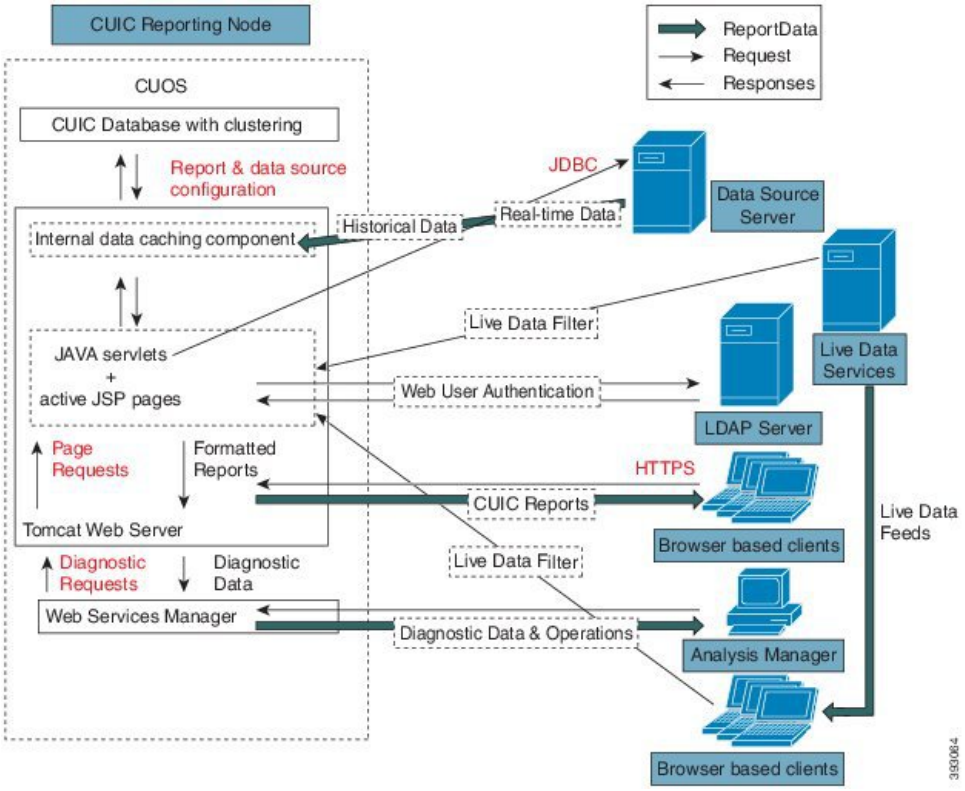
Unified Intelligence Center supports these data sources:

- Microsoft SQL-based or Informix data sources for Real-time and Historical reports. A SQL-based data source is a valid JDBC-compliant database and the schema that store the reporting data.
- Streaming data sources for Live Data reports

These data source servers are supported for these reports:

- Contact center enterprise reports, including those displayed in Cisco Finesse gadgets
- Importable Unified CVP reports
- SocialMiner reports
- Enterprise Chat and Email reports

Figure 12: Unified Intelligence Center Architecture



Live Data with Unified Intelligence Center

Unified CCE publishes real-time updates in agent, skill group, and calltype states through WebSockets. Unified Intelligence Center reports consume these messages directly from Live Data Services and display the updates in real time.

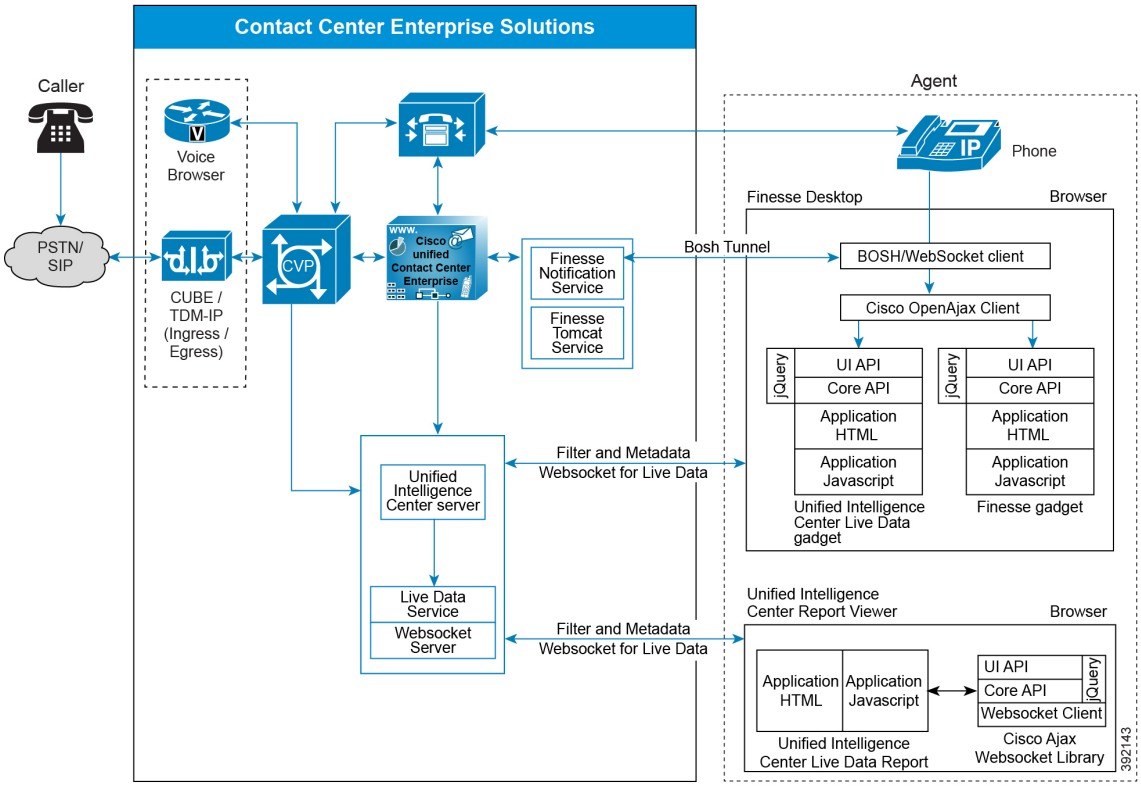
In Cisco HCS for Contact Center, the Live Data Service resides on its own VM.

Live Data reports can run in the Unified Intelligence Center Report Viewer. Cisco Finesse desktops can show Live Data reports in gadgets. They use a WebSocket tunnel from the Cisco Finesse desktop parent container to one of the Live Data Services. The gadget creates the tunnel during loading.

If a WebSocket connection fails, the Live Data reports automatically fail over to the currently connected Live Data node.

This figure shows the architecture for Live Data reporting in a contact center enterprise deployment:

Figure 13: Live Data Reports in Contact Center Enterprise



Administration & Data Server as the Data Source

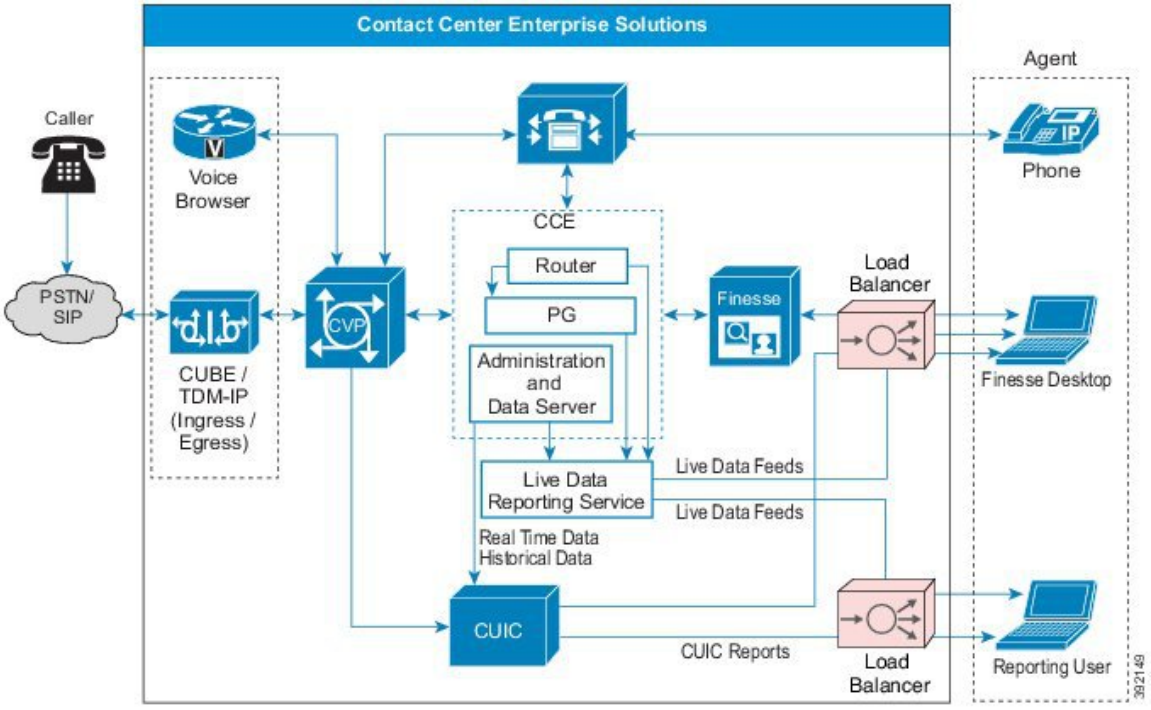
The Unified Intelligence Center stock reports use the Administration & Data Server as their data source. A contact center enterprise deployment can have multiple Administration & Data Servers. Unified Intelligence Center uses the database and its views as the tables for the data source queries.

The Unified Intelligence Center installation adds two data sources to the reporting (member) nodes:

- The historical data source, which supplies data for the historical reports and user integration
- The realtime data source, which supplies data for the realtime reports

Your deployment can use the same AW-HDS for both data sources, or you can configure a different server for each data source. Unified Intelligence Center requires an AW-HDS as a data source for standard historical reports. But, it can also use an AW as a data source for standard real-time reports. Unified Intelligence Center requires an AW-HDS-DDS or HDS-DDS as the data source for custom reports on TCD records.

Figure 14: Unified Intelligence Center Deployment with Unified CCE for Historical and Real-Time Reporting



Unified CVP as the Data Source

For deployments that import reports from Unified CVP, Unified Intelligence Center uses the CVP Reporting Server as the data source.

The CVP Reporting Server houses the Reporting Service and hosts an IBM Informix Dynamic Server (IDS) database management system. The database's schema is available to enable you to write custom reports for the database. Your solution can include several CVP Reporting Servers.

Unified Intelligence Center only connects to the CVP Reporting Server. The CVP Reporting Server mediates between the other Unified CVP subcomponents and Unified Intelligence Center.

Unified Intelligence Center in WAN Deployments

You can distribute the Unified Intelligence Center cluster over the WAN. Each node in a cluster requires a connection to every other node. In a cluster, either over a LAN or WAN, the configuration objects that are created on one node automatically replicate to the other nodes. The replication uses bandwidth across the WAN. But, since you create configuration objects infrequently, this affects the WAN bandwidth less often than running reports.

An object is instantly available users on the local node. It can take a few seconds before the object replicates to other nodes. The WAN bandwidth for replication depends on the configuration of the cluster.

Site Organization

A Unified Intelligence Center cluster can have a maximum of eight nodes. To have fully redundant clustering, each site is at most four nodes. As qualified, each Unified Intelligence Center supports up to 200 reporting

users under the standard reporting profile. Although you can have up to eight nodes in a cluster, you cannot exceed the Reference Design limit for the maximum reporting users.

The primary (controller) node is at the primary site. The primary node hosts the following services:

- Administration application
- Scheduler

WAN Failures

Each Unified Intelligence Center node buffers replication data to send to other nodes in the cluster. During a connection loss, the node queues the data until contact with the other nodes is restored. Each node continues to work independently during a connectivity failure. The queue holds up to 1600 MB. If connectivity is restored before the buffer exhaustion, the node synchronizes at a rate proportional to the amount of queued data and the connection bandwidth.

When the node nears buffer exhaustion, it sends an alarm. If connectivity is not restored before the buffer exhaustion, then replication is reset. By resetting replication, the node can continue to run reports and work independently. A secondary node that resets replication requires full synchronization with the primary node (primary database backup and restore on secondary node) after reconnecting with the primary node. If replication is reset, then all created or modified objects on the secondary node are rolled-back to the state of the primary database.

If the primary node fails, reinstall and revert to a saved backup. Back up the primary node periodically to avoid data loss.

For more information on data replication, see the *Administration Console User Guide for Cisco Unified Intelligence Center*.

Unified Intelligence Center Administration

The Unified Intelligence Center Administration server provides operations, administration, maintenance, and provisioning (OAMP) functions. The Administration server is the primary interface for configuring and provisioning devices in a Unified Intelligence Center cluster. You deploy and access the administration functions on the primary (controller) node in the cluster.

Cisco Unified Intelligence Center uses Hazelcast for application clustering. Hazelcast provides a second-level cache for the Unified Intelligence Center application layer. When any entity (for example: report, report definition, and so on) cached by Hazelcast is updated in one of the Unified Intelligence Center nodes, it must be invalidated and reloaded in all the other Unified Intelligence Center nodes in the cluster. The Hazelcast cluster automatically takes care of it by publishing clusterwide notifications containing the identifiers of such entities which must be invalidated.

In Unified Intelligence Center, the default mechanism for Hazelcast cluster discovery or formation is UDP multicast. Unified Intelligence Center uses the Multicast group IP address 224.2.2.3 and port 54327. You cannot change these settings in Unified Intelligence Center.

The UDP multicast based discovery mechanism will not work for the customer in the following scenarios:

- When the network has multicasting disabled.
- If the nodes in the Unified Intelligence Center cluster are in different subnets.

In such scenarios, you can change the discovery mechanism to TCP/IP. You can form the CUIC application cluster using TCP/IP instead of the default UDP Multicast based discovery mechanism.

For more information on the administration functions or on cluster configuration using Hazelcast, see the *Administration Console User Guide for Cisco Unified Intelligence Center* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-intelligence-center/products-maintenance-guides-list.html.

Note You do not use the Administration application for system-level functions such as network options, certificates, upgrades, and SNMP and Alert settings.

Throttling for Historical and Real-Time Reports

The Unified Intelligence Center throttling mechanism prevents servers from freezing or encountering an Out-of-Memory situation under extreme load.

Note Do not use the throttling mechanism for any sizing purposes. The throttling mechanism only prevents an Out-of-Memory situation. Throttling does not ensure a good quality of service. If your deployment is overused, the level of service can degrade substantially before the throttling mechanism activates.

Always use the sizing calculator to determine the proper reporting resources for your solution.

Processing report data consumes the most memory in Unified Intelligence Center. The throttling mechanism controls memory consumption due to reporting activity.

Unified Intelligence Center measures reporting activity through the *report row*. This measure of reporting activity gives you flexibility. You can run a few large reports or many small ones and the throttling mechanism is equally effective without requiring any tuning.

Unified Intelligence Center measures reporting activity through the *report row*. This measure of reporting activity gives you flexibility. You can run a few large reports or many small ones and the throttling mechanism is equally effective without requiring any tuning.

Tests with the stock reports show that 2 KB is a conservative estimate for the size of a report row. Based on that estimate, a Unified Intelligence Center server can load a maximum of 250,000 report rows into memory before the server runs out of memory.

To enforce this limit, each Unified Intelligence Center keeps count of the report rows currently loaded into memory. All reporting operations check that count to determine if they can load more report rows into memory. When you reach the limit, reporting operations fail and display an error as follows:

- **Violations while fetching data from the data source**—The report execution cancels and marks the report as failed. Unified Intelligence Center does not take partial results. The system either reads all the data for a request or marks the report as failed and stores none of the data.
- **Violations while preparing an HTTPS response for a browser**—Unified Intelligence Center rejects the display request. An error message says that the server is low on resources and cannot render the report.

Reference Design and Topology Design Considerations

Reference Design Model Considerations

12000 Agent Reference Design Considerations for Cisco HCS for Contact Center

Consider these points in your Cisco HCS for Contact Center 12000 Agent Reference Design solution.

Components	Design Considerations
CUIC	A maximum of 6 CUIC nodes are supported (3 nodes on each side) accommodating 1200 Reporting users. If one side completely fails, then only 3 CUIC nodes are available supporting up to 600 reporting users.
AW-HDS	A maximum of 6 AW-HDS nodes are supported (3 nodes on each side) accommodating 1200 Reporting users. If one side completely fails, then only 3 CUIC nodes are available supporting up to 600 reporting users.
CVP	The solution supports a maximum of 10000 VRU calls.

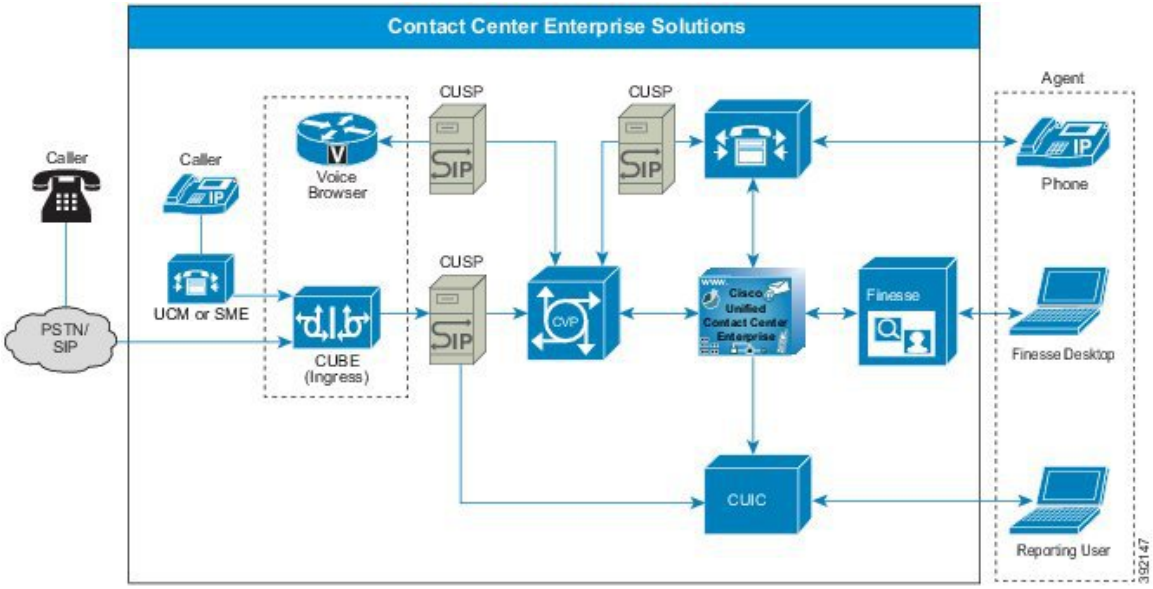
Unified CM SME Deployment

Cisco Unified Communications Manager Session Management Edition (Unified CM SME) integrates with Unified CVP as a dial peer configurator or aggregator to connect to multiple Unified CM clusters in a contact center enterprise solution.

Configure Unified CM SME as a back-to-back SIP user agent. As an aggregator of multiple Unified CM clusters, it routes the call to the appropriate cluster based on the dial plan.

This figure shows the Unified CM SME deployment.

Figure 15: Unified CM SME Deployment



Unified CM SME does not support high availability and is a single point of failure. Consider these design points to minimize the effect of network connectivity or component failures in Unified CM SME.

- Deploy Unified CM SME in redundant clustered mode (at least 1+1 publisher subscriber) at the egress side of Unified CVP.
- Configure **Session Refresh** and **Session Timer** in the Gateway and CUBE. This configuration clears call sessions from the gateway and releases Unified CVP Call Server ports if there is a Unified CM SME failure.
- In a Unified CM SME failure, all call server ports are cleared after the customer drops the call.

Note Call supplementary services do not work for the already established calls once the Unified CM SME fails.

A momentary network connectivity failure to Unified CM SME results in the following limitations:

- Unified CM SME does not clear the call if the agent hangs up during a momentary connectivity failure. This results in a stale cached entry and ports hanging in the Unified CVP application. In such cases, the caller should drop the call to clear the stale cached entry.
- The call does not get cleared from the agent desktop and the agent cannot receive any incoming calls. The agent remains in the talking state and cannot clear the call from the desktop. In such cases, manually clear the call from the phone.
- Because of a delay in call clearance, the call reporting data can reflect inaccurate details for call duration and reason code.

For more information about Unified CM SME configuration, see *Configuration Guide for Cisco Unified Customer Voice Portal* available at: http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-and-configuration-guides-list.html.

Global Deployments Considerations

- The maximum Round Trip Time (RTT) between the main site and a remote site is restricted to 400 milliseconds.
- The maximum RTT between the main site components and the customer premise is restricted to 200 milliseconds.
- The maximum RTT between the Side A components and Side B site is restricted to 80 milliseconds.

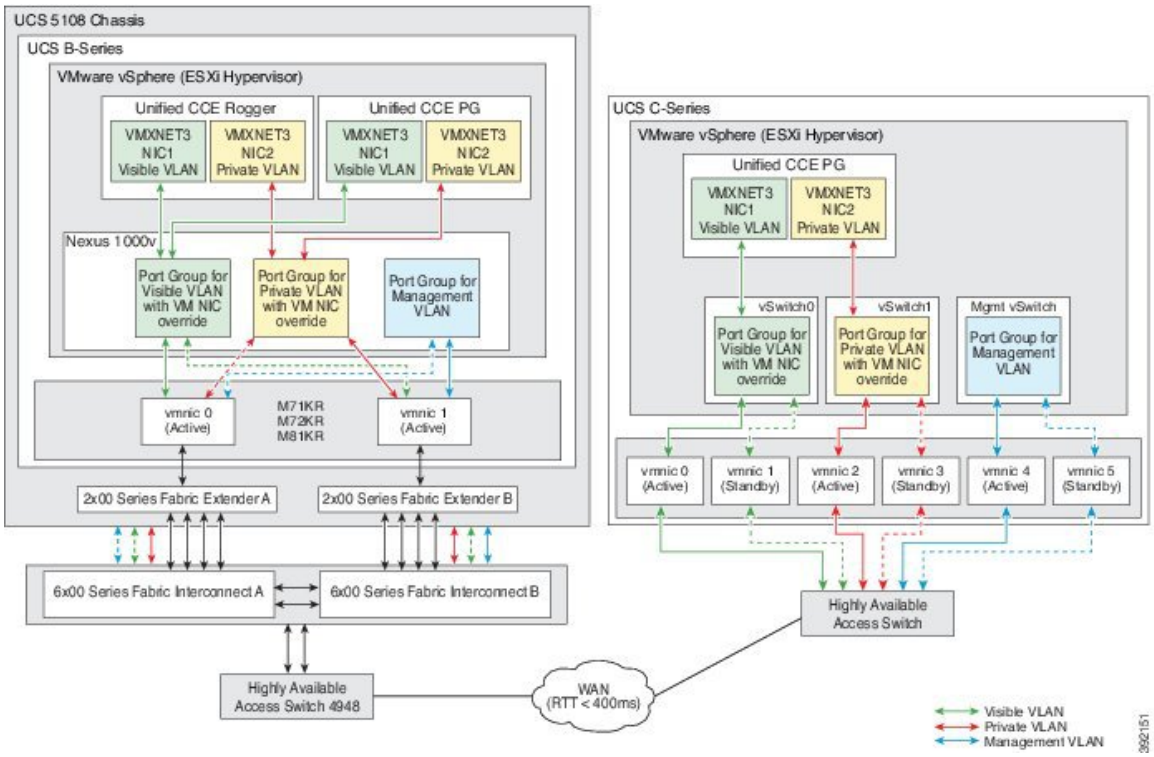
Note If you use Precision Queues, you can have a maximum of 12 Unified CM PIMs combined between the central and remote site.

- Use the hostname for CVP Media Servers and configure the IOS gateways to point to the local CVP servers.

UCS Network Design for Global Deployments

This figure shows the default design for a global deployment to meet Public and Private Network communications requirements. The Main Site uses UCS B Series blades and the **Remote Site** uses a UCS C-series server.

Figure 16: UCS Network Reference Design for Global Deployment



Call Survivability in Distributed Deployments

Distributed deployments require design guidelines for other voice services at the branch. For example, take a branch that is a remote Unified CM site supporting both ACD agent and nonagent phones. In this deployment, the PSTN Gateway handles not only ingress calls for Unified CVP. It also handles ingress or egress calls for the regular non-ACD phone.

Branch reliability in WANs may be an issue in a centralized Unified CVP model because they are typically less reliable than LAN links. The call survivability function must be considered for both the Unified CVP and non-CVP calls. For Unified CM endpoint phones, survivability is accomplished by using a Cisco IOS feature known as Survivable Remote Site Telephony (SRST).

For Unified CVP calls, a combination of services from a TCL script (survivability.tcl) and SRST functions handle call survivability. The survivability TCL script monitors the SIP connection for all calls that ingress through the remote gateway. If a signaling failure occurs, the TCL script takes control of the call and redirects it to a configurable destination. The destination choices for the TCL script are configured as parameters in the Cisco IOS Gateway configuration.

Note When the called number is in "E164" format, the survivability script removes the "+" sign from the called number before forwarding it to Unified CVP. Unified CVP or ICM does not support the "+" sign in the beginning of DNIS.

Alternate destinations for this transfer include another IP destination (including the SRST call agent at the remote site), call restart, call restart with a new destination, *8 TNT, or hookflash. With transfers to the SRST call agent at the remote site, the most common target is an SRST alias or a basic ACD hunt group.

Voice mail and recording servers do not send Real-Time Control Protocol (RTCP) packets in reverse direction toward the caller (TDM Voice Gateway). This can falsely trigger the media inactivity timer of the survivability script. It is important to apply the survivability.tcl script carefully to the dial peers. A call might drop if it goes to the voice mail or to a recording element. One method is to use a separate dial peer for voice mail or recording calls, and not associate the Unified CVP survivability script for those dial peers. Another method is to disable the media inactivity on the survivability script associated with the voice mail or recording dial peers.

For further information on configuration and application of these transfer methods, see the *Configuration Guide for Cisco Unified Customer Voice Portal* at http://www.cisco.com/c/en/us/support/customer-collaboration/unified-customer-voice-portal/products-installation-and-configuration-guides-list.html.

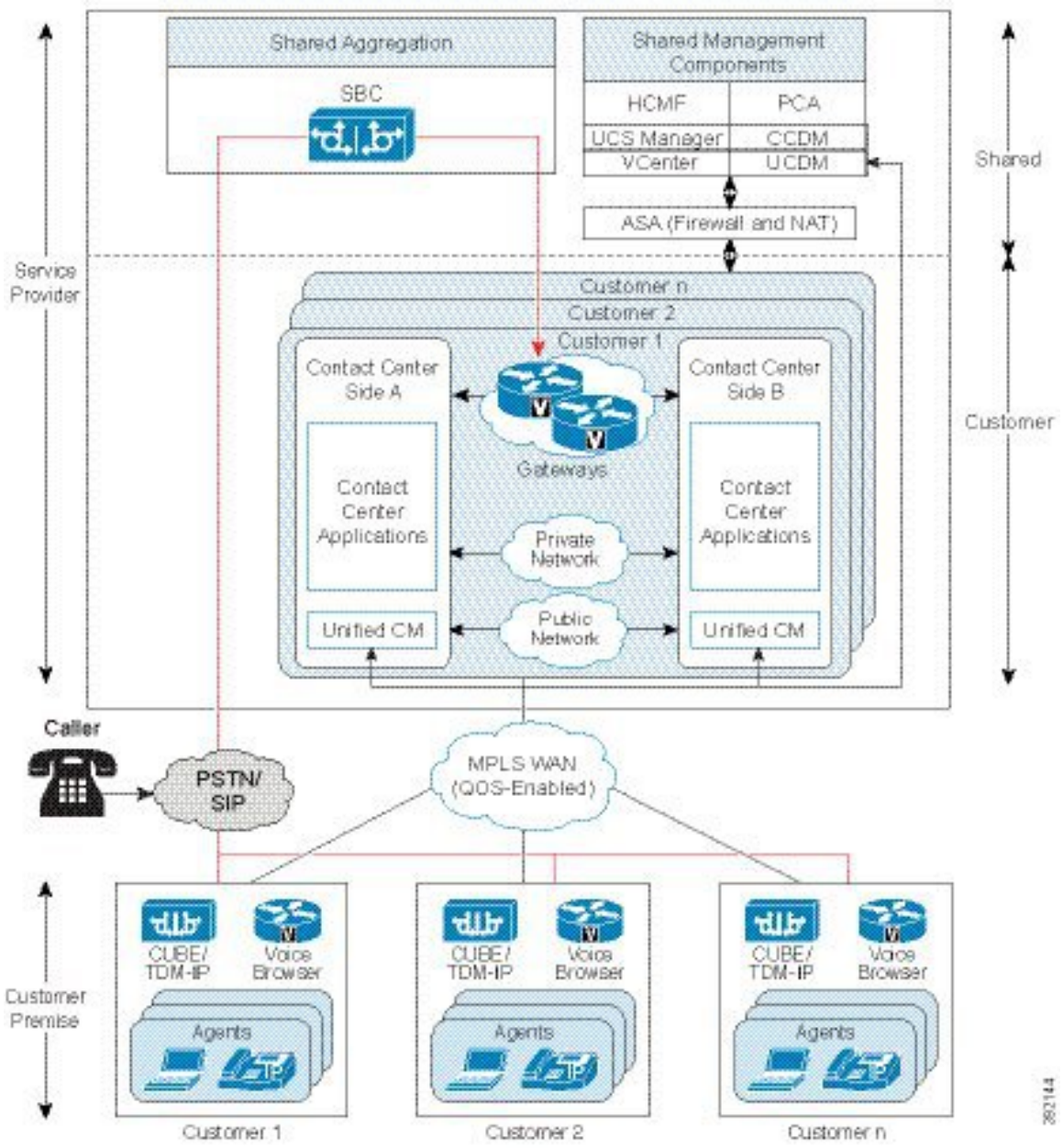
Note To take advantage of alternate routing on signaling failures, use the survivability service on all gateways pointing to Unified CVP. Always use this service, unless you have a specific implementation that prevents using it.

Router requery is not supported when using SIP REFER with Unified CVP Comprehensive Call Flow when the survivability service is handling the REFER message from Unified CVP. Other call flows can support router requery with REFER when Cisco IOS handles the REFER without the survivability service or if Unified CM handles the REFER. For third-party SIP trunks, the support of router requery with REFER depends on their implementation and support for SIP REFER.

Local Trunk Design Considerations

This figure shows the options for handling a local trunk at the customer premise: CUBE or a TDM gateway.

Figure 17: CUBE or TDM Gateway at the Customer Premise



CUBE at the Customer Premise

Consider the following if you use a CUBE at the customer premise:

- The CUBE gateway and the Cisco VXML gateway reside at the customer premise and calls are queued at the customer premise.
- You can colocate the CUBE and VXML gateway on the same ISR. You can also locate them on different ISRs where the number of VRU ports to agents ratio is small.

- CUBE Integrated Services Router (ISR) provides the security, routing, and Digital Signal Processors (DSPs) for transcoders.
- Use Redundant CUBE and Cisco VXML ISRs for failover and redundancy.
- Size the WAN bandwidth appropriately for calls from SBC to CUBE at the customer premise.
- CUBE supports flow-through mode. Flow-around mode is not supported.

TDM Gateway at the Customer Premise

You can route PSTN calls using local gateway trunks if you prefer to keep your E1/T1 PSTN.

Consider the following if you use the TDM gateway at the customer premise:

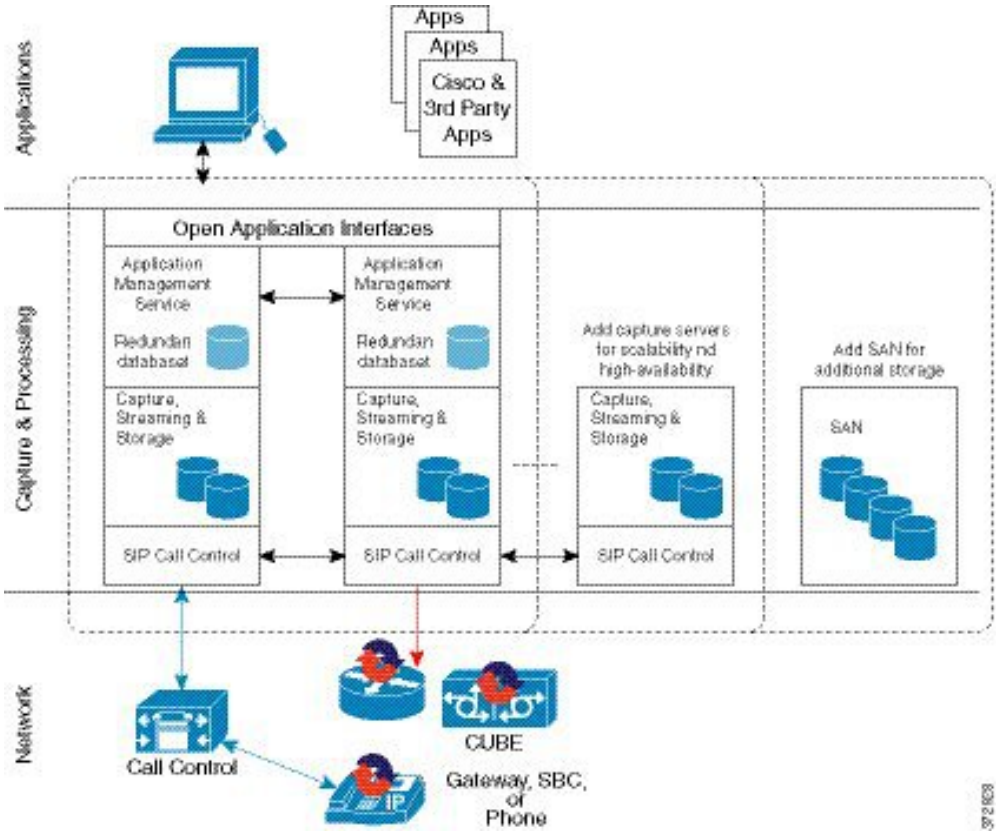
- Both the Cisco TDM Gateway and the Cisco VXML gateway reside at the customer premise.
- PSTN delivery is at the local customer premise.
- The media stays local at the customer premise for the local PSTN breakout. The VRU call leg is deterministically routed to the local VXML gateway. It only uses the centralized resources in spill-over scenarios.
- When media is delivered to a different site, Unified CM location-based call admission control limits the number of calls over the WAN link.
- Calls local to a customer premise use the G.711 codec. Calls going over the WAN link can use the G.729 codec to optimize the WAN bandwidth.
- The ASR/TTS server for local breakout is at the customer premise and resides on a UCS or equivalent server.
- An incoming call for the contact center must originate from the TDM gateway to anchor the call to the survivability service. Manually configure the contact center dialed number to route the calls to Unified CM.

Note Manually modify the call routing from TDM gateway for the session target to route the call directly to Unified CVP.

Optional Cisco Components Design Considerations

MediaSense Design Considerations

Figure 18: Cisco MediaSense Topology



MediaSense Capabilities

Platform	Capabilities
Phone	For a list of supported phones, see the *Compatibility Matrix* for your solution.
Supported Model	2vCPU, 4vCPU, and 7vCPU profiles
Voice Codec	G.711 and G.729
Session	For session-related details, see *Virtualization for Cisco MediaSense* at http://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/virtualization-cisco-mediasense.html.
Media Forking	CUBE, Phone, and TDM
Network	Intercluster communication over WAN is not supported.

SocialMiner Design Considerations

You deploy SocialMiner as a large single server. SocialMiner does not support redundant topologies for high availability.

You can deploy the server inside or outside the corporate firewall in "Intranet" and "Internet" topologies:

- The Intranet topology provides the additional security of your network firewall to reduce the risk of an external party accessing the system. Use this topology when SocialMiner accesses internal sites, such as an internal forum site.

  However, this topology prevents partners without VPN access from using SocialMiner. If an external agency manages your public relations functions, this topology prevents easy access by the agency. You also cannot render SocialMiner OpenSocial Gadgets in public internet containers such as iGoogle.

  The Intranet topology complicates proxy configuration, but it simplifies directory integration.

- The Internet topology puts SocialMiner outside of your network firewall. It relies on the built-in security capabilities of the SocialMiner appliance.

  Whether this topology's security is acceptable or not depends on how you use the system and your corporate policies. For example, if SocialMiner only handles public postings in your solution, you do not risk disclosure of sensitive information if it is compromised.

  The Internet topology can complicate directory integration.

You can deploy SocialMiner so that some users access the server through a firewall or proxy. For the customer chat interface, you can deploy the SocialMiner server behind a proxy server or firewall. This reduces the risk of it being abused and limits access by those outside the firewall.

Task Routing Considerations

Task Routing

Task Routing describes the system's ability to route requests from different media channels to any agents in a contact center.

You can configure agents to handle a combination of voice calls, emails, chats, and so on. For example, you can configure an agent as a member of skill groups or precision queues in three different Media Routing Domains (MRD) if the agent handles voice, e-mail, and chat. You can design routing scripts to send requests to these agents based on business rules, regardless of the media. Agents signed into multiple MRDs may switch media on a task-by-task basis.

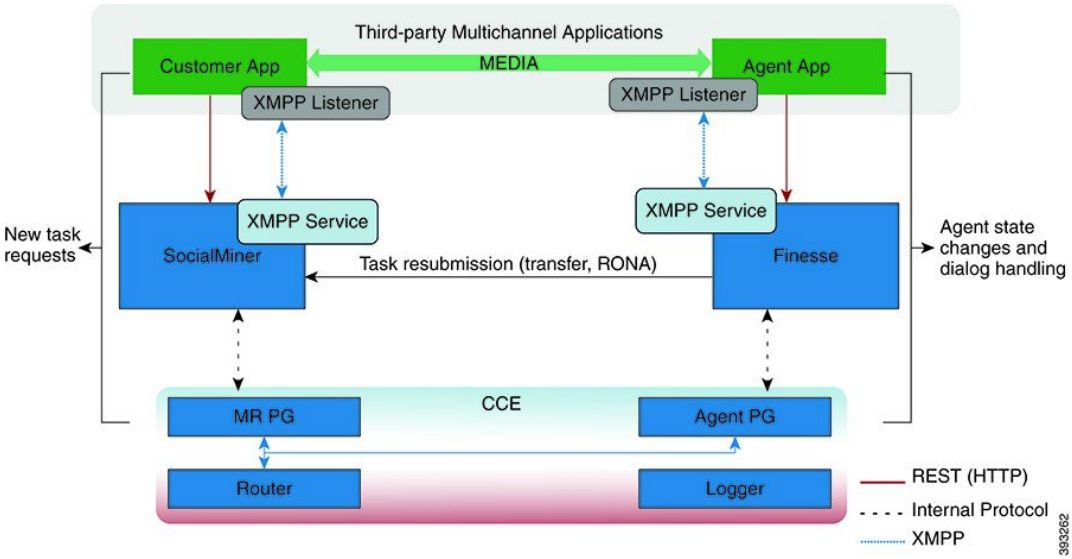
Enterprise Chat and Email provides universal queue out of the box. Third-party multichannel applications can use the universal queue by integrating with CCE through the Task Routing APIs.

Task Routing APIs provide a standard way to request, queue, route, and handle third-party multichannel tasks in CCE.

Contact Center customers or partners can develop applications using SocialMiner and Finesse APIs in order to use Task Routing. The SocialMiner Task API enables applications to submit nonvoice task requests to CCE. The Finesse APIs enable agents to sign into different types of media and handle the tasks. Agents sign into and manage their state in each media independently.

Cisco partners can use the sample code available on Cisco DevNet as a guide for building these applications (https://developer.cisco.com/site/task-routing/).

Figure 19: Task Routing for Third-party Multichannel Applications Solution Components



SocialMiner and Task Routing

Third-party multichannel applications use SocialMiner'sTask API to submit nonvoice tasks to CCE.

The API works in conjunction with SocialMiner task feeds, campaigns, and notifications to pass task requests to the contact center for routing.

The Task API supports the use of Call variables and ECC variables for task requests. Use these variables to send customer-specific information with the request, including attributes of the media such as the chat room URL or the email handle.

Note CCE solutions support only the Latin 1 character set for Expanded Call Context variables and Call variables when used with Finesse and SocialMiner. Arrays are not supported.

CCE and Task Routing

CCE provides the following functionality as part of Task Routing:

- Processes the task request.
- Provides estimated wait time for the task request.
- Notifies SocialMiner when an agent has been selected.
- Routes the task request to an agent, using either skill group or precision queue based routing.
- Reports on contact center activity across media.

Finesse and Task Routing

Finesse provides Task Routing functionality via the Media API and Dialog API.

With the Media API, agents using third-party multichannel applications can:

- Sign into different MRDs.
- Change state in different MRDs.

With the Dialog API, agents using third-party multichannel applications can handle tasks from different MRDs.

Task Routing Use Cases

Use the Task Routing APIs to request, queue, route, and handle third-party multichannel tasks in CCE.

We support these use cases:

- Agents handling multiple concurrent tasks in a single Media Routing Domain (MRD), and across MRDs.
- Interruptible MRDs. Agents handling tasks in those MRDs can be interrupted by tasks in other MRDs. For example, you can set an email MRD to be interruptible, meaning that an agent handling an email task can be interrupted by a task from another MRD, such as a voice call or a chat.
- Blind transfer to a specified script selector. Direct transfer is not supported.
- Agent sign out with assigned tasks. Those tasks are either closed or transferred, depending on the agent's dialogLogoutAction setting in the Finesse Media API.
- RONA. If an agent does not accept an offered task within the Start Timeout threshold for the MRD, Finesse resubmits the task for routing and makes that agent not routable.
- Context Service integration. You can register SocialMiner with Context Service in order to store data for Task Routing task contacts.

Task Routing Task Flow

This task flow describes a typical multichannel scenario, in which an agent is configured to handle email and chat tasks.

The email Media Routing Domain (MRD) is interruptible, and the agent is set to accept interrupts in that MRD. Because the email MRD is interruptible, an agent handling an email task can be interrupted with tasks from other MRDs. Because the agent is set to accept interrupts, the state of the agent, email task, and Finesse dialog in the email MRD change to INTERRUPTED if the agent is assigned a task in another MRD. The agent also cannot perform work on the email task while interrupted.

The chat MRD is non-interruptible.

The partner has developed email and chat applications that are integrated with the Finesse and SocialMiner APIs.

1. A customer sends an email to the company. The email application submits a new email task request to CCE. The task request includes a script selector and Call and ECC variables with customer-specific information, including the handle to the email.
2. CCE maps the script selector to a call type, which determines which routing script to run. The routing script queues the email task to the appropriate skill group or precision queue in the email MRD.

3. The email application polls for status and Estimated Wait Time (EWT).
4. An agent signs in to the email MRD and changes to Ready state.
5. CCE assigns the email to the agent. The Call and ECC variables used to create the task are included in the dialog's media properties. The application uses these variables to allow the agent to reply to the email. The agent starts work on the email dialog in Finesse.
6. Another customer sends a chat request to the company. The chat application submits a new chat task request to CCE. The Call and ECC variables include the chat room URL.
7. CCE maps the script selector submitted with the chat request to a call type, which determines which routing script to run. The routing script queues the chat task to the appropriate skill group or precision queue in the chat MRD.
8. The chat application polls for status and Estimated Wait Time (EWT).
9. While working on the email task, the same agent signs in to the chat MRD and changes to Ready state.
10. CCE assigns the chat task to the agent. The application uses the Call and ECC variables to add the agent to the chat room with the customer. The agent starts work on the chat dialog in Finesse.
11. The agent's state in the email MRD changes to Interrupted, and the email dialog state changes to Interrupted. The application disables actions for the email dialog.
12. The agent transfers the chat task to a different script selector. Finesse closes the chat dialog and resubmits the task to SocialMiner. The application closes the chat room.
13. The agent is not handling other non-interruptible dialogs, and the email dialog becomes active.
14. The agent continues working on the email dialog. The agent pauses the dialog to take a short break, and then resumes the dialog.
15. When the email reply is complete, the agent performs wrap up work for the dialog. The agent closes the dialog. Finesse sends a handle event to SocialMiner for the email task. The application sends the email reply to the customer.

Task Routing Design Impacts

Enterprise Reference Design Task Routing Support

All solutions support task routing, with the exception of the Small Contact Center solution for HCS for Contact Center.

Task Routing Deployment Requirements

Task Routing for third-party multichannel applications deployment requirements:

- Finesse and SocialMiner are required. Install and configure Finesse and SocialMiner before configuring the system for Task Routing.

  See the Finesse documentation at https://www.cisco.com/c/en/us/support/customer-collaboration/finesse/tsd-products-support-series-home.html.

  See the SocialMiner documentation at https://www.cisco.com/c/en/us/support/customer-collaboration/socialminer/tsd-products-support-series-home.html.

By default, access to the Social Miner administration user interface is restricted. Administrator can provide access by unblocking the IP addresses of the clients. For more details, see the *Control Social Miner Application Access* topic in the *Cisco Social Miner Installation and Upgrade Guide* guide at https://www.cisco.com/c/en/us/support/customer-collaboration/socialminer/products-installation-guides-list.html.

- You can install only one SocialMiner machine in the deployment.
- SocialMiner must be geographically colocated with one side of the Media Routing Peripheral Gateway (MR PG).
- Install SocialMiner in a location from which CCE, Finesse, and the third-party multichannel SocialMiner Task Routing application can access it over the network.

  If you install SocialMiner in the DMZ, open a port for CCE and Finesse to connect to it. The default port for CCE to connect to SocialMiner is port 38001. Finesse connects to SocialMiner over HTTPS, port 443.

  Install the third-party multichannel application locally with SocialMiner, or open a port on the SocialMiner server for the application to connect to it.

Task Routing Sizing and Capacity Limits

The sizing and capacity limits for Task Routing for third-party multichannel applications are the following:

- Maximum multimedia agents per system: 2,000 agents
- Maximum tasks per hour: 15,000 tasks
- Maximum concurrent tasks: 20,000 tasks
- Maximum tasks per agent across all an agent's MRDs: 5 tasks
- Maximum tasks per agent in a single MRD: 5 tasks
- Task submission rate: 5 tasks per second

  SocialMiner throttles the task submission rate to CCE to 5 tasks per second. SocialMiner holds a maximum of 10,000 tasks in the queue for submission. If the queue exceeds 10,000 tasks, then SocialMiner discards the additional tasks with the disposition code NOTIFICATION_RATE_LIMITED. Once the queue is ready again, additional tasks are added to the queue.

Task Routing Bandwidth, Latency, and QoS Considerations

For SocialMiner, the network must have sufficient bandwidth to reliably support HTTP. Task Routing Rest API requests carry only metadata; they do not carry media. If the customer application connects to SocialMiner via XMPP to receive task status notifications, the network must reliably support a persistent TCP connection with the SocialMiner XMPP server. Connecting toSocialMiner via XMPP does not significantly impact network bandwidth.

To calculate the required bandwidth for Task Routing tasks for the Finesse desktop, use the *Finesse Bandwidth Calculator*, available at: https://www.cisco.com/c/en/us/support/customer-collaboration/finesse/products-technical-reference-list.html.

For CCE, bandwidth, latency, and QoS considerations are the same for Task Routing tasks as they are for voice calls. A Task Routing task uses the same bandwidth as a voice call.

Related Topics

Bandwidth, Latency, and QoS for

Unified SIP Proxy Design Considerations

Consider these points when adding Cisco Unified SIP Proxy (CUSP) into your solution:

- CUSP is a VM that can reside on UCS B, C, and E series servers.
- A standard CUSP topology consists of 2 redundant, geographically separated gateways. The gateways have one proxy module each. They use SRV priority for redundancy of proxies. They do not use HSRP.
- CUSP can coreside with VXML or TDM gateways.
- Configure TDM gateways with SRV or with Dial Peer Preferences to use the primary and secondary CUSP proxies.
- Set up CUSP with Server Groups to find the primary and back up Unified CVP, Unified CM, and VXML gateways.
- Set up Unified CVP with a Server Group to use the primary and secondary CUSP proxies.
- Set up Unified CM with a Route Group with multiple SIP Trunks, to use the primary and secondary CUSP proxies

Performance Matrix for CUSP Deployment

CUSP baseline tests in isolation on the proxy give a maximum capacity of 450 TCP transactions per second. Use this as the highest benchmark and most stressed condition allowable. From the proxy server perspective, a CVP call entails an average of four separate SIP calls:

- Caller inbound leg
- VXML outbound leg
- Ringtone outbound leg
- Agent outbound leg

When a consult with CVP queuing occurs, the session incurs an extra four SIP transactions, effectively doubling the number of calls.

`Record Route` is turned off by default on CUSP.

Note Always turn the `Record Route` setting off on the proxy server. This avoids a single point of failure, allows fault tolerant routing, and increases the performance of the Proxy server. Using that setting on the proxy server doubles the impact to performance. It also breaks the high availability model. The proxy becomes a single point of failure for the call, if the proxy goes down.

Call Disposition with CUSP

The following sections discuss configuration of Cisco IOS Gateways using SIP. These examples highlight certain configuration concepts.

Cisco IOS Gateway Configuration

With Cisco IOS Gateways, dial peers are used to match phone numbers, and the destination can be a SIP Proxy Server, DNS SRV, or IP address. The following example shows a Cisco IOS Gateway configuration to send calls to a SIP Proxy Server using the SIP Proxy's IP address.

```
sip-ua
 sip-server ipv4:10.4.1.100:5060

dial-peer voice 1000 voip
 session target sip-server
 ...
```

The **sip-server** command on the dial peer tells the Cisco IOS Gateway to use the globally defined SIP Server that is configured under the **sip-ua** settings. In order to configure multiple SIP Proxies for redundancy, you can change the IP address to a DNS SRV record, as shown in the following example. The DNS SRV record allows a single DNS name to be mapped to multiple Reporting Servers.

```
sip-ua
 sip-server dns:cvp.cisco.com

dial-peer voice 1000 voip
 session target sip-server
 ...
```

Alternatively, you can configure multiple dial peers to point directly at multiple SIP Proxy Servers, as shown in the following example. This configuration allows you to specify IP addresses instead of relying on DNS.

```
dial-peer voice 1000 voip
 session target ipv4:10.4.1.100
 preference 1
 ...
dial-peer voice 1000 voip
 session target ipv4:10.4.1.101
 preference 1
 ...
```

In the preceding examples, the calls are sent to the SIP Proxy Server for dial plan resolution and call routing. If there are multiple Unified CVP Call Servers, the SIP Proxy Server would be configured with multiple routes for load balancing and redundancy. It is possible for Cisco IOS Gateways to provide load balancing and redundancy without a SIP Proxy Server. The following example shows a Cisco IOS Gateway configuration with multiple dial peers so that the calls are load balanced across three Unified CVP Call Servers.

```
dial-peer voice 1001 voip
 session target ipv4:10.4.33.131
 preference 1
 ...
dial-peer voice 1002 voip
 session target ipv4:10.4.33.132
 preference 1
 ...
dial-peer voice 1003 voip
 session target ipv4:10.4.33.133
 preference 1
 ...
```

DNS SRV records allow an administrator to configure redundancy and load balancing with finer granularity than with DNS round-robin redundancy and load balancing. A DNS SRV record allows you to define which hosts should be used for a particular service (the service in this case is SIP), and it allows you to define the load balancing characteristics among those hosts. In the following example, the redundancy provided by the three dial peers configured above is replaced with a single dial peer using a DNS SRV record. Note that a DNS server is required in order to do the DNS lookups.

```
ip name-server 10.4.33.200
 dial-peer voice 1000 voip
 session target dns:cvp.cisco.com
```

With Cisco IOS Gateways, it is possible to define DNS SRV records statically, similar to static host records. This capability allows you to simplify the dial peer configuration while also providing DNS SRV load balancing and redundancy. The disadvantage of this method is that if the SRV record needs to be changed, it must be changed on each gateway instead of on a centralized DNS Server. The following example shows the configuration of static SRV records for SIP services handled by cvp.cisco.com, and the SIP SRV records for cvp.cisco.com are configured to load balance across three servers:

```
ip host cvp4cc2.cisco.com 10.4.33.132
ip host cvp4cc3.cisco.com 10.4.33.133
ip host cvp4cc1.cisco.com 10.4.33.131
```

(SRV records for SIP/TCP)

```
ip host _sip._tcp.cvp.cisco.com srv 1 50 5060 cvp4cc3.cisco.com
ip host _sip._tcp.cvp.cisco.com srv 1 50 5060 cvp4cc2.cisco.com
ip host _sip._tcp.cvp.cisco.com srv 1 50 5060 cvp4cc1.cisco.com
```

(SRV records for SIP/UDP)

```
ip host _sip._udp.cvp.cisco.com srv 1 50 5060 cvp4cc3.cisco.com
ip host _sip._udp.cvp.cisco.com srv 1 50 5060 cvp4cc2.cisco.com
ip host _sip._udp.cvp.cisco.com srv 1 50 5060 cvp4cc1.cisco.com
```

SIP Proxy Dial-Plan Configuration

If you have a SIP Proxy, use different VRU labels for the Unified CM routing client and the CVP routing clients. The Unified CM routing client uses its VRU label to send a call to the CVP Call Server to hand off call control first. The CVP routing client uses its VRU label to send a call to the VXML Gateway for treatment. When a call comes to CVP, CVP transfers to the CVP routing client's VRU label. It then delivers the call to the VXML Gateway for queuing treatment.

Structure the dial plan in your SIP Proxy as follows:

```
[Unified CM routing client VRU label + correlation-id]: pointing to CVP servers
[CVP routing client VRU label + correlation-id]: pointing to VXML Gateways
```

Enterprise Chat and Email Design Considerations

Enterprise Chat and Email provides web and email interaction management through a common set of web servers and pages for agents and administrators. It integrates with the contact center enterprise solution to provide universal queuing of contacts to agents from different media channels.

For more architectural and design information, see the *Enterprise Chat and Email Design Guide* at http://www.cisco.com/c/en/us/support/customer-collaboration/cisco-enterprise-chat-email/products-implementation-design-guides-list.html.

Figure 20: Enterprise Chat and Email Design Considerations

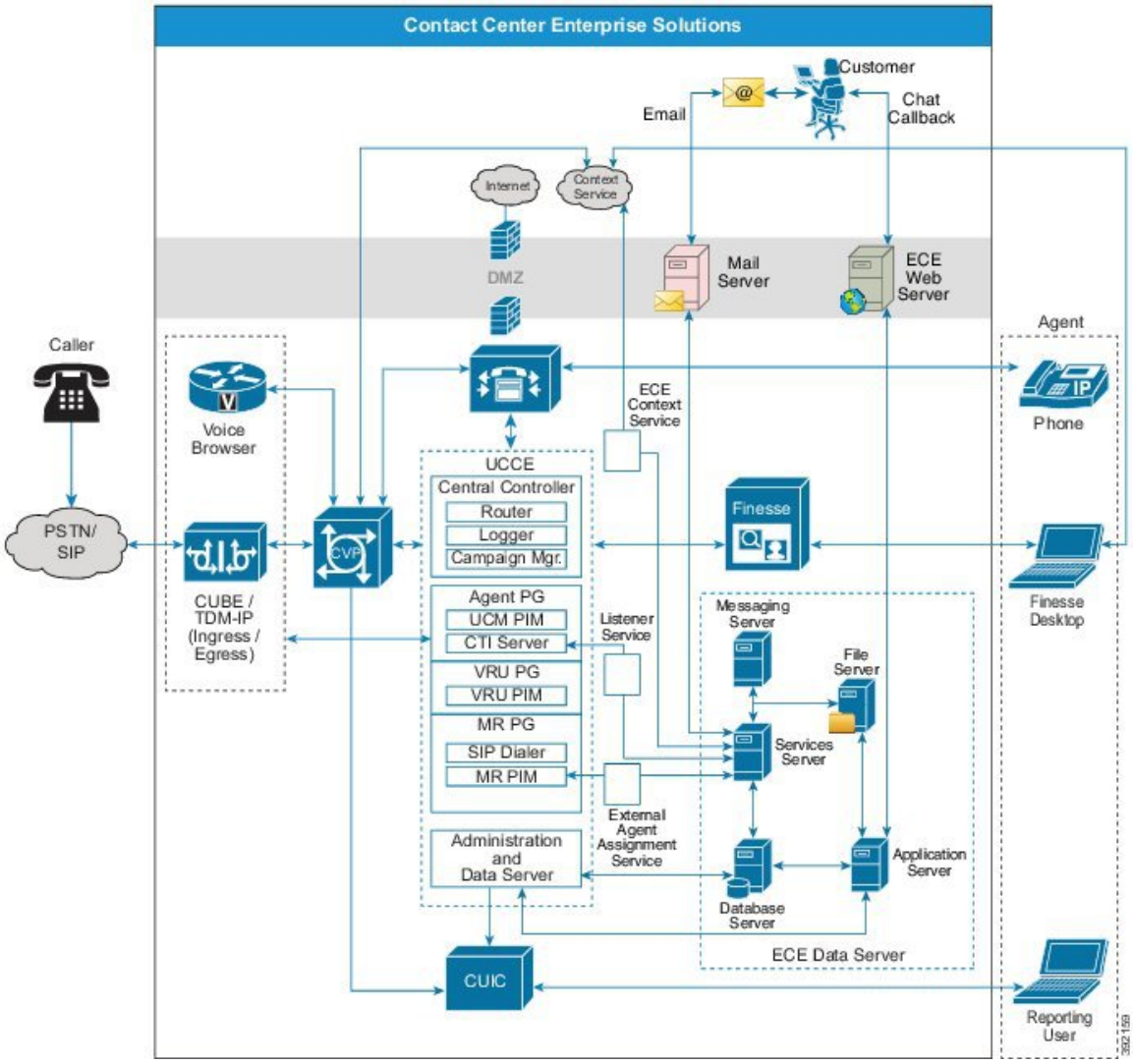


Figure 21: Enterprise Chat and Email Design Considerations

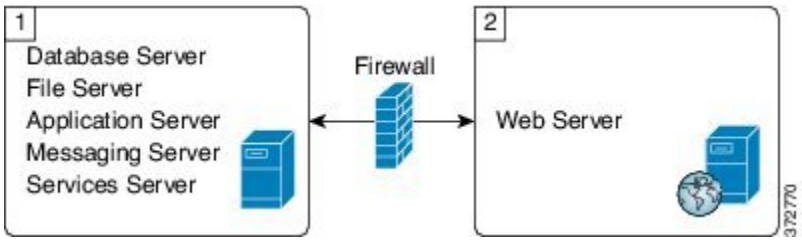
Enterprise Chat and Email Deployment Options

Due to the modular, component-based architecture, Enterprise Chat and Email (ECE) caters to the growing demands for concurrent user loads. To provide the flexibility to suit deployments of varied sizes, ECE supports various components that you can distribute across various servers in your solution.

Collocated Deployment

In a collocated deployment, you install the web server on one server and install all other components on a separate server. You can install the web server outside the firewall, if necessary.

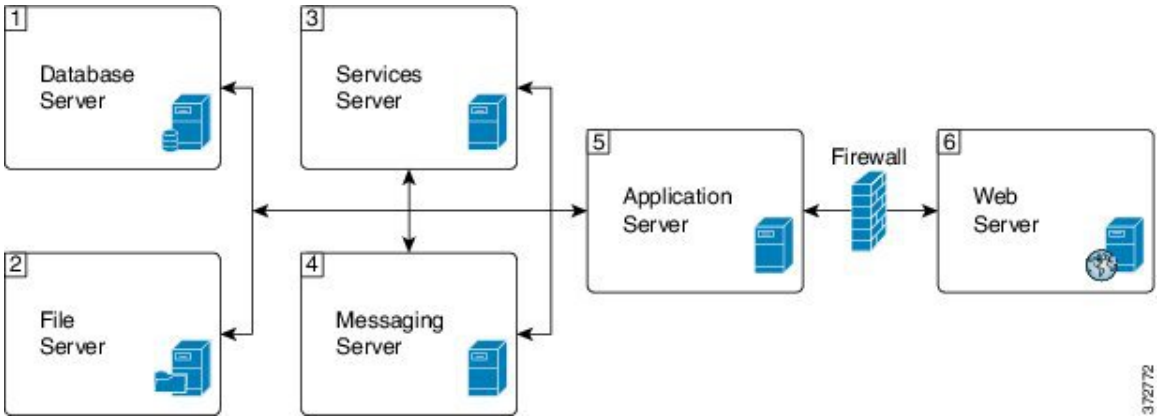
Figure 22: Collocated Deployment



Distributed-Server Deployment

In a distributed-server deployment, you install each component on a separate server, with the web server outside the firewall. You can restart the application, messaging, services, and web servers independently in this configuration without restarting any other servers.

Figure 23: Distributed-Server Deployment



Silent Monitoring Design Considerations

For supervisors managing teams, Unified CM-based Silent Monitoring provides a non-invasive mechanism to monitor agent voice calls.

Legacy span-based silent monitoring is supported only for Non-Reference Designs.

Unified CM-based Silent Monitoring Design Considerations

Unified CM-Based Silent Monitoring Call Flow

Cisco Finesse provides silent monitoring functionality through Unified CM Silent Monitoring. Cisco Finesse works with Unified CM Silent Monitoring as follows:

1. The supervisor application sends a REST request to the Cisco Finesse server to begin silent monitoring.
2. The Cisco Finesse server sends the AgentSuperviseCall() message to Unified CCE to start the silent monitoring session.
3. Unified CCE sends the CallStartMonitor() message to Unified CM.
4. Unified CM instructs the supervisor phone to call the Built-In Bridge (BIB) on the agent phone.

5. The supervisor phone calls the BIB on the agent phone.
6. The agent phone forwards a mix of the agent voice stream and customer voice stream.
7. Unified CM sends call events for the silently monitored call to Unified CCE.
8. Unified CCE sends update events to the Cisco Finesse server.
9. The Cisco Finesse server sends XMPP updates to the Cisco Finesse supervisor application.

Cisco Finesse does not support silent monitoring of mobile agents. Supervisors cannot silent monitor mobile agents and mobile supervisors cannot perform silent monitoring.

Supervisors cannot perform silent monitoring from a Cisco Finesse IP Phone. Supervisors can only perform silent monitoring from the Cisco Finesse desktop.

Unified CM-Based Silent Monitoring Impacts

Unified CM accomplishes silent monitoring with a call between the supervisor (monitoring) device and the agent (monitored) device. The agent phone mixes and sends the agent's conversation to the supervisor phone, where it plays to the supervisor.

Unified CCE supports the Silent Monitoring functionality available in Unified CM. Unified CM Silent Monitoring supports only one silent monitoring session and one recording session for the same agent phone.

Note Unified CM Silent Monitoring does not support mobile agents.

Unified CM Silent Monitoring can monitor any Unified CCE agent desktop, if the following conditions exist:

- The monitored agent uses a compatible Cisco Unified IP phone or Cisco IP Communicator. For more details, see the *Compatibility Matrix* for your solution.
- The contact center uses a compatible version of Cisco Unified CM. For more information, see the *Compatibility Matrix* for your solution.

Supervisors can use any Cisco IP Phone, including Cisco IP Communicator, to silently monitor agents.

Unified CM Silent Monitoring works the same as other call control functionality provided by Unified CM (such as conference and transfer). When the silent monitoring session begins, the desktop sends a message through Unified CCE, through Unified CM, and out to the phones where silent monitoring executes.

Messaging through Unified CCE and Unified CM impacts Unified CCE performance.

Third-Party Component Design Considerations

Use the following design considerations when any third-party component connects to Unified CCE through CTI Server.

All-Event Client Limits

The CTI server uses All-Event clients.

Maximum for Two vCPU Small Agent PG OVAs

On each Agent PG, you can have a maximum of seven All-Event clients on the CTI server. Cisco Finesse uses two of these clients.

Maximum for Four vCPU Large Agent PG OVAs

VMs built from the large OVA with four vCPU can support more All-Event Clients and monitor-mode connections. On each Agent PG, you can have a maximum of 20 All-Event clients on the CTI server. Cisco Finesse uses two of these clients.

All-Event Clients

The Cisco Finesse desktop solution uses two of the available All-Event clients. Some of the other possible consumers of the clients are:

- Outbound Dialer
- Real-Time Adherence (2)
- Some third-party recording vendors (2)
- Enterprise Chat and Email (2)
- B+S CRM Connectors

DNS Server Deployment Considerations

Consider the following when configuring a DNS servers for your solution:

- Configure the DNS servers for reverse lookup.
- Do not configure the DNS servers beyond a NAT network boundary.
- Deploy redundant DNS servers with low latency on the connection with the servers performing lookups.

Load Balancer Design Considerations

Load Balancers for Cisco Finesse Sign-In

After agents sign in to the Cisco Finesse desktop, the Cisco Finesse desktop client caches the IP address of both Cisco Finesse servers. If a Cisco Finesse server goes out of service, the Cisco Finesse client automatically redirects and signs the agent in to the other Cisco Finesse server. Given this client-side logic, the use of a load balancer for failover purposes is not supported.

However, the following are two scenarios in which you can use a load balancer with Cisco Finesse.

Note These scenarios only apply to the Cisco Finesse desktop. These scenarios do not apply to Cisco Finesse IP Phone Agents.

Navigation to the Cisco Finesse Sign-In Page

When an agent navigates to a Cisco Finesse server that is down or not reachable, the agent cannot access the sign-in page. The agent receives an error and must manually sign in to the other Cisco Finesse server. To avoid this manual step, you can use a load balancer with URL redirect mode to direct the agent to an active Cisco Finesse server. The Cisco Finesse SystemInfo REST API provides the status of the Cisco Finesse server. For details about this API, see the *Cisco Finesse Web Services Developer Guide*.

When you configure a load balancer to determine the status of the Cisco Finesse servers, use this call flow:

1. The agent points their browser to the load balancer.
2. The load balancer redirects the agent browser to an appropriate Cisco Finesse server.
3. The agent signs in to the Cisco Finesse server directly. At this stage, the load balancer is no longer part of the call flow.

Direct Use of the Cisco Finesse API

If you use the Cisco Finesse REST API directly, the call flow cannot use the Cisco Finesse client-side failover logic. You can opt to use a load balancer to manage high availability. The load balancer is part of a custom application which, like all custom applications, Cisco does not support. You or a Cisco partner provide the required support for the load balancer.

Remember that there are two connections between Cisco Finesse clients and the Cisco Finesse server:

- A REST channel for requests and responses
- An XMPP channel to send notifications from the server to the client

Both channels for a given client must connect to the same Cisco Finesse server. You cannot connect the load balancer to the REST connection for one Cisco Finesse server and to the XMPP channel connection for the other Cisco Finesse server. This setup provides unpredictable results and is not supported.

Load Balancers for Cisco Unified Intelligence Center (CUIC)

Unified Intelligence Center can be accessed using load balancers. The following conditions apply:

- Live Data reports cannot be accessed through the load balancer.
- Load balancer is not supported when CUIC nodes and browser clients are split across a WAN.

Load Balancers for CVP

You can use load balancers with the Unified CVP solution components to provide load distribution and high availability for HTTP, HTTPS, and MRCP traffic. Load balancers can spread the rendering of the VXML pages between VXML Gateways and VXML Server. Load balancers can also spread the fetching of the media files for VRU scripts execution from media servers.

Note If your solution is MRCPv2, use CUSP for load balancing.

CVP now supports any third-party load balancer that meets these requirements:

- Supports both SSL offloading and SSL pass-through

- Supports load balancer high availability
- Does not have mandatory session stickiness
- Uses cookie-insert for persistence
- Distribution algorithm is round-robin

Load Balancers for the Unified CCE Administration Tool

You can use a load balancer with the Unified CCE Administration tool in these scenarios.

Navigation to the Unified CCE Administration Sign-In Page

When administrators or supervisors navigate to the Unified CCE Administration tool on a server that is down or not reachable, they cannot access the sign-in page. They receive an error and must manually sign in to Unified CCE Administration on the other server. To avoid this manual step, you can use a load balancer with URL redirect mode to direct their sessions to an active server.

Usage scenario:

1. Users point their browsers to the load balancer.
2. The load balancer redirects the browser sessions to an appropriate Unified CCE Administration server.
3. The users sign in to the Unified CCE Administration server directly.

Direct Use of the Unified CCE Administration API

If you use the Unified CCE Administration REST API directly, you can opt to use a load balancer to manage high availability. The load balancer is part of a custom application which, like all custom applications, Cisco does not support. You or a Cisco partner provide the required support for the load balancer.

Load Balancers with Enterprise Chat and Email

Do not use the load balancer's redirect mode with Enterprise Chat and Email

Recording Design Considerations

Network-Based Recording Design Considerations

The network-based recording (NBR) feature supports software-based forking for Real-time Transport Protocol (RTP) streams. With media forking, you can create midcall multiple streams (or branches) of audio and video for a single call. You can then send the streams of data to different destinations. To enable network-based recording using CUBE, refer to its configuration guide. You can configure specific commands or use a call agent. CUBE acts as a recording client and MediaSense recorder acts a recording server.

Note Network-based recording works with the call survivability feature.

This figure shows the call flow for network-based recording.

Figure 24: Network-Based Recording Call Flow

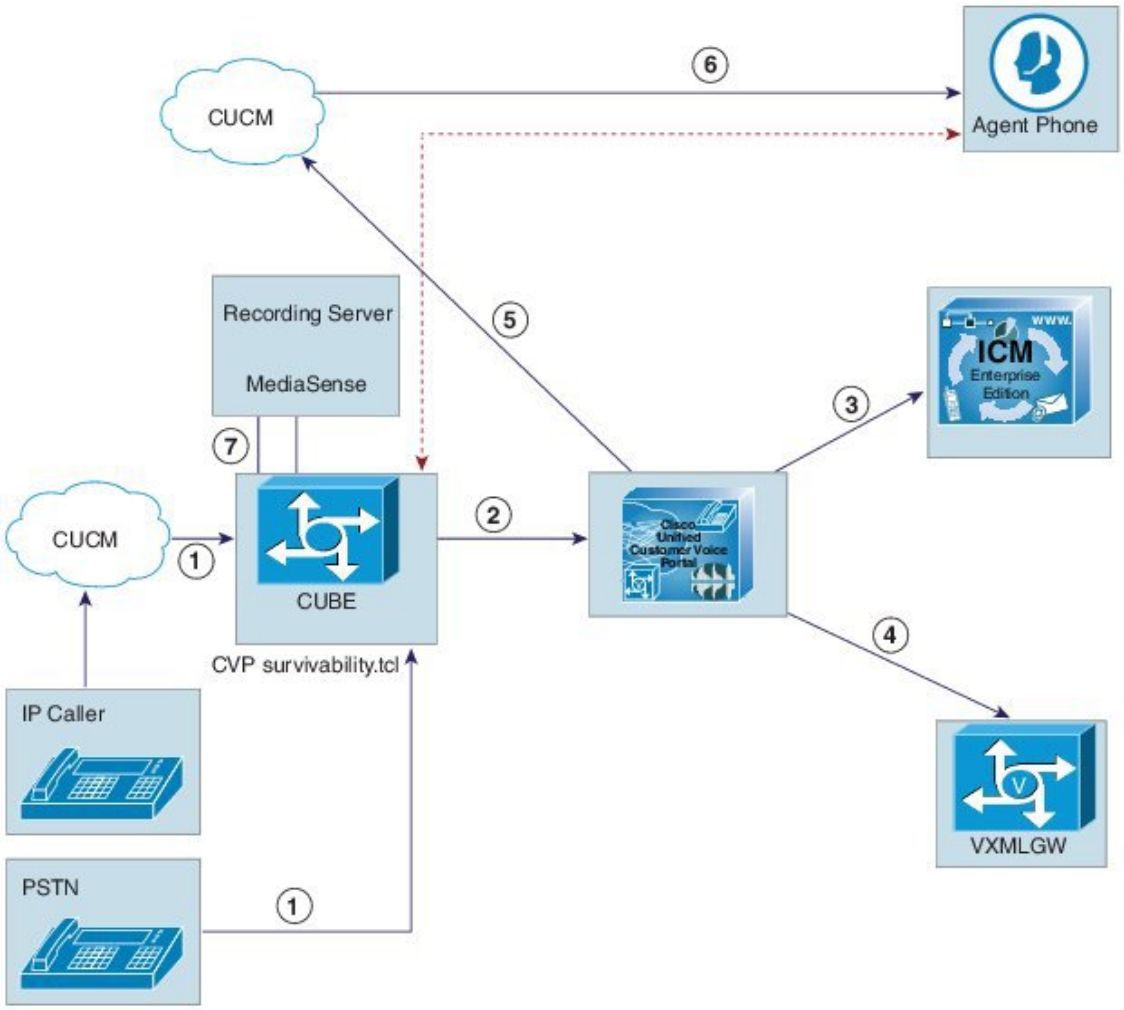


Figure 25: Network-Based Recording Call Flow

A typical call flow for network-based recording is as follows:

1. The incoming call arrives at CUBE.
2. The Ingress Gateway sends the call to Unified CVP.
3. Unified CVP sends the incoming call request to Unified CCE and gets a VRU label.
4. Unified CVP sends the call to the VXML Gateway. The caller hears the VRU. However, the call is not recorded.
5. After the agent is available, Unified CVP connects the caller to the agent.
6. Network-based recording starts for this conversation.

Network-Based Recording Limitations

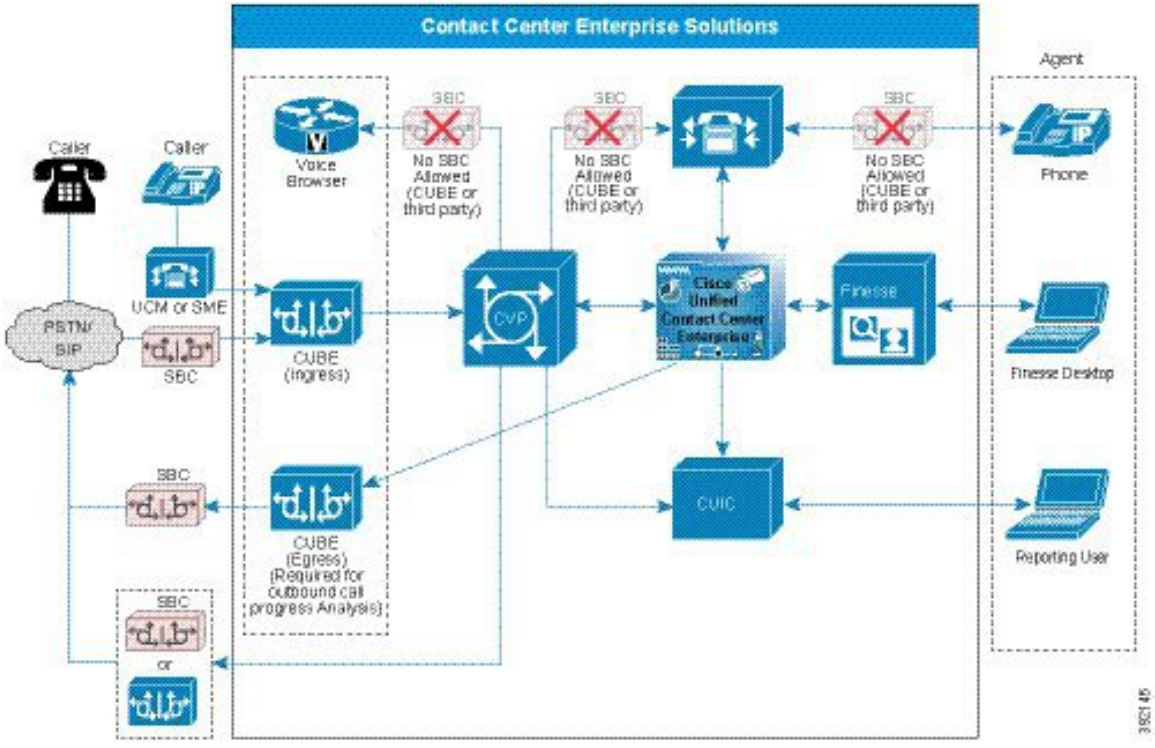
- For agent to agent call transfer, network-based recording does not work but phone-based recording does. If you want to use network-based recording, you can use an ISR gateway between Unified CVP and Unified CM.
- The NBR feature is supported only on selected IOS image trains. For more information about the supported IOS image trains, see the *Compatibility Matrix* for your solution.

Session Border Controllers

You can use a third-party Session Border Controller (SBC) as an ingress or egress border element between the PSTN and your contact center. Most solutions require CUBE between a third-party SBC and Unified CVP to ensure support for all contact center enterprise features. You cannot use an SBC in most other communication legs in your deployment.

The following diagram shows where you can and cannot use an SBC:

Figure 26: SBC Implementation



Note CUBE includes both SBC functionality and features for contact center enterprise solutions. You can replace CUBE with a third-party SBC where it supplies only SBC functions. Your solution still requires CUBE for the contact center enterprise features.

We do not actively test third-party SBCs.

Consider the following general design guidelines and limitations when you consider adding an SBC to your deployment.

Support Policy

Cisco may ask you to remove an SBC from the call flow temporarily for any interoperability issues. If the issue is not found without the SBC, then Cisco TAC hand over the case to the manufacturer of that SBC.

Third-Party SBC Use Without CUBE

When you connect to a third-party SIP device, including a SIP PSTN service provider, use a CUBE. If you do not place a CUBE between Unified CVP and the SIP device, ensure that both sides are compatible with thorough integration testing.

When connecting to a PSTN SIP Trunking service without a CUBE, carefully consider how to secure the connection between the contact center and the service provider. Also consider how to accomplish NAT and address hiding. Otherwise, the service-provider network can have full access to the contact center network. As the service-provider interconnect interface provided by Cisco, CUBE addresses both of these concerns.

For solutions that do not use certain CVP features, you might connect a third-party SBC directly to CVP. Such solutions require careful testing to ensure interoperability.

Important Designs that directly connect a third-party SBC to Unified CVP require a special exception from Cisco to deploy.

Supported Features

Without specific testing, support for any feature is not guaranteed. Past testing generally showed support for these features:

- G.711ulaw, G711alaw, and G.729 (no Annex B) codecs
- DNIS and ANI presentation
- SIP/TCP on the SBC's internal interface and SIP/UDP on the external interface
- CVP-based Queuing
- CVP applications with DTMF
- CVP-based intrasite transfers using re-INVITE
- Unified CM-based intrasite transfers and conferences
- SBC-based midcall codec negotiation. Basic call flows generally work, but complicated call flows are less likely to work.
- Cisco Unified Communications Manager (Unified CM) midcall codec negotiation (with transcoder insertion where needed). Basic call flows generally work, but complicated call flows are less likely to work.
- SBC converting SIP INFO messages from CVP to RFC2833 tones (for DTMF-based transfers)

Note Timing issues between CVP and the SBC can result in CVP disconnecting the call before the SBC completes the transfer.

- REFER transfers with SBC in REFER

Note CVP expects a BYE or a final NOTIFY that terminates the subscription from the SBC when the transfer is complete. If the SBC does not send either of these, the call hangs for the duration of the subscription and the CVP license is not released.

- SIP 302 Redirect responses with SBC in consume mode
- CVP-based Redirect on No Answer
- Call hold

Unsupported Features

Without specific testing, support for any feature is not guaranteed. Past testing generally showed that these features are unsupported:

- Outbound Option with SIP Call Progress Analysis
- Courtesy Callback
- Call survivability and associated features (survivability.tcl script, local branch SRST and TOD routing, Hookflash, TBCT)
- Handling of VRU PG failure and any downstream failure handling through survivability
- Queue at the edge (using CVP `SendToOriginator` feature)
- Video call flows
- SIP over TLS and SRTP
- Locations-based CAC
- REFER with Replaces
- SBC configured as a SIP proxy (instead of CUSP) for messages between Cisco components
- Network Trunk Group Utilization and Reporting
- Trunk group utilization
- SIP Resource Availability Indicator (RAI) dynamic call routing
- SIP dial-peer based recording

Generally, features are unsupported because various Unified CCE and CVP features rely on specific CUBE functions that are not present in third-party SBCs.

Other Caveats

The following are other caveats to consider if you add a third-party SBC to your deployment:

- Most third-party SBCs do not generate a Cisco-Guid header which Unified CCE uses for end-to-end call tracking.
- If you use SIP over TCP between the SBC and CVP, some calls can drop if an SBC switches over through its high-availability feature. For each CVP server used, exactly one call can drop after the switch over occurs. All other calls in progress at the given CVP server stay active with stateful signaling and media. The dropped call is the first that sends a SIP message to the SBC after the switch over. SIP over UDP does not display this behavior.
- In solutions that use third-party SBCs, the network might have a firewall between the SBC and the subnet with the contact center solution components. The firewall configuration allows any SIP-related communication between the SBC and CVP.

  For the SIP traffic over TCP, CVP creates an outgoing connection with the SIP port for the SBC. On CVP, an idle TCP connection remains in the ESTABLISHED state for 4 hours, even if there are no calls between the SBC and CVP.

  The firewall configuration might free such idle connections without CVP detecting it. When CVP next receives incoming calls, a few calls fail because CVP cannot send the SIP requests on the outgoing TCP connection to the SBC. Calls can fail until CVP establishes a new connection to the SBC.
- In some scenarios, CVP should send busy and ring-no-answer notifications to the SBC. In such cases, use the Remote-Party-ID header and manipulate the header to include "--CVP" at the end of the display name.
- Not all SIP service providers support advanced features such as REFER, 302 Redirect Messages, DTMF-based take-back-and-transfer, or data transport (UUI, GTD, NSS, and so on).
- Unified CM can use an UPDATE method for session refresh on its signaling path to CVP. This use case is untested, and therefore unsupported, when CVP connects directly to a third-party SBC.
- CVP supports both IPv6 SDP and IPv4 SDP. Use Alternate Network Address Format (ANAT) when the SBC uses both IPv4 and IPv6 in the session description.

Cisco Virtualized Voice Browser

We have not tested Cisco VVB with any third-party SBC.

Speech Recognition and Text to Speech

Automatic Speech Recognition (ASR) or Text-to-Speech (TTS) Server cannot use silence suppression and must use the G.711 codec.