Active Recording Ecosystem Solution Guide

Genesys Voice Platform 8.1.0

12/30/2021

Table of Contents

Active Recording Ecosystem Solution Guide

The Active Recording Ecosystem 8.1 Solution Guide provides an overview of the Genesys Active Recording Ecosystem. These pages are valid for all 8.x releases of Media Server and SIP Server. See the summary of the highlighted topics below:

About the Active Recording Ecosystem Solution

Find out about the Active Recording Ecosystem:

- Overview
- New in This Release
- Call Recording

Feature Configuration

Find out about the supported features that include these topics:

- Call Flows
- Recording

all topics>>

Installation and Configuration

Find out about how to install the components:

- CIM
- SIP Server
- Media Server for MSR
- Media Server for MSML
- Genesys Media Server

Troubleshooting

Find out about troubleshooting:

- Troubleshooting

Overview

Active Recording Ecosystem Overview

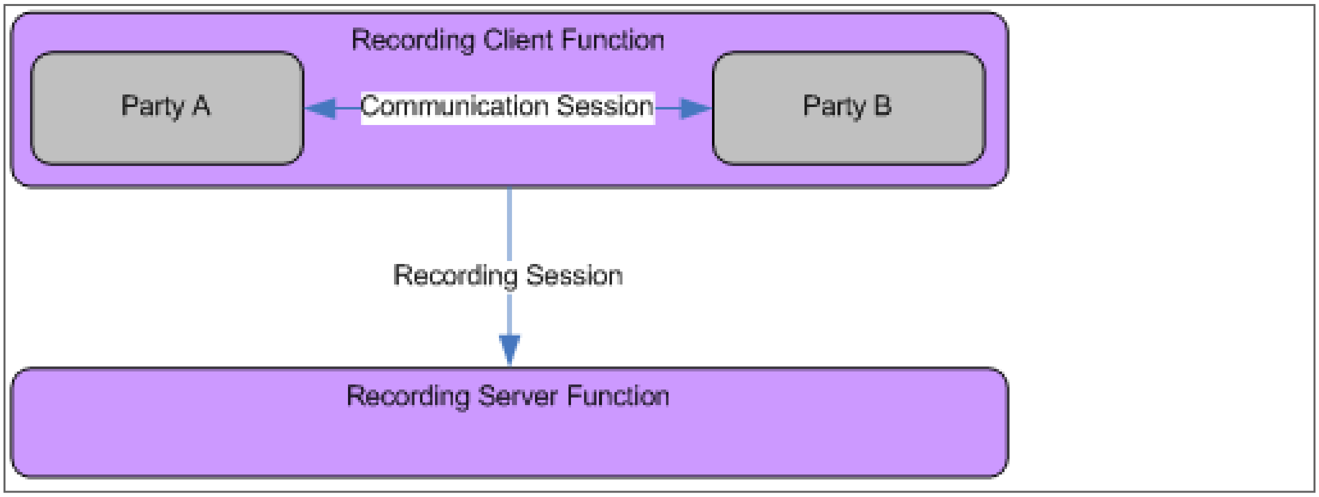
The Active Recording Ecosystem uses Media Stream Replication (MSR) for a fully Active recording solution with Dual Channel Recording. SIP sessions to the recorder provide basic call information and voice (Real-time Transport Protocol (RTP)) data. MSR is where Media Server replicates the RTPs and makes them available to the recording server. Additional events and information are provided by the T-Server part of SIP Server.

Recording Server Function—The recording server function is a logical function that acts as the collector of the recorded media. The recording server is also a SIP User Agent that negotiates the parameters for the recorded media, and typically uses CTI to receive call metadata as well as run-time controls. The recording server function also provides the capability to search and retrieve recorded media, and may provide real-time analysis of the media. The term "server" does not necessarily mean that the recording server only functions as a server; the recording server is also a T-lib client.

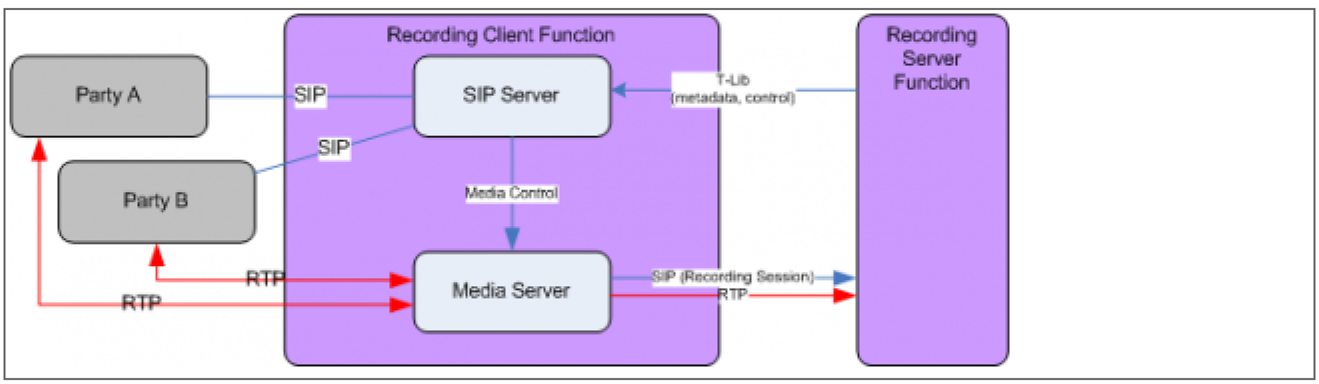
Recording Client Function—A SIP User Agent that acts as the source of the recorded media. This is a logical function that may span across multiple components, specifically: the combination of SIP Server and Media Server is the recording client function in Genesys environment. In some environments, the recording client function may be the combination of the switch and the handset.

Communication Session—The session created between the endpoints, such as a conversation between the customer and the agent. The Communication Session, sometimes also called the Recorded Session, may be on any transport, but for Genesys, SIP Server is managing the Communication Session, so the transport is SIP.

Recording Session—A session created for the purpose of recording a Communication Session. The Recording Session is SIP. Multiple Recording Sessions can be applied to a single Communication Session.



This diagram illustrates the relationship between a Communication (or Recorded) Session and a Recording Session.



This diagram illustrates the relationship between the Recording Client and Server functions, from the point of view of SIP Server and Media Server.

Core Capabilities

The following is a list of some of the core capabilities of the Active Recording Ecosystem.

- Full-time recording—Record every call for a specific DN through configuration.
- Selective recording—A recording decision is made to record a call when the Communication Session is first established. T-lib recording functions are provided to allow third parties such as a routing strategy or a business rule to make the decision to record the call.

- Dynamic recording—Recording Sessions are established on an as-needed basis after the Communication Session is established. T-lib recording functions are provided to allow third parties, such as Agent Desktop, to record on demand.
- Pre-recording—Pre-recording is an extension to selective recording that starts the recording when the Communication Session is being established. The recording server function provides the ability for third parties, such as Agent Desktop, to request call recording at any time during the call (and even shortly after the end of the call) for the call to be recorded in its entirety.
- A call recording can be started while supervisor monitoring is enabled.
- Real-time control of the call recording—The recording can be paused and resumed on demand by the agent, or by the workflow when the customer is providing sensitive data such as a PIN.
- The ability to integrate with a third party recording server to receive the call recording for real-time monitoring and quality analysis:
    - Multiple simultaneous Recording Sessions can be active for the same Communication Session for different purposes.
    - The call can also be recorded locally while streaming to third party recorders.
- Support of secured communications such as SIPS and SRTP.

Deployment Model

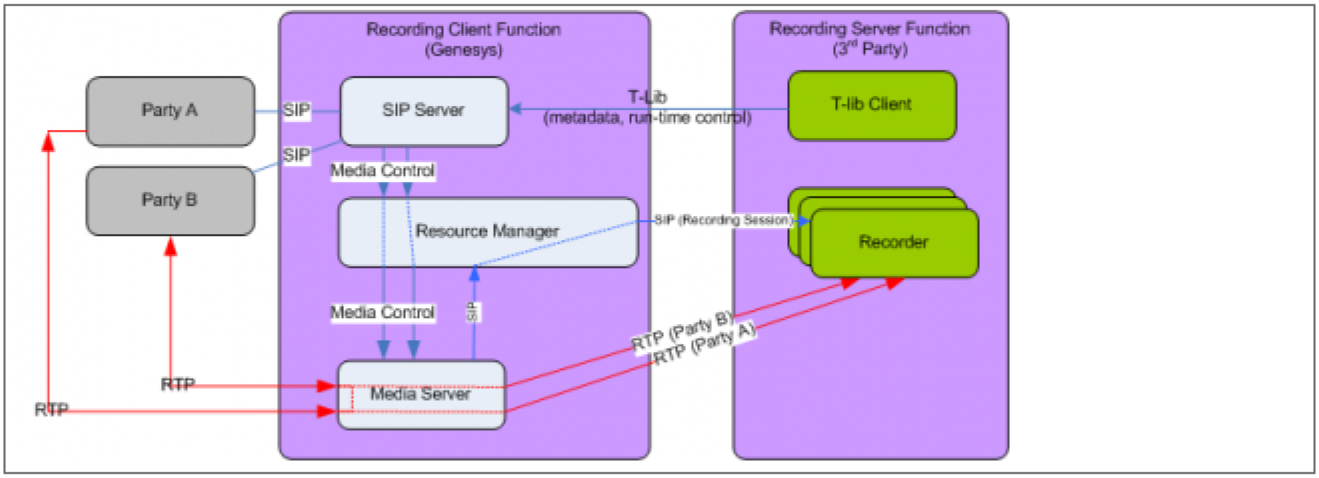
Genesys provides the Recording Client Function and is composed of the following core components:

- SIP Server—Handles communication sessions.
- Genesys Voice Platform (GVP) Media Server—Provides media stream replication for the third party recorder.
- GVP Resource Manager—Locates the addresses of a cluster of third party recorders.

A third party recorder provides the Recording Server Function, and is composed of two core components:

- T-lib client—Initiates recording requests to specific calls for dynamic recording and run-time controls for pause and resume functions. The client also receives call metadata from T-lib events.
- Recorder—A SIP User Agent that handles recording sessions and persists the replicated media streams.

When a recording request is initiated, SIP Server directs the media of Party A and B towards Media Server. SIP Server uses media control to instruct Media Server to replicate media stream towards the media recorder. When Media Server receives the media streams, the media stream establishes a recording session (a SIP dialog) towards the third party call recorder. For more information about recording sessions, see Media Server Recorder.



This diagram illustrates the Call Recording Solution.

New in this Release

New Features

- Support of Geo-Location.
- Support of Alpha-numeric DNs.
- Audio tone for compliance recording.

Recording

This section describes the types of recording supported:

- Video Recording
- Genesys Voice Platform

How Call Recording Works

Depending on how the feature is configured, the basic call flow for call recording is as follows:

1. Call recording is initiated in one of the following ways:
   - Static configuration—Recording is enabled through static DN-level configuration on either the customer side (`Trunk` DN) or on the agent side (`Extension` DN or Agent Login).
   - Routing strategy—The routing strategy initiates recording through the `TRouteCall` request that it sends to SIP Server.
   - T-Library client or third party recorder—A T-Library client initiates recording through a `TPrivateService` request that it sends to SIP Server.
2. Based on this trigger, SIP Server builds a request URI that includes key recording-related parameters. It then sends this request URI in an `INVITE` to Resource Manager.
3. Resource Manager determines the right Media Control Platform (MCP) to provide the service, and then forwards the `INVITE` to the selected MCP, in order to set up the service.
4. SIP Server sends additional Media Server Markup Language (MSML) instructions in `SIP INFO` messages, telling Media Server to start the recording.

For additional control over the established recording session, the T-Library `TPrivateService` request can be used to initiate new actions—for example, to pause or resume recording. SIP Server forwards the resulting MSML instructions in new `INFO` messages.

Media Format

Supported Media File Format

MSML-based call recording supports the .wav and G.711 stereo file formats.

Building the Request URI

Building the Request URI for the Recording

SIP Server builds the Request URI for the call recording in a number of ways, depending on configuration and type of call recording:
`sip:msml=<conf-id>@<resource-managaer>;<dn>=<DN>;record`

where,

- `msml`—is the fixed part of the URI. It identifies the protocol as MSML.
- `conf-id`—is the unique identifier for the MSML/recording session. It ensures that all users are connected to the correct Media Server.
- `DN`—is the DN of the endpoint that SIP Server will record.
- `Record`—Identifies "record" as the type of MSML session. Genesys Media Server can then properly handle recording separately from other MSML services.

Dynamic Call Recording

Call recording can be started on an as-needed or "emergency" basis during an ongoing call. To initiate dynamic recording, recording-related parameters are included in the `Extensions` attribute in either of the following T-Library messages:

- TRouteCall
- TPrivateService

TRouteCall

The URS routing strategy must be configured to include recording-related parameters in the `TRouteCall` request that it sends to SIP Server. The `Extensions` attribute must include the key record, with one of the following values:

- `source`—The recording will be initiated on the routing destination DN (agent) and will continue as long as the agent stays in the call.
- `destination`—The recording will be initiated on the DN that sent the call to the routing point (customer) and will continue as long as the customer stays in the call.

TPrivateService

The T-library client or third party recorder must include recording-related parameters in the `PrivateService` request that it sends to SIP Server. To initiate dynamic recording with `TPrivateService`, the request uses the following parameters:

Attribute	Value
PrivateMsgID	Specifies the type of recording operation to be performed: • `GSIP_RECORD_START (3013)`—Starts the recording.
ThisDN	Specifies the DN on behalf of which the recording operation is requested. This DN must be registered

Attribute	Value
	by the T-Library client
ConnectionID	References the ID for the call that is currently being recorded.

Extensions	Specifies key-value pairs used to control the recording session:
	• `record`—Set to source or destination.
	• `id`—Adds a recording identifier to the recording session. This identifier must be globally unique; it is passed back in the recording session. If this parameter is not included in the request, SIP Server will construct a unique identifier based on the `recording-filename` option.
	• `dest`—Overrides the default location of the 3rd party recording server.
	• `params`—Adds additional parameters that are passed as general key-value pairs in the request.
	There parameters will appear in the recording session. For example,
	`AttributeExtensions...`
	`'record' 'source'`
	`'id' '32980asdf320990ad'`
	`'dest' 'sip:172.24.129.75:5070'`
	`'name1' 'value1'`
	`'name2' 'value2'`

Reasons	Specifies any reasons. Processed the same as for all other T-Library requests.

Mid-Call Control

Mid-Call Control of the Recording Session

Using `TPrivateService` requests, T-Library clients can control, in real-time, an ongoing recording session. The client can pause, resume, or stop the recording. SIP Server translates recording-related parameters from the request to `INFO` messages that it sends to Genesys Media Server.

Supported mid-call actions are as follows:

- Stop the recording.
- Pause the recording.
- Resume a paused recording.

To control mid-call recording, the `TPrivateService` request uses the following parameters:

Attribute	Value
PrivateMsgID	Specifies the type of recording operation to be performed: • `GSIP_RECORD_STOP (3014)`—Stops the recording. • `GSIP_RECORD_PAUSE (3015)`—Pauses the recording. • `GSIP_RECORD_RESUME (3016)`—Resumes the recording.
ThisDN	Specifies the DN on behalf of which the recording operation is requested. This DN must be registered by the T-Library client.
ConnectionID	References the ID for the call that is currently being recorded.
Reasons	Specifies any reasons. Processed the same as for all other T-Library requests.

Transfers

Recording During Transfers and Conferences

SIP Server supports continuous recording for calls that are transferred or added to a conference. Once recording is initiated (through DN configuration, routing strategy, or by T-library client), recording will continue for as long as one party that is set for recording remains in the call. Recording ends when no more recording-enabled parties are left.

Architecture

The overall recording architecture is designed to adapt to different target solutions, while the core of the basic call flow remains consistent throughout all target solutions. This section describes the architecture and contains the following sections:

- Core Components
- Call Recording
- SIP Server Call Recording Model

Components

There are four main components involved in the architecture, and their roles are described in each of the solutions that follow.

- **SIP Server**—The core competency of SIP Server is routing and call control, and SIP Server is responsible to initiate call recording by using media control to direct media towards Media Server.
- **Resource Manager**—A SIP Proxy that manages a pool of Genesys Media Servers and applies runtime policies such as ensuring call legs to the same conference are pinned to the same Media Server. For Resource Manager, it is also the SIP proxy for the Recording Session so that Resource Manager hides the high availability details of each third party vendor from other Genesys components such as Media Server. Resource Manager also generates the call detail record (CDR) that allows correlation with individual call recordings.
- **Media Server**—Performs the actual file-based recording or replicates the media stream to a third party recorder by establishing the Recording Session to the recording server function. Media Server is also responsible for negotiating the media between the endpoints, to minimize the need for transcoding, or to preserve security of the audio stream.
- **Reporting Server**—An optional component in this solution; this provides storage of CDR and call events for Resource Manager and Media Server, and provides a web service to provide the user (through Genesys Administrator) the ability to query CDR and other call event information.

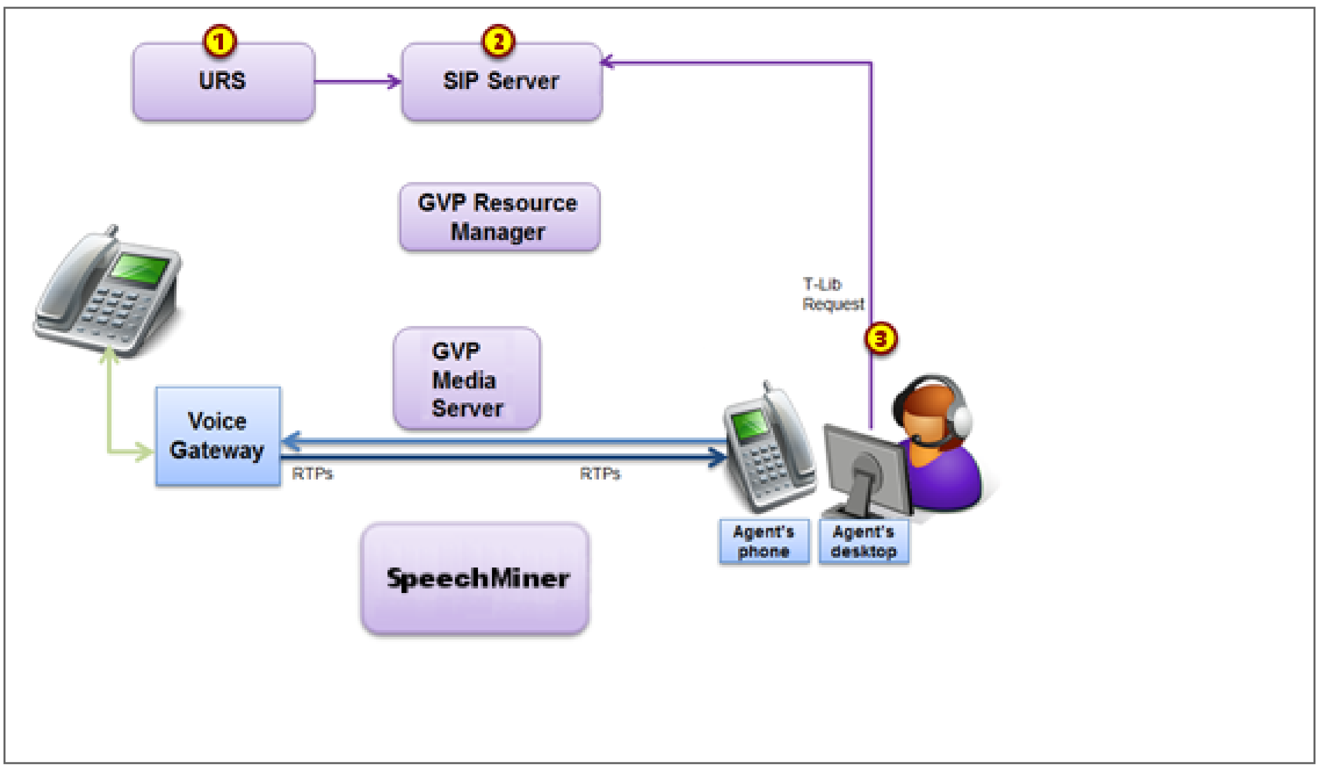
Call Recording

This section describes how call recording works.

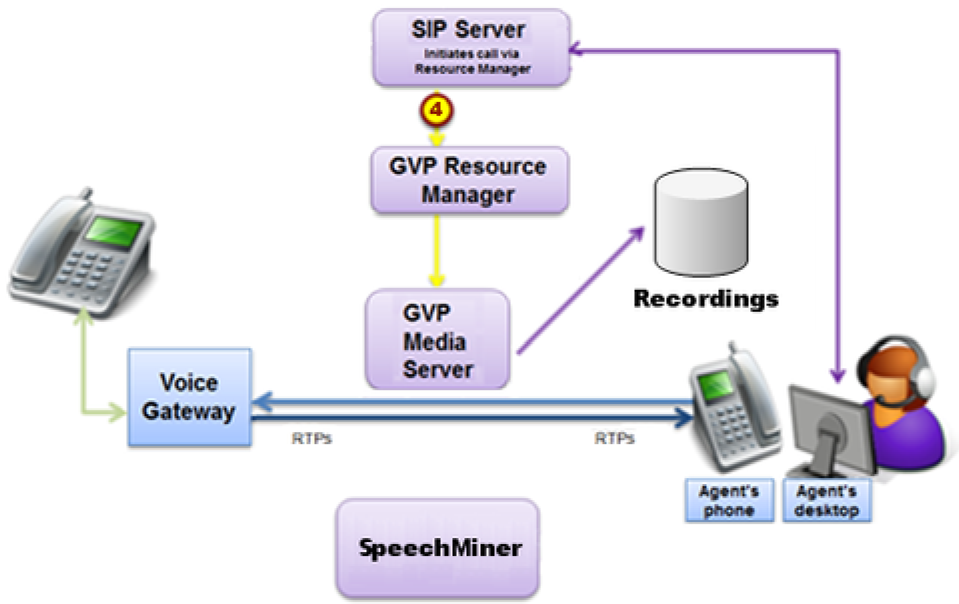
Call Recording Initiated

There are four ways call recording can be initiated. Once started, the recording process is the same, regardless of how it was initiated.
One of the following initiates recording:

1. SIP Server initiates recording itself, for example, because a DN is configured to always be recorded.
2. SIP Server informs third party Call Recording of a call with a DN that Call Recording supports. Third party Call Recording has a recording rule for the DN. Third party Call Recording evaluates the rule, determines that the DN must be recorded and requests recording by sending the T-lib request, `TrequestPrivateService`, to SIP Server.
3. A third party, for example an agent desktop, requests call recording by sending the T-lib request, `TrequestPrivateService`, to SIP Server.
4. A recording can be initiated by a Routing Strategy (extension `record=source` in `TrouteCall`).

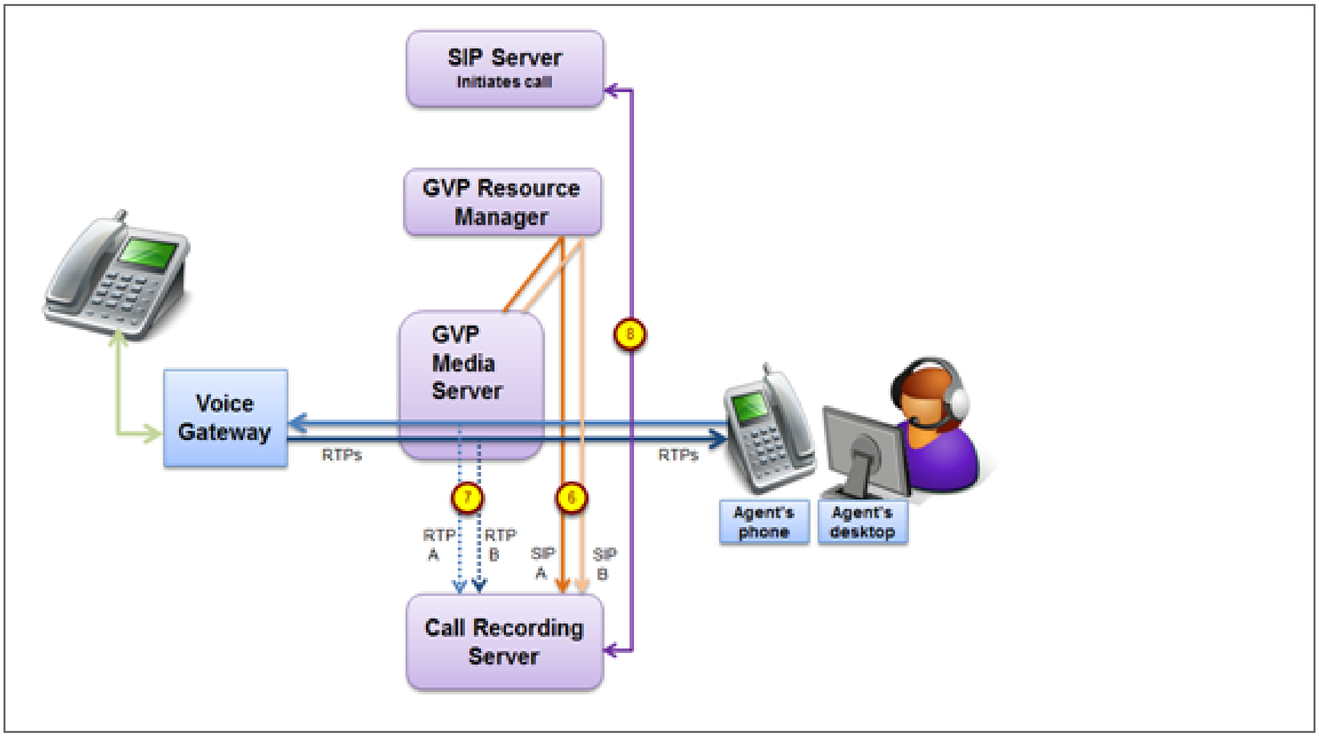


SIP Server Initiated Call Recording



5. Using media control, SIP Server invites Media Server to replicate the RTPs (two invites, one for each RTP stream).

Media Server Replicates RTPs

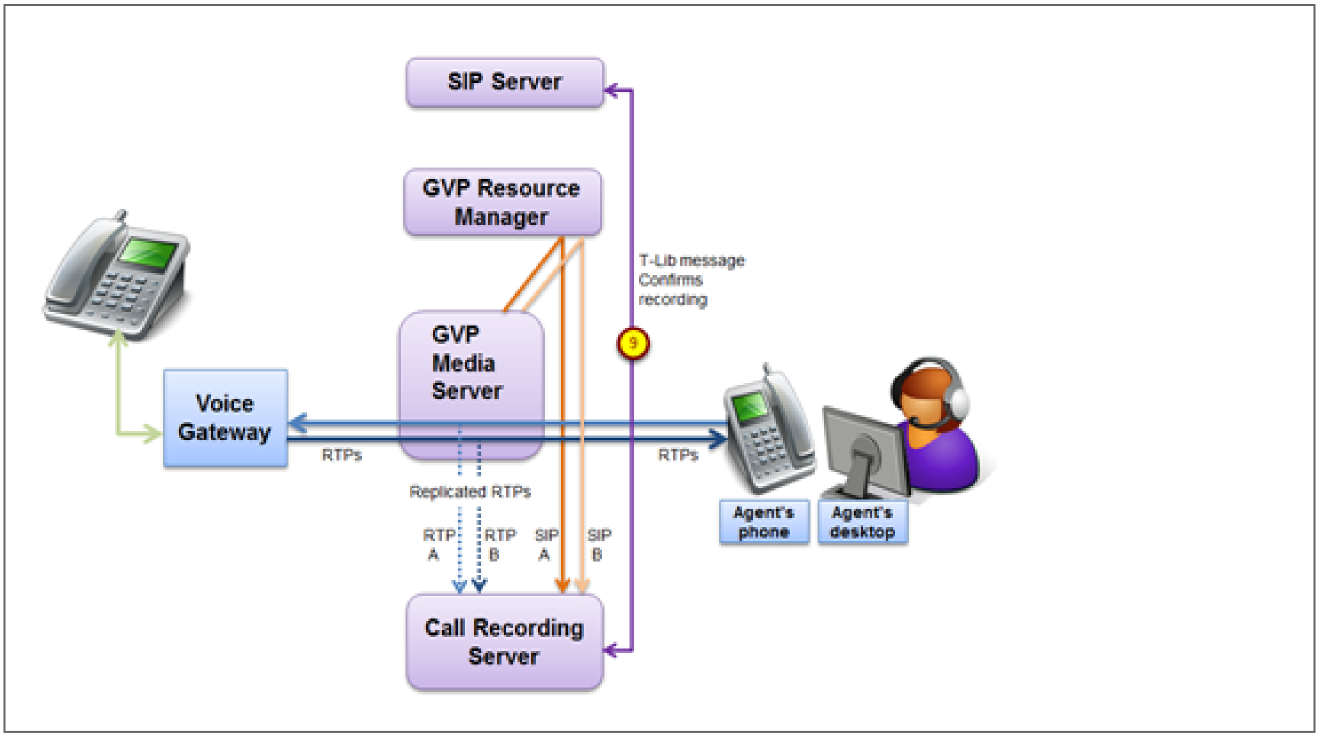


6. The Media Server replicates the RTP streams and:

- Sends the SIP invite messages to the Call Recording server (two invite messages, one for each RTP stream).
- Sends the RTPs to the Call Recording Server.

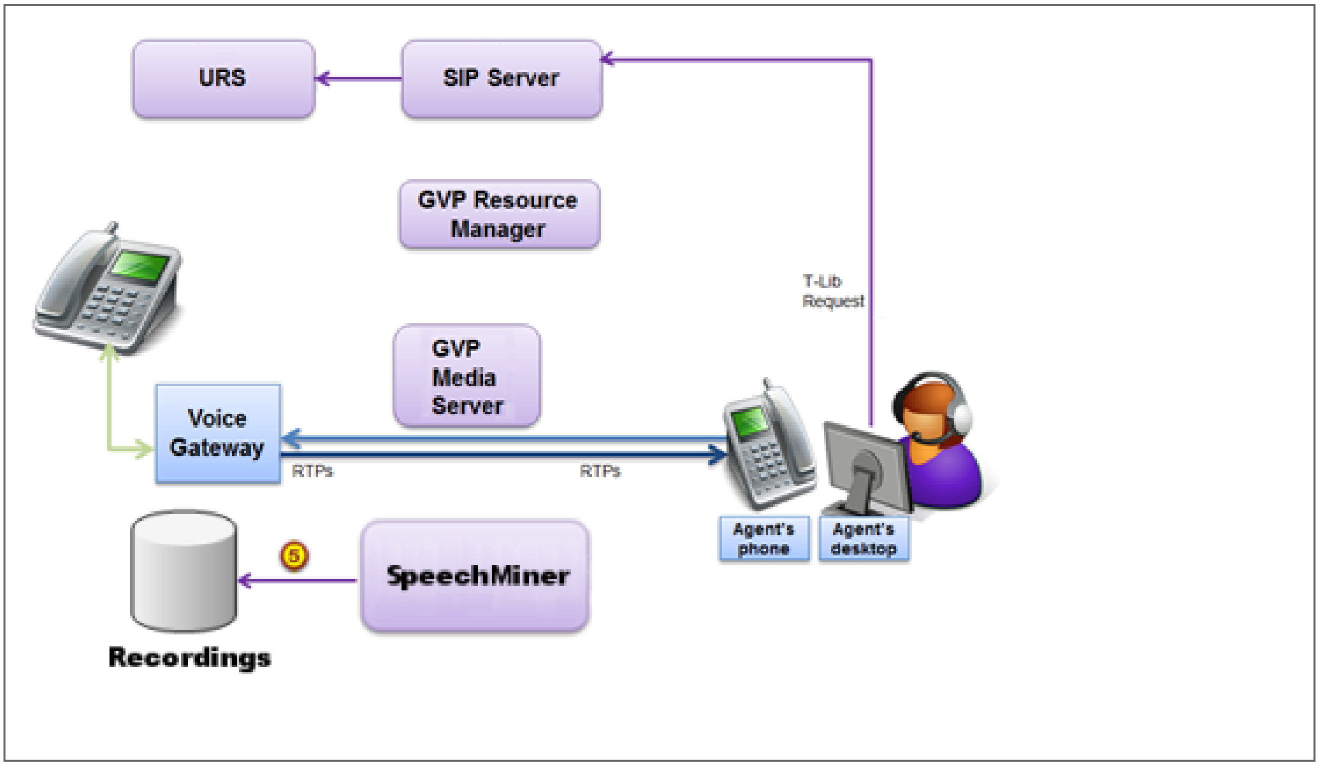
7. Recording starts. 8. The Call Recording server requests additional information, such as user attached data with T-Lib.

Recording Starts



9. At any time the Call Recording Server or a third party (for example, the agent desktop) can use a `TrequestPrivateService` message for pause, resume, start and stop.

Recording Ends

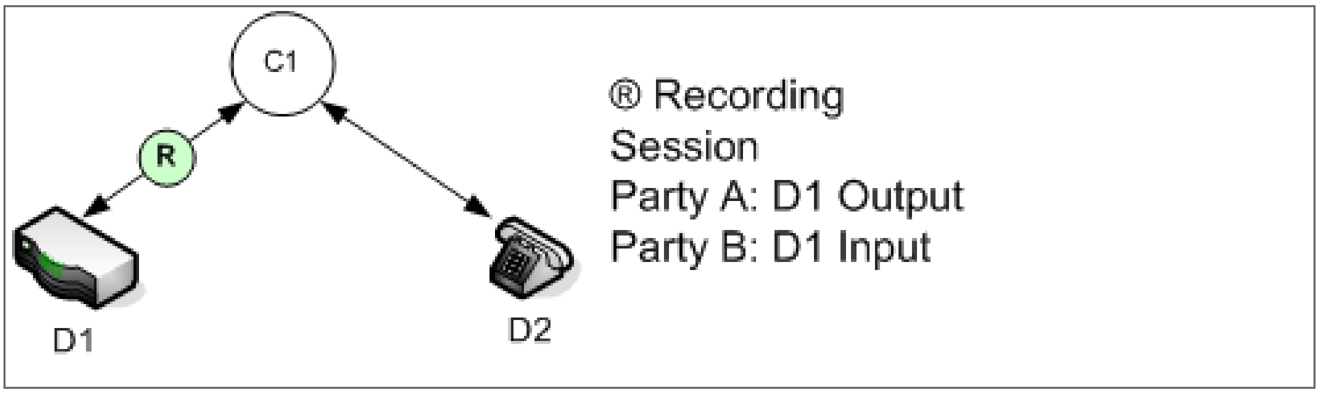


10. SIP messages from each stream indicate that the call has ended. The Call Recording Server stops recording.

Call Recording Model

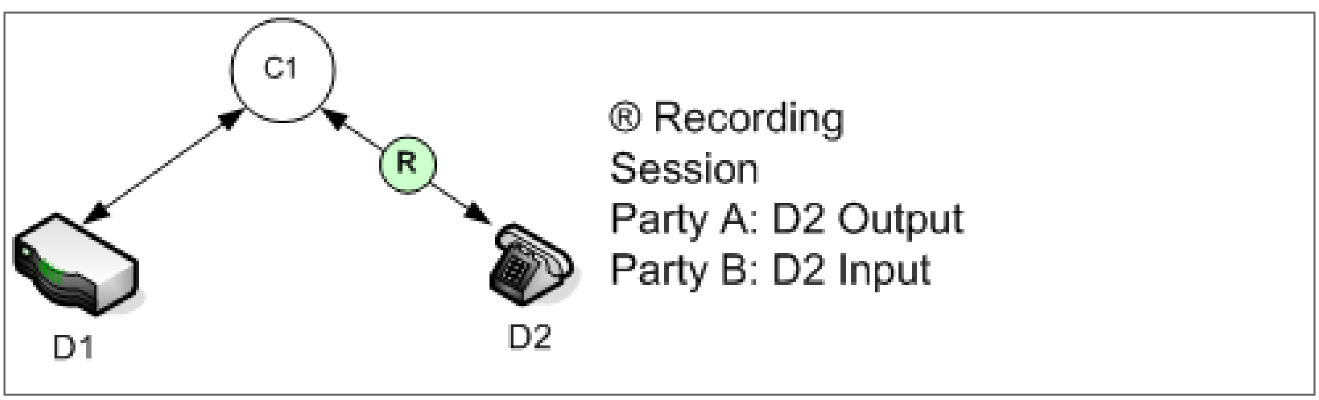
The basic model for initiating call recording is based on SIP Server connection model. Based on configuration or T-Lib functions, you can choose to have recording initiated from the origination device or the termination device. This is also known as trunk-side recording (origination device) or agent-side recording (termination device). The models are shown below:

Basic Model



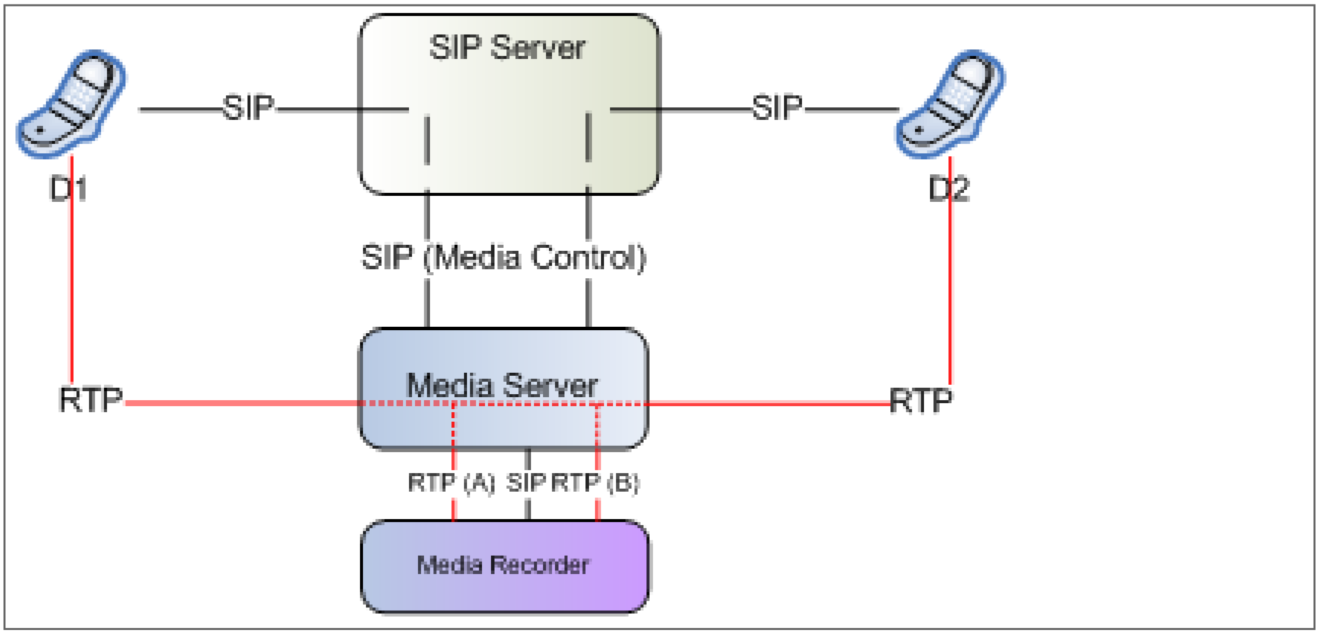
Recording from an Originating Device.

Basic Model



Recording from a Terminating Device.

Basic SIP/RTP Model



When call recording is requested, SIP Server invites a pair of call legs to GVP Media Server to perform the call recording by bridging the media between the devices through Media Server. Media Server may optionally split the media to a third party recorder for recording or real-time analysis.

Feature Configuration

The following sections describe the steps required to configure the Active Recording Ecosystem features.

- Call Flows
- Events and Models
- Recording
- Scalability and High Availability
- Interfaces
- Integrating with Third Party Vendors
- Geo-Location
- Audio Tones

Call Flows

This section describes the following call flow types:

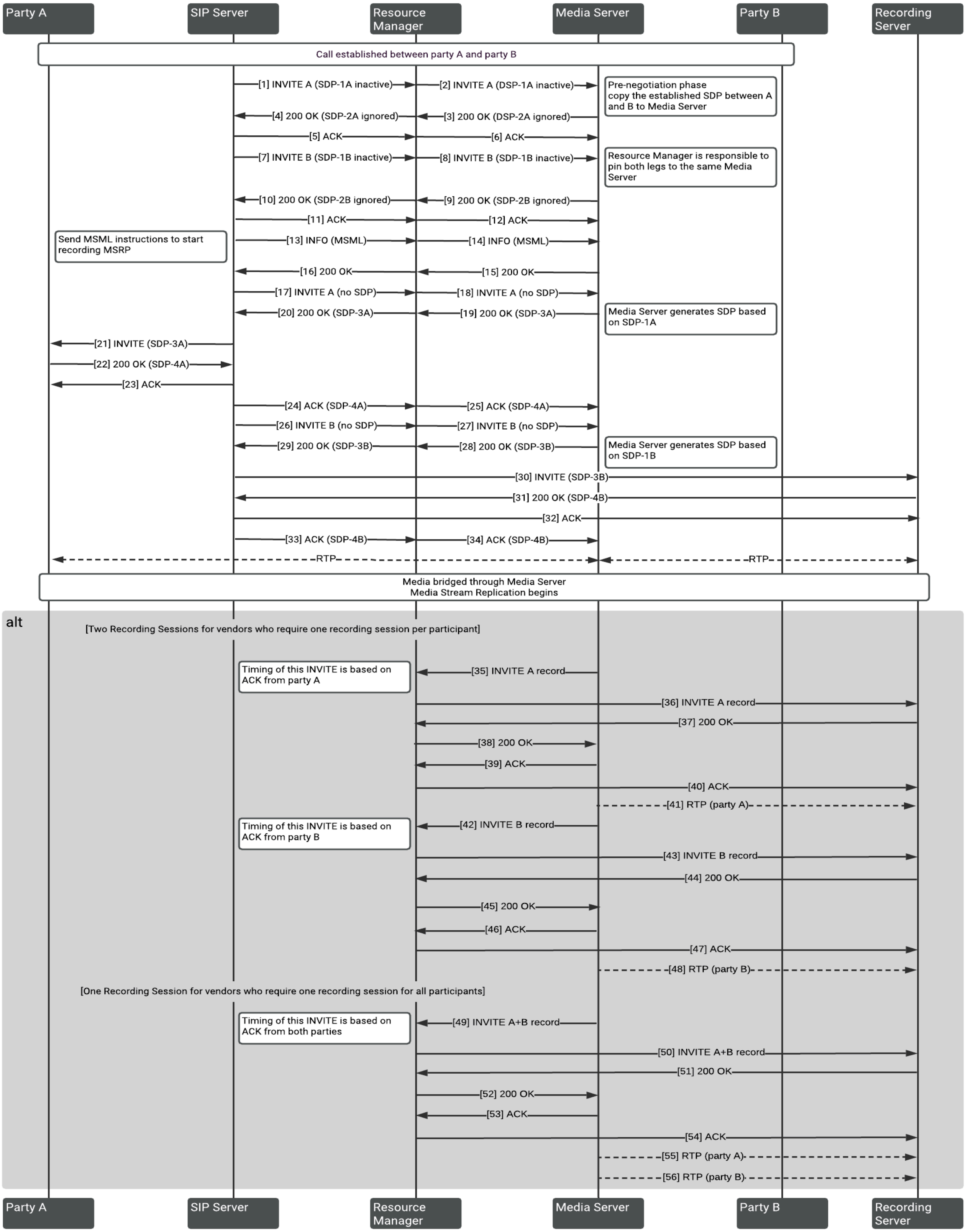
- Basic Call Flow
- Transfers
- Conference Calls
- Consultation Calls

Basic Call Flow

After Party A and Party B are connected and a recording request is made to SIP Server, SIP Server will initiate two sessions, one session for each party, to Media Server. SIP Server first INVITEs with the Session Description Protocol (SDP) offer from the connected parties to Media Server, and a second reINVITE to Media Server to get an SDP offer from Media Server. The offer from SIP Server is sent to the connected parties to proxy the media through Media Server. Once the media is established, Media Server initiates the media stream replication process and sends an INVITE to the recording server through Resource Manager. Depending on the configuration of the recording server, there are two ways to establish the recording session:

- Two sessions—One SIP recording session representing each participant in the call, meaning Media Server will initiate two INVITE requests to the recording server.
- One session—One SIP recording session containing all the media streams of both participants in the call.

The following diagram describes the call flow:

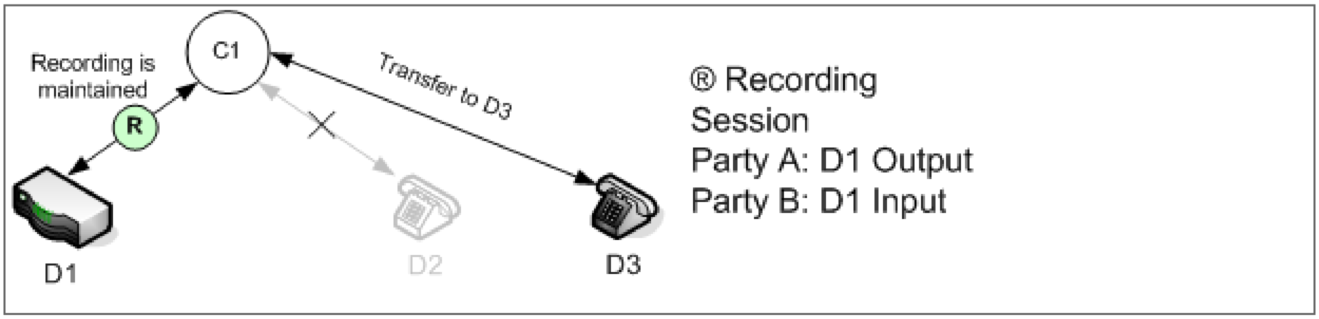

Party A
SIP Server
Resource Manager
Media Server
Party B
Recording Server
Call established between party A and party B
[1] INVITE A (SDP-1A inactive)
[2] INVITE A (DSP-1A inactive)
Pre-negotiation phase copy the established SDP between A and B to Media Server
[4] 200 OK (SDP-2A ignored)
[3] 200 OK (DSP-2A ignored)
[5] ACK
[6] ACK
[7] INVITE B (SDP-1B inactive)
[8] INVITE B (SDP-1B inactive)
Resource Manager is responsible to pin both legs to the same Media Server
[10] 200 OK (SDP-2B ignored)
[9] 200 OK (SDP-2B ignored)
[11] ACK
[12] ACK
Send MSML instructions to start recording MSRP
[13] INFO (MSML)
[14] INFO (MSML)
[16] 200 OK
[15] 200 OK
[17] INVITE A (no SDP)
[18] INVITE A (no SDP)
[20] 200 OK (SDP-3A)
[19] 200 OK (SDP-3A)
Media Server generates SDP based on SDP-1A
[21] INVITE (SDP-3A)
[22] 200 OK (SDP-4A)
[23] ACK
[24] ACK (SDP-4A)
[25] ACK (SDP-4A)
[26] INVITE B (no SDP)
[27] INVITE B (no SDP)
[29] 200 OK (SDP-3B)
[28] 200 OK (SDP-3B)
Media Server generates SDP based on SDP-1B
[30] INVITE (SDP-3B)
[31] 200 OK (SDP-4B)
[32] ACK
[33] ACK (SDP-4B)
[34] ACK (SDP-4B)
RTP
RTP
Media bridged through Media Server
Media Stream Replication begins
alt
[Two Recording Sessions for vendors who require one recording session per participant]
Timing of this INVITE is based on ACK from party A
[35] INVITE A record
[36] INVITE A record
[37] 200 OK
[38] 200 OK
[39] ACK
[40] ACK
[41] RTP (party A)
Timing of this INVITE is based on ACK from party B
[42] INVITE B record
[43] INVITE B record
[44] 200 OK
[45] 200 OK
[46] ACK
[47] ACK
[48] RTP (party B)
[One Recording Session for vendors who require one recording session for all participants]
Timing of this INVITE is based on ACK from both parties
[49] INVITE A+B record
[50] INVITE A+B record
[51] 200 OK
[52] 200 OK
[53] ACK
[54] ACK
[55] RTP (party A)
[56] RTP (party B)
Party A
SIP Server
Resource Manager
Media Server
Party B
Recording Server

Transfers

This section describes the types of transfer that the Active Recording Ecosystem supports.

Origination Device

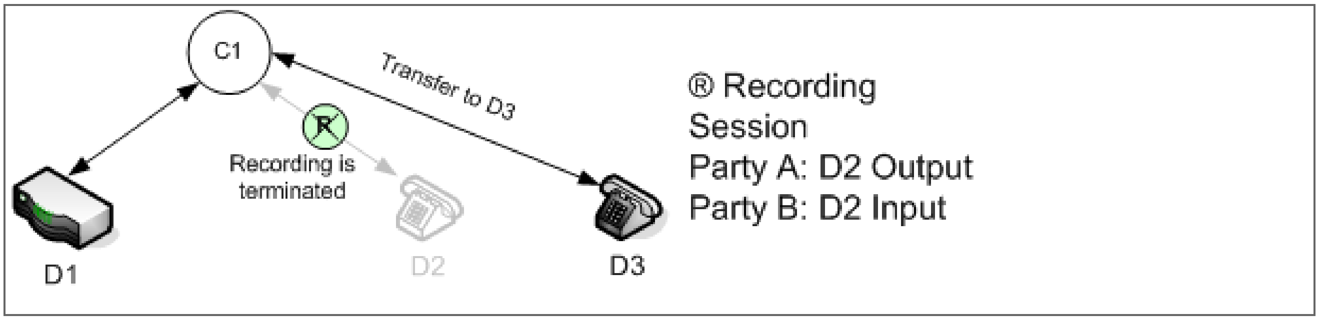
Recording from the Origination Device



When a call that is being recorded is transferred to another party, the recording can be affected differently depending on where the recording is initiated. The reason for differentiating the side of the recording is that call recording is "sticky" to the side of the connection that is chosen for recording. When the connection needs to transfer the call to another device, the call recording stays with the device. For example, if the connection is transferred from D2 to D3, the call recording is maintained if recording is initiated from the origination device, while the recording is terminated if the recording is initiated from the terminating device.

Termination Device

Recording from the Termination Device



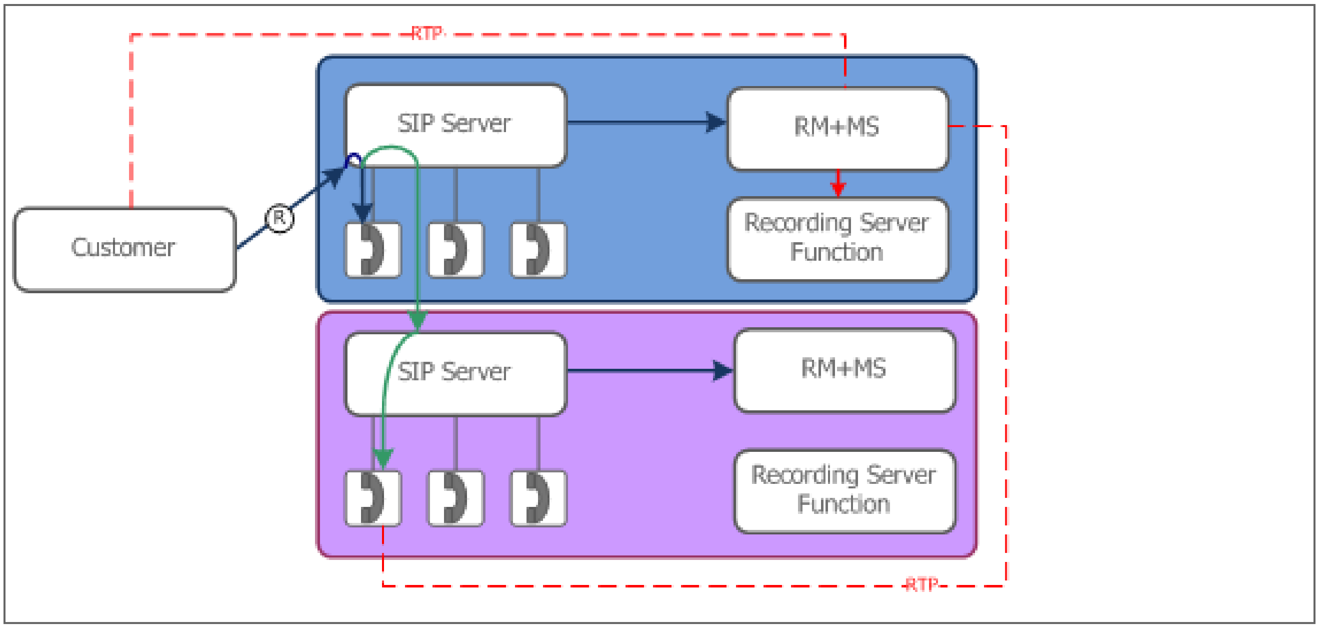
When the call is transferred to D3, the Recording Session is maintained and should expect a reINVITE to re-negotiate the media between D1 and D3. The media control dialog between SIP Server and Media Server is also maintained by only sending reINVITEs to the media control dialog.

Multi-site Transfers

There are a few models for transfers across SIP Server instances, depending on where the recording is started and how transfer is performed. SIP Server may stay on the signaling path when transferring a call to the remote SIP Server using reINVITE, or SIP Server may put itself out of the signaling path using REFER (set the SIP Server `oosp-transfer-enabled` option). When a call is transferred from one T-Server (SIP Server) to another T-Server instance, their local call identifiers, connection id, will be different on the different T-Servers. The customer interaction is identified by a common identifier CallUUID and this identifier is maintained after the call is transferred from one site to another.

Transfer with reINVITE

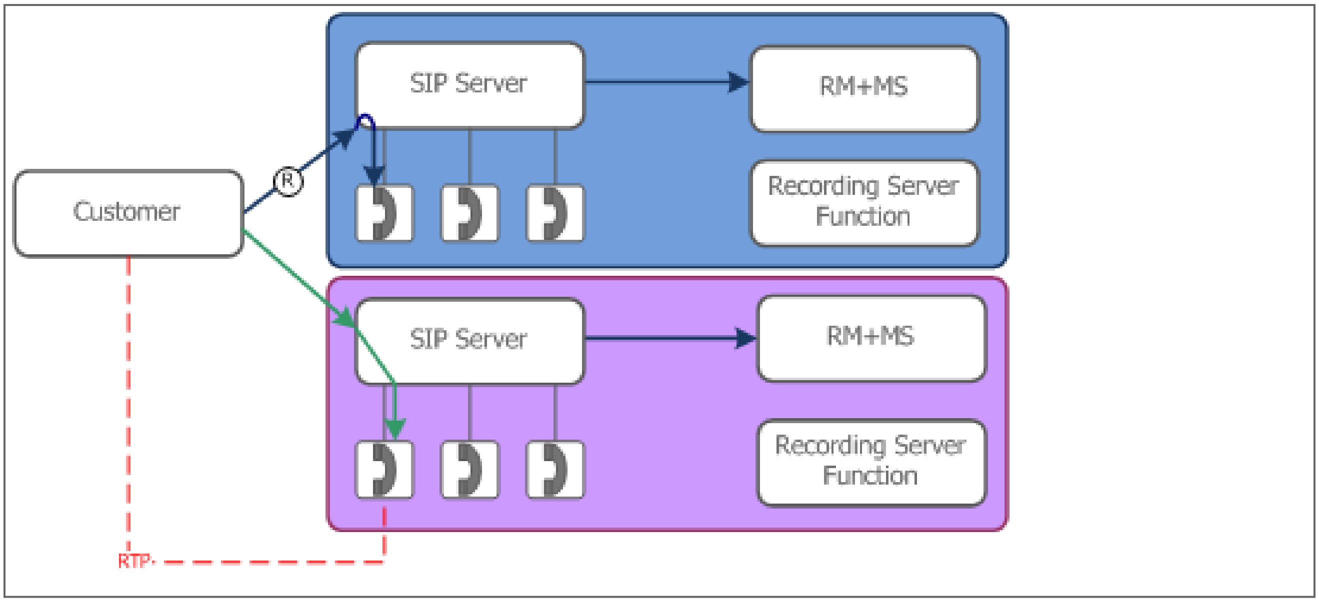
Customer recording, transfer with reINVITE



The agent on SIP Server (blue) transfers the call to the agent on SIP Server (purple) using reINVITE. Since the recording is enabled on the customer side, the recording remains in the call path, and the media pinned up in the blue site. Call recording is serviced by the recording server in the blue site as well.

Transfer with REFER

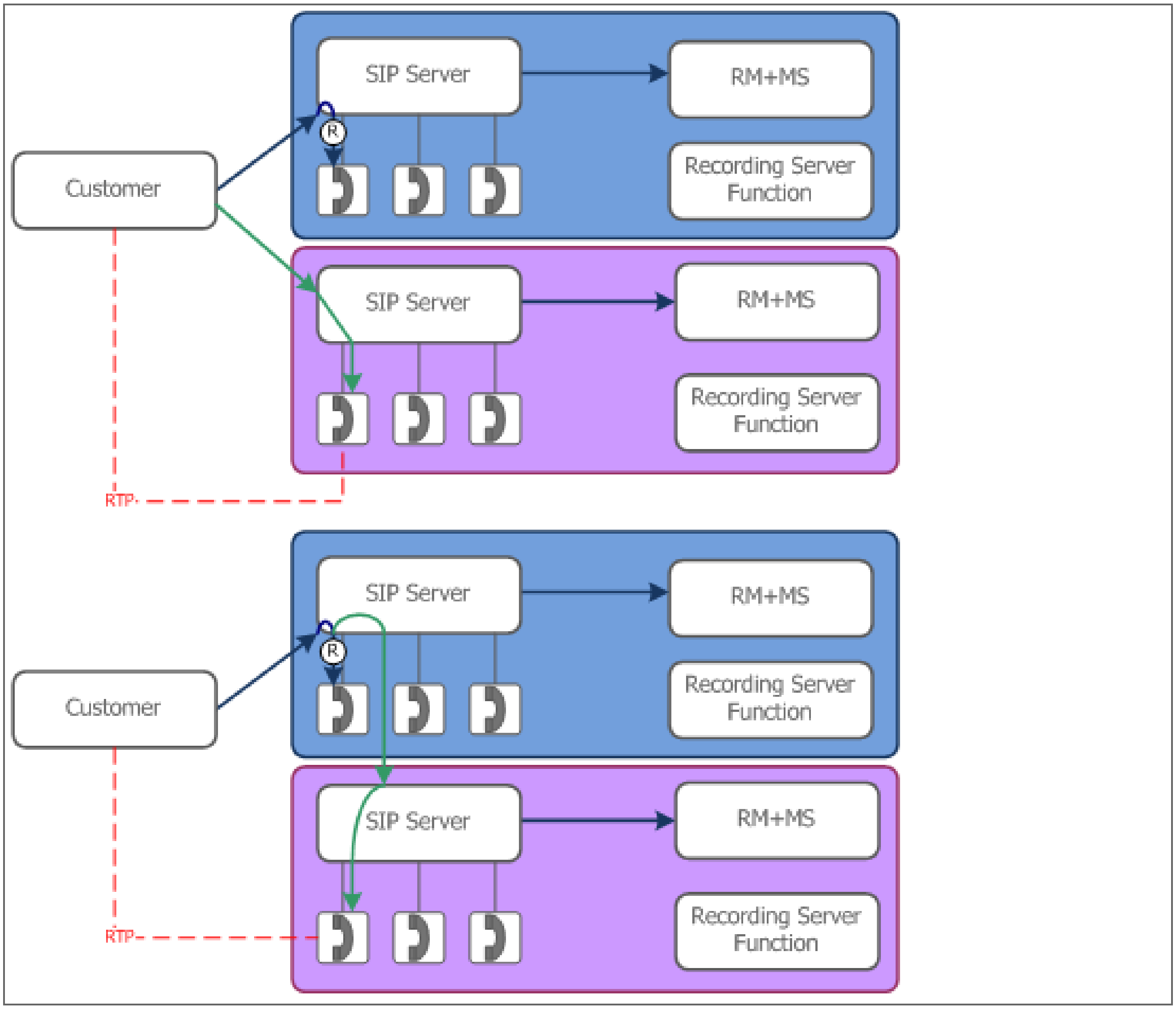
Customer recording, transfer with REFER



When the call is transferred from the blue site to the purple site with REFER transfer, the existing call recording will terminate. The transferred call coming into SIP Server (purple) is treated as an incoming call, and recording is not automatically enabled. Recording may be started again through the usual means in the purple site, but this is not automatic. As a future extension, the transferred call coming into SIP Server (purple) may contain a SIP extension to note that the call should be recorded, without the need to rely on a business rule to trigger recording again on the transferred site. This allows consistency of selection when selective recording is used – meaning when the call on the blue site is being recorded, the call should be recorded after the call is transferred as well. As a workaround, it is possible that a T-lib client can attach some user data into the call to indicate that the call is being recorded, and when the call is transferred to the purple site, another T-lib client can look for this user attached data for the recording indication.

Agent Recording

Agent Recording

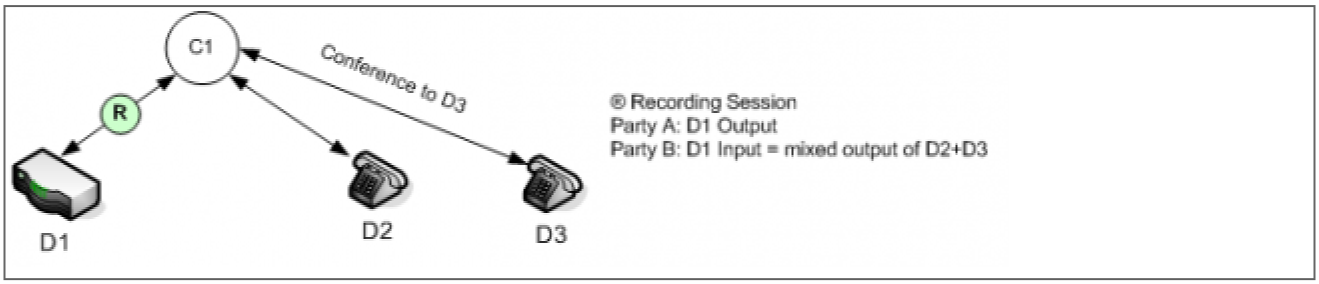


When agent recording is enabled and the call is transferred to another SIP Server, the agent recording is automatically terminated. This is basically the same as depicted in the Recording from the Termination Device figure, for a transfer to another agent within the same SIP Server. The type of transfer is irrelevant in this case since both will yield the same result. For both cases illustrated below, if the recording is to be started on the transferred agent, the media will pinned up on the purple site and the recording server in the purple site will be recording the media.

Conference

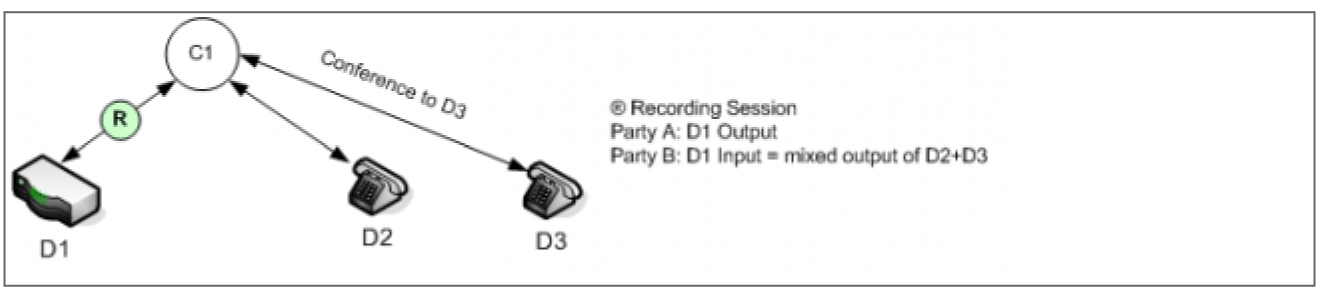
When a conference or supervisor monitoring requests to start recording, a participant can request the conference to be recorded through the recording service. You can reuse the existing SIP Server call structure and treat recording and conference as separate services, and the participant requesting recording will be recording the output of the conference.

Recording Conference



When a third party recorder is used, the recorder will be recording a mixed output of the customer and agent, plus a separate stream of the supervisor.

SIP Dialog



The structure of the SIP dialog when recording a conference.

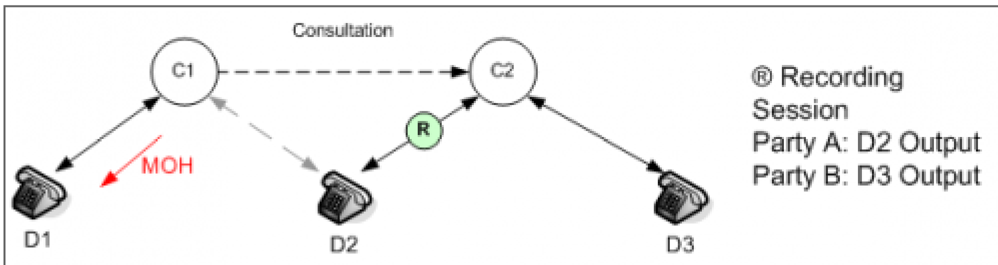
Consultation

When the agent initiates a consultation call and call recording is enabled on the agent DN, the call recording to record the consultation session as well is allowed. This is recognized as a single-dialog consultation mode where there is only a single active SIP dialog on the device. Set the DN `dual-dialog-enabled` option to `false` to allow recording of consultation calls. By default the value is set to `true` so consultation calls are not recorded. The following diagrams illustrate this scenario.

Before Consultation

The initial call when the customer (D1) is talking to the agent (D2).

During Consultation



When the agent initiates a consultation to the supervisor (D3), the existing SIP dialog is retained and so is the Recording Session.

Note: As a current limitation for consultation calls, recording is not available on the consulted party, so a Recording Session cannot be started on D3.

Events and Model Reference

In addition to capturing the call recording media, recording servers also receive call events from Genesys SIP Server or T-Servers. The Genesys Events and Models Reference provides a detailed guide on capturing events related to a call or agent events.

There are a few important notes related to multi-site call scenarios to clarify how to reference all call events related to the customer interaction across multiple T-Server instances.

1. The `CallUUID` is a unique call identifier generated on each T-Server instance (or SIP Server instance). When a call is transferred from one T-Server to another, a new `CallUUID` is generated on the other T-Server.
2. The `ConnID` is an identifier assigned to reference a call to initiate requests. In some cases, the `ConnID` is maintained across multi-site transfers. However, the `ConnID` is not guaranteed to be unique across multi-site transfers in some cases.

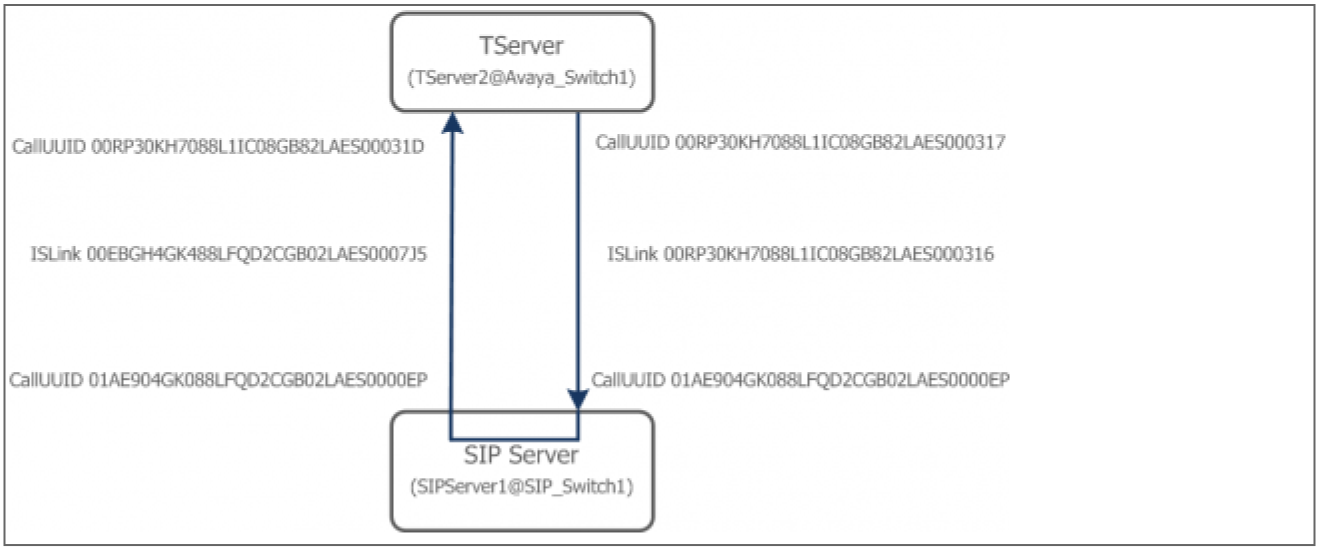
Genesys offers a mechanism called Inter-Site Link (IS-Link) to build a linked list of call identifiers for the customer interaction across multiple sites. The following is a short overview of how the mechanics work; for full details please refer to the "Multi-Site Call Scenarios" section in the Genesys Events and Models Reference.

When a multi-site call transfer is initiated, it is important to look for the `AttributeISLinkList` attribute provided by the `EventCallDataChanged` event on both of the T-Servers. In order to get call-based events, use the `RequestStartCallMonitoring` event to register for call-based events. When the call is being transferred out of a T-Server, the `EventCallDataChanged` event provides an additional attribute called `AttributeISLinkList`. This is a globally unique identifier that is linked to the destination T-Server. Use this ISLink identifier to locate the `CallUUID` associated with each T-Server instance, so that the entire call can be traced through all transfers across switches.

The `AttributeISLinkList` attribute provides two more keys:

- `location-name`—The switch name of the remote T-Server.
- `direction-role`—Either `source` or `target`. If set to `source`, this T-Server is initiating the transfer. If set to `target`, this T-Server is receiving the transfer.

ISLink



An example of a call transfer that results in a change of `ConnID` on T-Server. The ISLink can provide a way to link the `CallUUID` between different switches.

The call is initially on Tserver2, and transfers the call to SIPServer1.

The `EventCallDataChanged` event on Tserver2:

```
@09:41:00.5404 [0] 8.1.001.05 distribute call/party event: message EventCallDataChanged
               AttributeEventSequenceNumber            0000000000057726
               AttributeTimeStamp      508e957c00083f40
               AttributeConnID           009a021843038407
               AttributeCallUUID          '00RP30KH7088L1IC08GB82LAES000317'
               AttributeFirstTransferConnID     009a021843038407
               AttributeLastTransferConnID     009a021843038407
               AttributeISLinkList          [108] 00 01 03 00..
                               '00RP30KH7088L1IC08GB82LAES000316'(list) 'location-
name'        'SIPServer1@SIP_Switch1'
                                                                       'direction-
role'         'source'
```

When the call is received on SIPServer1, the `EventCallDataChanged` shows a common ISLink. This event provides the `CallUUID` and `ConnID` for this call on SIPServer1. Note that the `ConnID` event is common between the two T-Servers at this point.

```
@10:41:00.4955 [0] 8.1.001.03 distribute call/party event: message EventCallDataChanged
               AttributeEventSequenceNumber            000000000002d478
               AttributeTimeStamp      508e957c00078fc6
               AttributeConnID           009a021843038407
               AttributeCallUUID          '01AE904GK088LFQD2CGB02LAES0000EP'
               AttributeISLinkList          [108] 00 01 03 00..
                             '00RP30KH7088L1IC08GB82LAES000316'(list) 'location-name'
```

```
'Tserver2@Avaya_Switch1'
                                                                      'direction-
role'         'target'
```

SIPServer1 then transfers the call back to Tserver2. The `EventCallDataChanged` event shows the ISLink list as the second item in the list.

```
@10:41:03.5465 [0] 8.1.001.03 distribute call/party event: message EventCallDataChanged
               AttributeEventSequenceNumber            000000000002d4c3
               AttributeTimeStamp      508e957f000856fe
               AttributeConnID              009a021843038407
               AttributeCallUUID           '01AE904GK088LFQD2CGB02LAES0000EP'
               AttributeISLinkList           [214] 00 02 03 00..
                               '00RP30KH7088L1IC08GB82LAES000316'(list) 'location-
name'        'Tserver2@Avaya_Switch1'
                                                                      'direction-
role'         'target'
                              '00EBGH4GK488LFQD2CGB02LAES0007J5'(list) 'location-name'
'Tserver2@Avaya_Switch1'
                                                                      'direction-
role'         'source'
```

Tserver2 now receives the `EventCallDataChanged` evemt with the ISLink. Note that a new `CallUUID` and `ConnID` is assigned for this call.

```
@09:41:02.9674 [0] 8.1.001.05 distribute call/party event: message EventCallDataChanged
               AttributeEventSequenceNumber            000000000005775c
               AttributeTimeStamp      508e957e000ec324
               AttributeConnID              009a021843038407
               AttributeCallUUID           '00RP30KH7088L1IC08GB82LAES00031D'
               AttributeISLinkList           [108] 00 01 03 00..
                               '00EBGH4GK488LFQD2CGB02LAES0007J5'(list) 'location-name'
'SIPServer1@SIP_Switch1'
                                                                      'direction-
role'         'target'
```

Genesys Voice Platform Recording

Genesys Voice Platform (GVP) provides various media services beyond IVR, and depending on the usage, GVP can be configured as various different DN types on SIP Server. The following table shows the list of DN types and whether recording is supported for each DN type.

DN Type	Usage	Can be recorded by SIP Server	Generates Tevents
VoIP Service	Media services such as music-on-hold, conferencing, call parking, recording	No	No
Voice Treatment Port	Legacy IVR ports for both inbound and outbound IVR calls	Yes	Yes
Trunk Group DN	Call Progress Detection (CPD) and proactive notification (outbound GVP IVR calls)	Yes	Yes
Trunk DN	Inbound GVP IVR calls	No	No

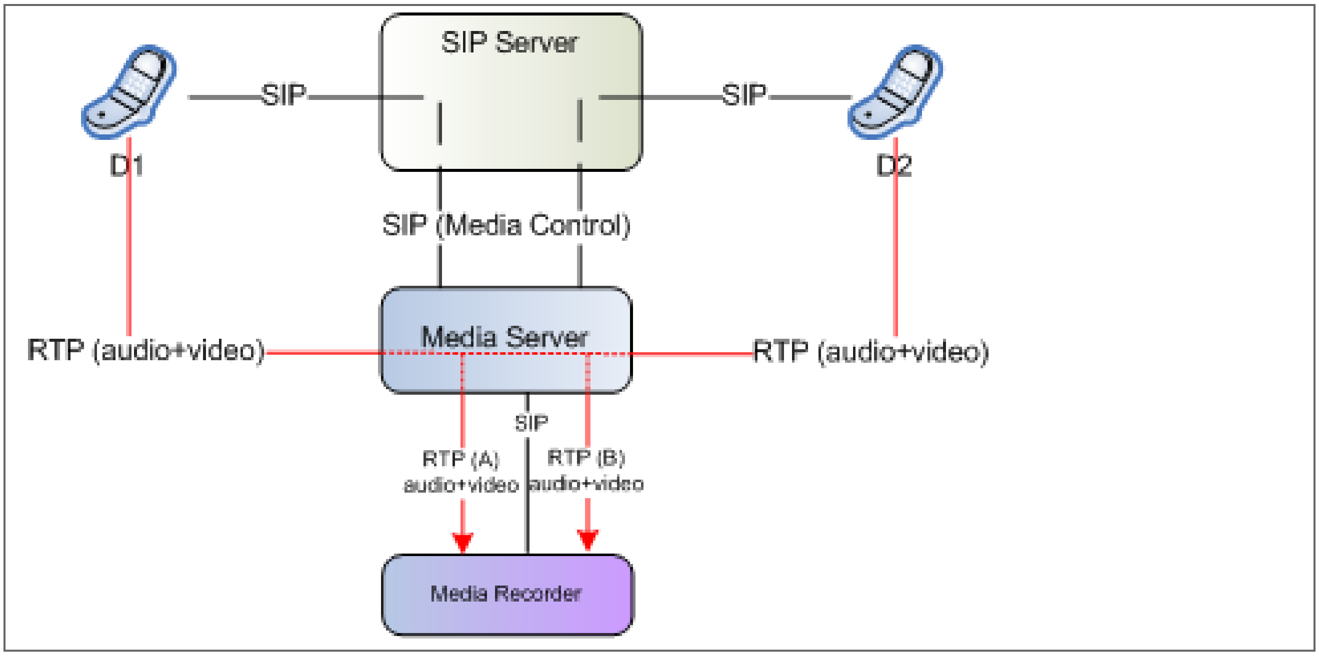
To achieve dynamic recording controls or recording rules, you must configure a Trunk Group DN for inbound IVR. However, note that Trunk Group DNs do not support prefix matching, so you will need to create one Trunk Group DN for each DID for the IVR.

Configuration

1. Create a DN with the type Trunk Group.
2. In the Annex tab, create a section named TServer, and configure the following options and values:
    - contact—sip:msml@<RMHost>:<RMPort>
    - refer-enabled—false
    - make-call-rfc3725-flow—1
    - ring-tone-on-makecall—false
    - request-uri—sip:msml@<RMHost>:<RMport>;gvp-tenantid=<Tenant_Name>
    - subscription-id—<the name of this Trunk Group DN>
    - cpd-capability—mediaserver

Video Recording

Video Recording



SIP Server currently supports video calls. In order to support recording of video streams, SIP Server must connect both audio and video streams to Media Server. Media Server will bridge both audio and video streams between the participants and replicate both audio and video streams towards the media recorder. This diagram is effectively the same as the basic call recording model.

Since Media Server is modeling call recording internally as a conference, Media Server must treat a video call as a simple video-switching type conference. This means the parties are seeing each other and no video rendering is done by Media Server.

While the basic video recording model is straightforward, there are a few additional considerations and design requirements:

- A client of SIP Server such as the media recorder or agent desktop may make requests for specific media streams to be recorded. The client may choose either audio only or audio+video streams. Media Server acknowledges the request and will only offer the requested media streams to the media recorder. For example, if the client only asks for audio only, then Media Server will only send audio stream in the SDP offer in the recording session to the media recorder.
- The media recorder may have its own local policy to determine which media streams to record if Media Server decides to offer both audio and video in the recording session. The media recorder may only want to store audio streams; to do so, the media recorder sends the SDP answer to accept the audio media stream by setting a non-zero port and reject the video media stream by setting a zero port.

- The client of SIP Server needs to get separate recording indication events for audio recording and video recording.
- For delivering inband indications for video recording, it can either be an explicit announcement before entering the recording, or provide visual indication continuously for the duration of the recording. The mechanism for handling the latter part is undefined for now, but it is expected that Media Server will need to provide rendering of the visual indication for the duration of the recording.
- When delivering the recorded media streams over a single SIP recording session, it will end up having four SDP `m=` lines in the SDP offer. The mapping of media streams to the devices is explained in the Interfaces section.

Scalability and High Availability

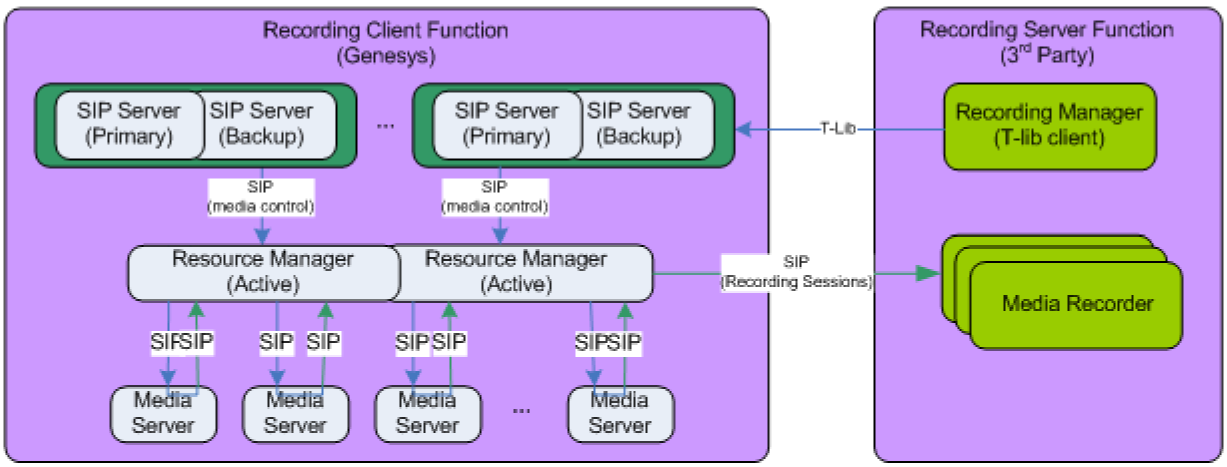
This topic discusses options/possibilities for scalability and high availability for the Active Recording Ecosystem. It contains the following sections:

- Scalability of Recording Client Function
- Scalability of Recording Server Function
- High Availability of Recording Server Function

Scalability

Scalability of Recording Client Function

The Recording Client Function is designed to be scalable, and it works the same way as a scalable SIP Server + Media Server deployment. The goal is to allow SIP Server to use the same unified Media Server instances to provide any new media services, and in this case, media stream replication to a recording server. The following image shows the basic scalable SIP Server + Media Server deployment along with the Recording Server Function.



In a single deployment, there are two Resource Managers running in active/backup or active/active pair. Resource Manager is responsible to route requests to the cluster of Media Servers by managing the capacity of each Media Server in the cluster. When there are multiple instances of SIP Servers, each SIP Server will configure a VoIP Service DN (`type=recorder`) to use the same pair of Recording Managers for all recording functions.

Resource Manager is also responsible to handle scalability issues at the recording server function. The scalability model for each recording vendor is different and will be discussed separately for each target deployment. Note that the recording client function and recording server function can be scaled independently so the above diagram allows SIP Server + Media Server to work with any type of recording vendor.

Scalability of Recording Server Function

The third party recorder can scale the number of simultaneous recording session by deploying a cluster of call recorders. GVP Resource Manager acts as a SIP Proxy to route recording sessions coming from Media Server to a call recorder instance from a pool of recorder instances, typically through round-robin fashion. Resource Manager can monitor recorder instances through OPTIONS pinging to ensure recording sessions are only forwarded to an available recorder instance. Resource

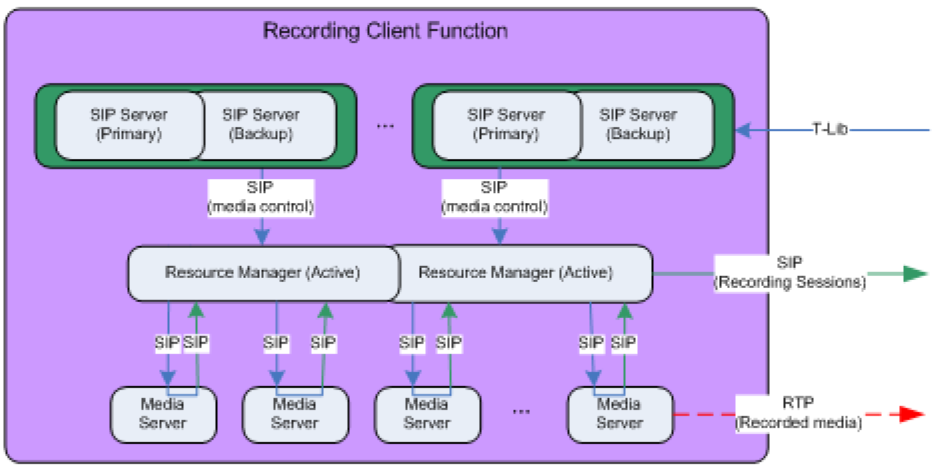
Manager can also provision the recorder instances with capacity limits to ensure that the recorder instances are not overloaded.

The T-lib clients can be scaled independently of the call recorder since recorder and T-lib clients have different connections.

High Availability

Recording Client Function

The high availability model for the Recording Client Function is essentially the same as the high availability model for Genesys SIP Server and Genesys Media Server.



Failover of SIP Server

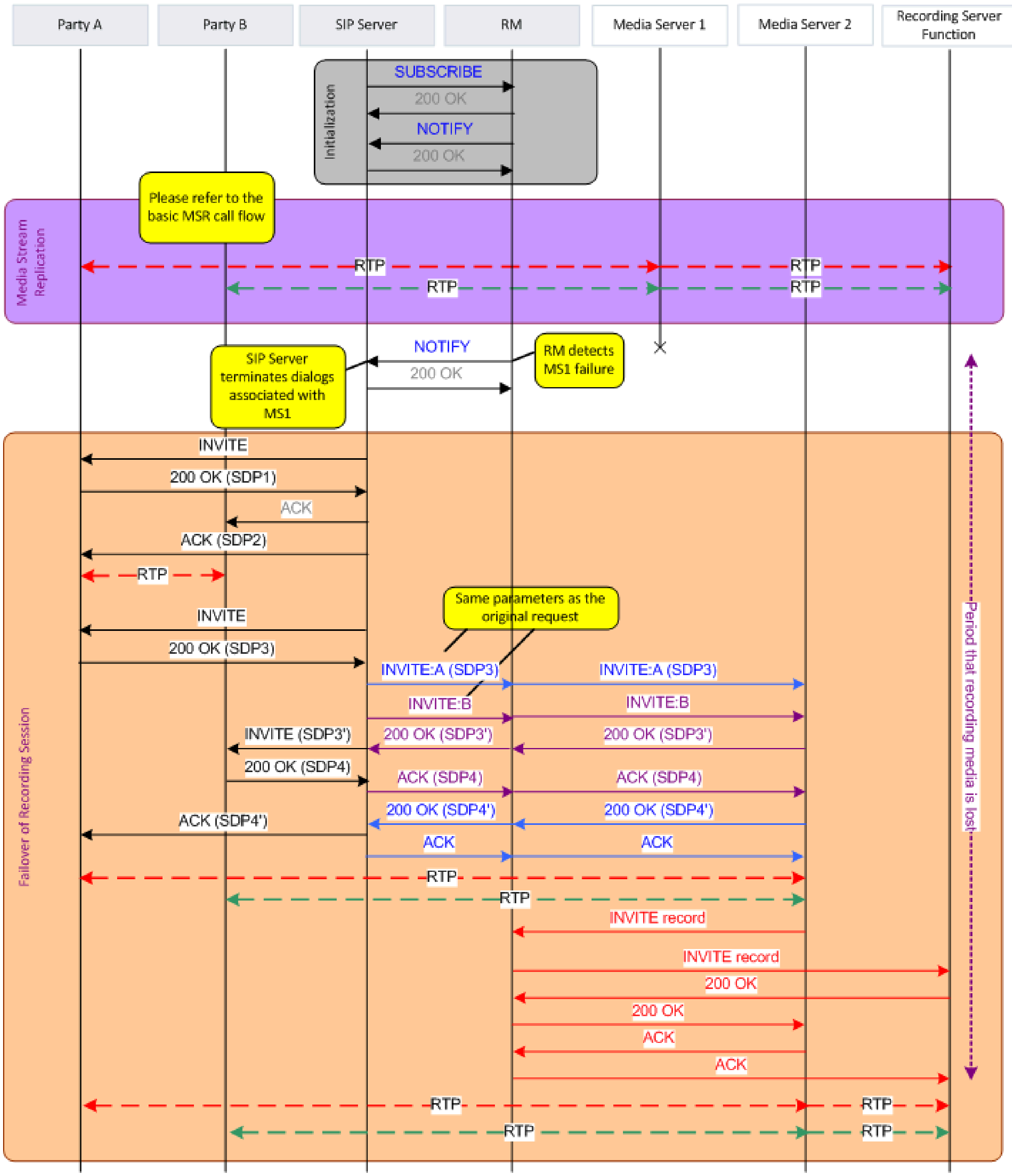
SIP Server can be deployed in an active/hot-standby pair; whenever the primary instance fails, the hot-standby instance will take over and will have knowledge of all established sessions. If a recording session has been established with a recording server, a failure of SIP Server will not affect the operation of the communication and the recording session.

Failover of Resource Manager

Resource Manager is a SIP Proxy that can operate in either active/hot-standby or active/active pair. When there is a failure at the Resource Manager, the remaining instance of Resource Manager will become active and continue to accept incoming requests. The remaining instance will remember the affinity of the recording sessions with Media Server as well as the recording server. The failure should be transparent to SIP Server, Media Server, and the recording server function.

Failover of Media Server

When the media is bridged through Media Server, Media Server becomes a single point of failure for the duration of the communication session. If Media Server fails, Resource Manager notifies SIP Server about the failure so that SIP Server can take alternative action on the call. SIP Server should re-establish the recording session by joining the endpoints to another available Media Server. When the endpoints are joined to the now-active Media Server, Media Server will establish a new recording session with the same parameters to the recording server function.



The total amount of recorded media lost is the sum of time for Resource Manager to detect MS1 failure plus the amount of time to failover the recording session to another Media Server instance.

Recording Server Function

Handling high availability of the Recording Server Function is the responsibility of the third party recorder. The following are typical ways to manage failover of a component in Recording Server Function, and Genesys does not restrict how third party manages high availability of Recording Server Function.

Failure of a call recorder:

- Separate signaling (SIP) and media recording (RTP) in the call recorder. Whenever there is a failure of signaling, a backup instance can take over without affecting the recording session. Whenever there is a failure of the media recording, another available media recorder can take over but signaling needs to reINVITE with the new address of the media recorder.
- If the T-lib client can detect a failure of a call recorder, the T-lib client can instruct recording to stop and then start recording again. This allows Media Server and resource manager to find another available recorder instance when establishing the recording session.

Failure of a T-lib client:

- Have a backup instance of the T-lib client to take over processing whenever the active instance fails. The backup instance establishes a separate T-lib connection to SIP Server and will only start to observe T-lib events from SIP Server when it becomes active.

Recording Duplication

To protect against any loss of recorded media in the case of a failure in the recording server function, the recorder has the option of duplicating the recording by establishing two independent recording sessions. The following steps outline the mechanism of duplication:

1. T-lib client requests for duplication or this can be a configuration parameter on SIP Server.
2. When SIP Server requests for recording, SIP Server will ask Media Server to create two separate recording sessions.
3. When Media Server initiates the Recording Session, Media Server must generate a different record identifier in the Request URI; this tells the Resource Manager that the duplicated of recording session is different than the first one so that Resource Manager may choose a different recording server based on the load balancing scheme.

Note: Media Server is still a single point of failure since RTP media is still pinned through a single instance of Media Server. This duplication only protects against the failure of SIP Server, failure of Resource Manager, and failure of recording server.

Interfaces

There are two interfaces relevant to the call recording architecture and are explained in the next subsections:

- T-Lib interface
- Media Server Recorder

T-Library

Enabling Call Recording

The T-lib interface allows recording to be enabled in three ways:

1. Through configuration—Set the `record` option to `true` in the DN object to instruct SIP Server to enable full-time recording for this DN. This is an existing feature.
2. Extension in the `TrouteCall` event to enable recording on trunk side or agent side. When calling `TrouteCall`, add the `record` key in the `extension` attribute and set the value to `source` for customer recording or `destination` for agent recording. This is an existing feature.
3. Adding a new extension in the `TrequestPrivateService` event to request call recording to be enabled on an existing connection as described in the following table:

RequestPrivateService—Request services that are supported only by certain T-Servers, and which are not covered by general feature requests.

Parameters	Description
AttrPrivateMsgID	This parameter is mandatory and must be equal to `GSIP_RECORD_START`.
AttrThisDN	This parameter is mandatory, and is the DN on behalf of which the operation is requested. It must be registered by the T-Client, but not necessary be a party on the call (for example, the supervisor may request recording of the agent's call).
AttrConnectionID	This parameter is mandatory and references the ID for the call to record.
AttrExtensions	Additional request parameters: • record (string)—Set to `source` to record from this DN referenced in this connection. Set to `destination` to record from the other DN referenced in this connection. This parameter is optional, and defaults to `source`. • Id (string)—Adds a recording identifier to the recording session. This identifier must be globally unique and is passed back in the recording session. This parameter is optional and if not present, Media Server constructs a unique identifier. • Dest (string)—Overrides the default SIP location of the recording server. THis parameter is optional. • Dest2 (string)—Overrides a second SIP location

Parameters	Description
	of the recording server for duplication of recording. This parameter is optional. • Params (string)—Additional parameters can be passed as generic name-value pairs. These parameters will show up in the recording session.
AttrReasons	The reasons. These are processed the same as for all other T-Library requests.

Note: SIP Server responds to the request with either `EventACK` to confirm the acceptance of the request, or `EventError` if the operation cannot be performed.

Runtime Control of Recording

When the recording session is established, T-lib interface allows run-time control of the recording for pause, resume, and stop. The following table describes a new extension for `TrequestPrivateService`:

RequestPrivateService—Request services that are supported only by certain T-Servers, and which are not covered by general feature requests.

Parameters	Description
AttrPrivateMsgID	Specifies the operation. Choose one of the following values: • `GSIP_RECORD_STOP`—Stop the recording. • `GSIP_RECORD_PAUSE`—Pause the recording. • `GSIP_RECORD_RESUME`—Resume the recording.
AttrThisDN	The DN on behalf of which the operation is requested. It must be registered by the T-Client, but not necessary be a party on the call.
AttrConnectionID	References the ID for the call being recorded.
AttrExtensions	Additional request parameters: • Params (string)—Additional parameters can be passed as generic name-value pairs that modifies the recording session.
AttrReasons	The reasons. These are processed the same as for all other T-Library requests.

Note: SIP Server responds to the request with either `EventACK` to confirm the acceptance of the request, or EventError if the operation cannot be performed.

Recording indication

There are two mechanisms for SIP Server to provide recording indication:

- After SIP Server successfully started recording on Media Server, SIP Server updates UserData to the call with GSIP_REC_FN with the file name of the recording. A T-lib client monitoring the call will receive an EventAttachedDataChanged with GSIP_REC_FN. This is an existing functionality for legacy Stream Manager recording. For clients who only want to know whether a recording has been enabled any time during the call, this userdata is sufficient.
- For clients such as Interaction Workspace who need to render the current recording state for the call, GSIP_REC_FN is not sufficient as a recording indicator. SIP Server provides a new UserData called GSIP_RECORD to provide the current state of recording for this call. Whenever SIP Server knows there is a change in recording, SIP Server sends an EventAttachedDataChanged with GSIP_RECORD to update the value of the key. This key has three values:
  - On—Recording is currently in progress.
  - Off—No recording in progress.
  - Paused—Recording is currently in progress but no media is currently captured.

Media Server Recorder

A recording session is basically a SIP session initiated by Media Server (routed via GVP Resource Manager) to the recorder. The following subsections describe the behavior of the recording session.

Format

Format of Request URI

A simple SIP INVITE request should be used for sending a request from Media Server to the recorder to enable recording from a replicated media stream. The request URI contains:

1. Call UUID
2. Unique identifier for the recording session, this can be generated by Media Server or supplied in the `<param name=”id”/>`
3. The DN of the party being recorded
4. The DN of the party that initiated the recording
5. Other parameters for the recording session

Depending on the recorder type, Media Server can be configured to send one SIP recording session per party in the call or a single SIP recording session containing both parties in the call.

For example, DN1 and DN2 are being recorded in the call and DN1 is the party that initiated the recording.

For recorders that accept one recording session per party, the following shows the requests URIs for each recording session:

```
sip:record=unique-identifier@recorder_address;calluuid=Call_UUID;dn=DN1;recordDN=DN1;other_parameters…
```

```
sip:record=unique-identifier@recorder_address;calluuid=Call_UUID;dn=DN2;recordDN=DN1;other_parameters…
```

The following shows the request URI when Media Server establishes a single recording session for both parties:

```
sip:record=unique-identifier@recorder_address;calluuid=Call_UUID;dn=DN1;otherDN=DN2;recordDN=DN…
```

Mapping

Mapping SDP Media Streams

One recording session per party

In the initial INVITE from Media Server, Media Server must provide the SDP offer or answer copied from party A or party B. Since the SDP answer provides the list of codecs being selected, Media Server should take the codec list and pass it as the m= line. If video media stream is also included by party A and B, by default Media Server must offer both media types (two `m=` lines) in the recording session unless the recording request specifies otherwise.

The recorder must generate an SDP answer in the `200 OK` response containing the IP and port in the answer. Media server must take this `IP:port` as the destination for the replicated media stream. The rest of the SDP content is irrelevant to Media Server.

The recorder may reject recording of video media stream by setting to zero port in the SDP answer. Media server must allow the recording session to continue but also indicate video recording state as off.

One recording session for both parties

If only audio media type is recorded, then the SDP offer will contain only two media `m=` lines. The first `m=` line must be the media of the party identified by `dn=<DN>`, and the second `m=` line must be the media of party identified by `otherDN=<DN2>`.

Mid-dialog

Mid-dialog Session Updates

There are a few scenarios where the recording session needs to be updated:

- Pause—Whenever pause is requested by `TrequestPrivateService`, Media Server will send a reINVITE request to the recorder and set the SDP with `a=inactive` for all media streams to pause the streams.
- Resume—Whenever resume is requested by `TrequestPrivateService`, Media Server will send a reINVITE request to the recorder and set the SDP with `a=sendonly` for all media streams to resume the streams.
- Mute—Whenever one of the party in the communication session updates the SDP to mute the line, Media Server will send a reINVITE to the recorder and set the SDP with `a=inactive` for all media streams to pause the streams.
- Call transfer/conference/consultation—Whenever there is a change to change/add/remove party in the call, Media Server will send a reINVITE request to the recorder to update the SDP. All of the session attributes and SIP dialog headers remain unchanged except the version in the o= line is incremented.
- Session timer—As per RFC4028, Media Server will send a reINVITE with no SDP changes.

Secured Recording

Secured Recording Session

When the recording session requires to be secured, there are a few scenarios to consider:

- If the communication session is secured with SIPS/SRTP, Media Server will receive the encrypted media stream that Media Server will proxy between the endpoints. However, Media Server shall not pass the same encrypted media stream to the recording session and provide the existing crypto keys to the recording server. This is apparently a violation of security for a middle man to escrow security keys to another party. The proper way to establish a secure recording session is by decrypting the communication session and re-encrypt the recording session using a new set of keys negotiated between recording client and recording server. This adds performance burden to Media Server to process decryption and encryption.
- If the communication session is not secured but the recording session is required to be secured, Media Server will need to encrypt the recorded media towards the recording server.
- If the communication session is secure by the recording session does not require security, then Media Server will need to decrypt the recorded media towards the recording server.

Failures

Failure Handling

The Recording Session may fail to establish for various reasons:

- Recorder responds negatively to the INIVTE request of the recording session
- Resource Manager fails to find an available recorder instance due to failures
- Resource Manager runs out of Recorder resources

As soon as Media Server recognizes the recording session has failed, Media Server notifies SIP Server that the recording is stopped. SIP Server updates the recording indication `GSIP_RECORD` to `off`, so the agent will be aware that the recording is not active.

Control for Third Party Screen Recording

The following tables show how the T-lib interface provides methods for passing a user event from the agent desktop to the recording vendor to trigger a screen capture request.

Agent Desktop to T-Server

RequestSendEvent—Send a specific event to the DN specified. This method is used by the agent desktop to deliver screen capture request to the third party recording vendor.

Parameters	Description
AttrThisDN	The DN on behalf of which the operation is requested. It must be registered by the T-Client, but not necessary be a party on the call.
Tevent	The event to deliver to recording server. • `thisDN`—Set to the same as the value for `AttrThisDN` • `messageType`—Set to `EventUserEvent` • `connectionID`—Set to the connection ID of the voice call if available • `userData`—Set the following user data key value pair: ◦ `screen-capture`—`start`, `stop`, `pause`, `resume`, `query` to note the type of screen capture operation. `query` does not make any changes to the current screen capture state, the server should provide a screen-capture-indication in response.

Note: SIP Server responds to the request with either `EventACK` to confirm the acceptance of the request, or `EventError` if the operation cannot be performed.

T-Server to Recording Vendor

EventUserEvent—The recording vendor who monitors for the agent DN will receive this event.

Parameters	Description
thisDN	The Agent's DN.
messageType	Set to `EventUserEvent`
connectionID	Set the connection ID of the voice call if available.
userData	Set the following user data kvp: • screen-capture—`start`, `stop`, `pause`, `resume`,

Parameters	Description
	query to note the type of screen capture operation.

Recording Indication for Voice Interactions

Recording Vendor to T-Server

RequestSendEvent—Send a specific event to the DN specified. This method is used by the agent desktop to deliver screen capture request to the third party recording vendor.

Parameters	Description
AttrThisDN	The DN that is to receive this event.
Tevent	The event to deliver to recording server. • thisDN—Set to the same as the value for AttrThisDN • messageType—Set to EventUserEvent • connectionID—Set to the connection ID of the voice call if available • userData—Set the following user data key value pair: ◦ screen-capture—enabled, diabled, paused to note whether the screen capture is currently enabled.

Note: SIP Server responds to the request with either EventACK to confirm the acceptance of the request, or EventError if the operation cannot be performed.

T-Server to Agent Desktop

EventUserEvent—The recording vendor who monitors for the agent DN will receive this event.

Parameters	Description
thisDN	The Agent's DN.
messageType	Set to EventUserEvent
connectionID	Set the connection ID of the voice call if available.
userData	Set the following user data key value pair:

Parameters	Description
	• screen-capture—`enabled`, `diabled`, `paused` to note whether screen capture is currently enabled.

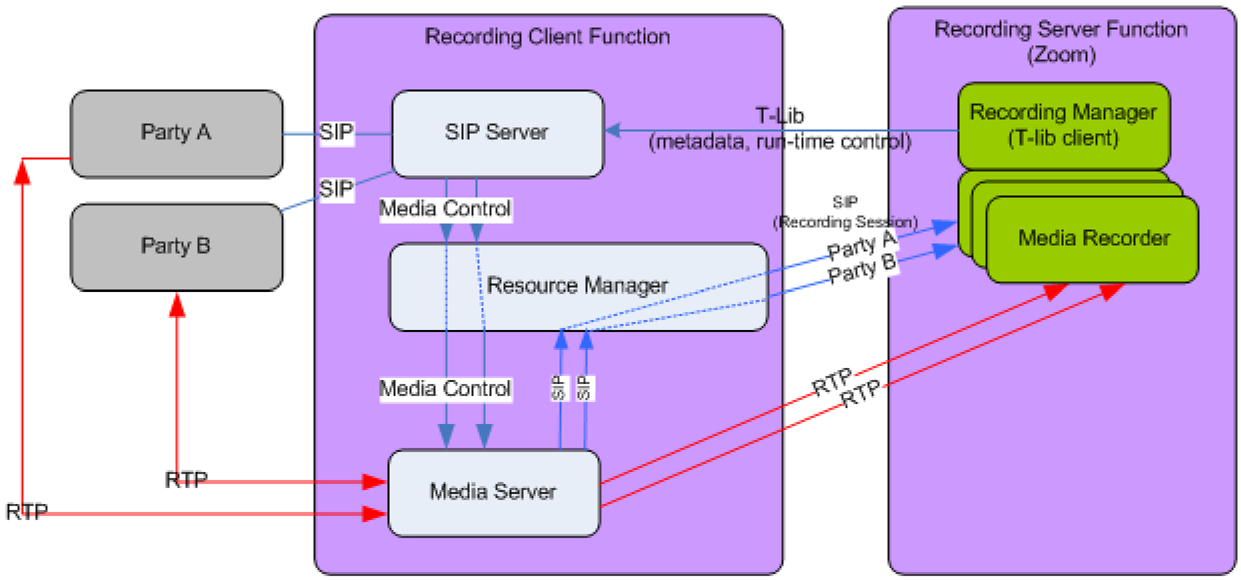
Integrating with Third Party Vendors

In this solution, third party vendors will be acting as the recording server function to provide storage of recorded media, as well as Quality Management. The Recording Client Function contains the following components:

- SIP Server
- Media Server
- Resource Manager

The Recording Manager (from third party vendors), is a logical component that acts as a T-lib client that receives call metadata from T-lib events. The recording manager can also initiate recording requests using T-lib to specific calls for dynamic recording, as well as using run-time controls for pause/resume.

When a recording request is initiated, SIP Server directs the media of Party A and B towards Media Server. SIP Server uses media control to instruct Media Server to replicate the media stream towards the media recorder.



Technical Requirements for Third Party Vendors

1. Dynamic recording—By default, `record=source` for all dynamic recording requests and normally this would be the extension (agent) DN. Third party vendors need to support a flag in the recording rule to change the parameter to `record=dest` so that the trunk DN and its geo-location is selected.
2. Full-time recording—using the same configuration as inbound calls, SIP Server would use the geo-location of the trunk DN for recording outbound calls, however, Media Server and SIP Server would

present the Recording Session to third party vendors with the trunk DN set as the Record DN. Third party vendors currently does not support Record DN that is not an extension DN. Third party vendors do not associates call events with the recording and hence the call recording cannot be found by third party vendors query interface. Third party vendors need to support this by handling the Record DN with the trunk DN set.

Geo-Location

Active Recording Ecosystem uses geo-location to provide a multi-site deployment with the capability to select specific pools of Media Servers and recording servers that are located at specific sites. The following sections describe Geo-location:

- Geo-Location Configuration
- Geo-Location Support

Geo-Location Configuration

Geo-location is configured in two places:

- DN objects in a switch
- Resource Groups for MCP and Recording Servers.

You can assign a geo-location tag for each DN (of type Trunk DN, Route Point DN, Extension DN, and Trunk Group DN). The `geo-location` parameter is configured in the `TServer` section of these places.

To assign a geo-location tag for a Resource Group (for MCP and Recording Server separately), use the Resource Group Wizard and set the geo-location as part of the Wizard process.

Usage

Geo-location is selected for each call depending on the usage model.

SIP Server selects the geo-location with the following order or preference for inbound calls:

1. Geo-location configured in the extensions of RequestRouteCall.
2. Geo-location configured in the Routing Point DN.
3. Geo-location configured in the inbound Trunk DN.
4. Geo-location configured in the DN where the recording is enabled.

For outbound calls, the following order of preference is used:

1. Geo-location configured in the extensions of RequestRouteCall.
2. Geo-location configured in the Routing Point DN.
3. Geo-location configured in the Agent DN.
4. Geo-location configured in the outbound Trunk DN if recording is enabled.

Full-time Recording

When a DN is configured to be recorded, the geo-location set at the DN. When more than one DN involved in the call has the geo-location set (for example, both the inbound Trunk DN and the Routing Point DN have the `geo-location` parameter set), then SIP Server selects the geo-location based on the order of preference listed above.

Selective Recording from a Routing Strategy

If `record=source` is set in the `RequestRoutecall` extensions, the geo-location of the inbound Trunk DN of the call is selected (if it is configured). If `record=destination` is set in the `RequestRoutecall` extensions, the geo-location of the agent (Extension DN) is selected.

Dynamic Recording

When dynamic recording is initiated by the T-lib `RequestPrivateService` function, either by a third party recording vendor, or by Interaction Workspace, the geo-location is selected based on the recorded DN in the call. Specifically:

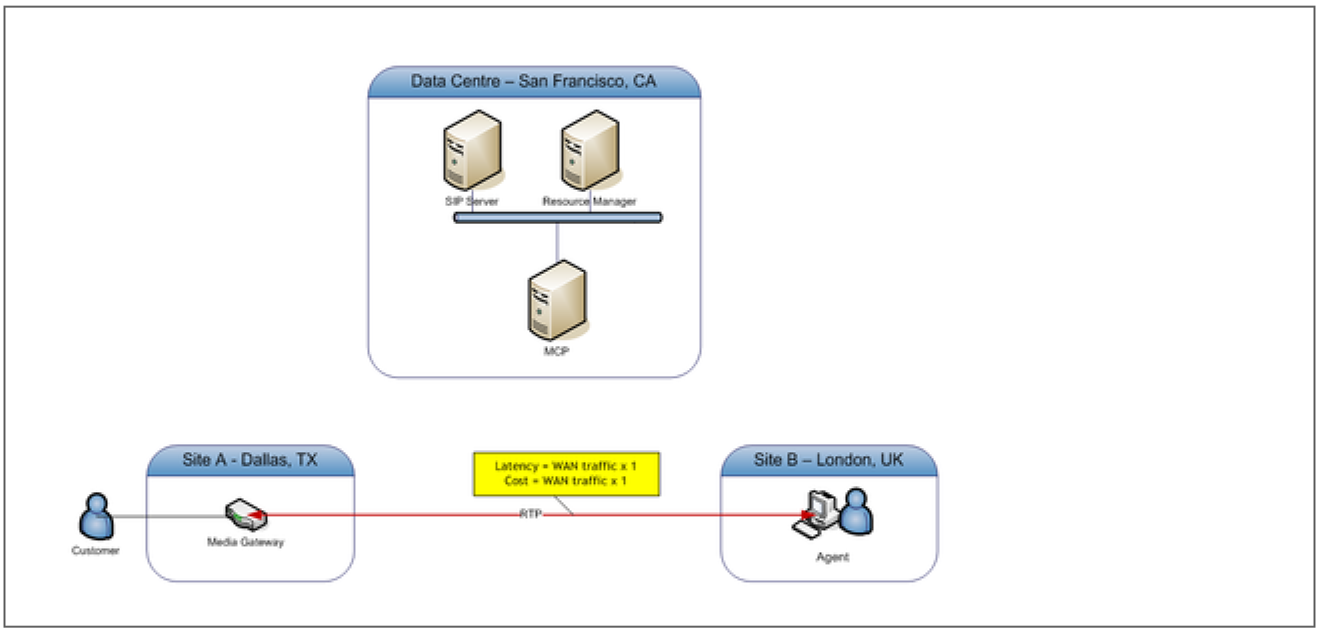
- If `RequestPrivateService` is requested with `AttrExtensions` as `record = source`, the geo-location configured for `thisDN` is selected. `record=source` is the default value if the extension is not defined.
- If `RequestPrivateService` is requested with `AttrExtensions` as `record = destination`, the geo-location configured for `otherDN` is selected.

Geo-Location Support

Geo-location support provides a multi-site deployment with the capability to select specific pools of Media Servers and recording servers that are located specific sites. The main motivation for selecting specific Media Servers is either to minimize WAN traffic or to minimize the latency introduced to a conversation when call recording is enabled.

No Recording

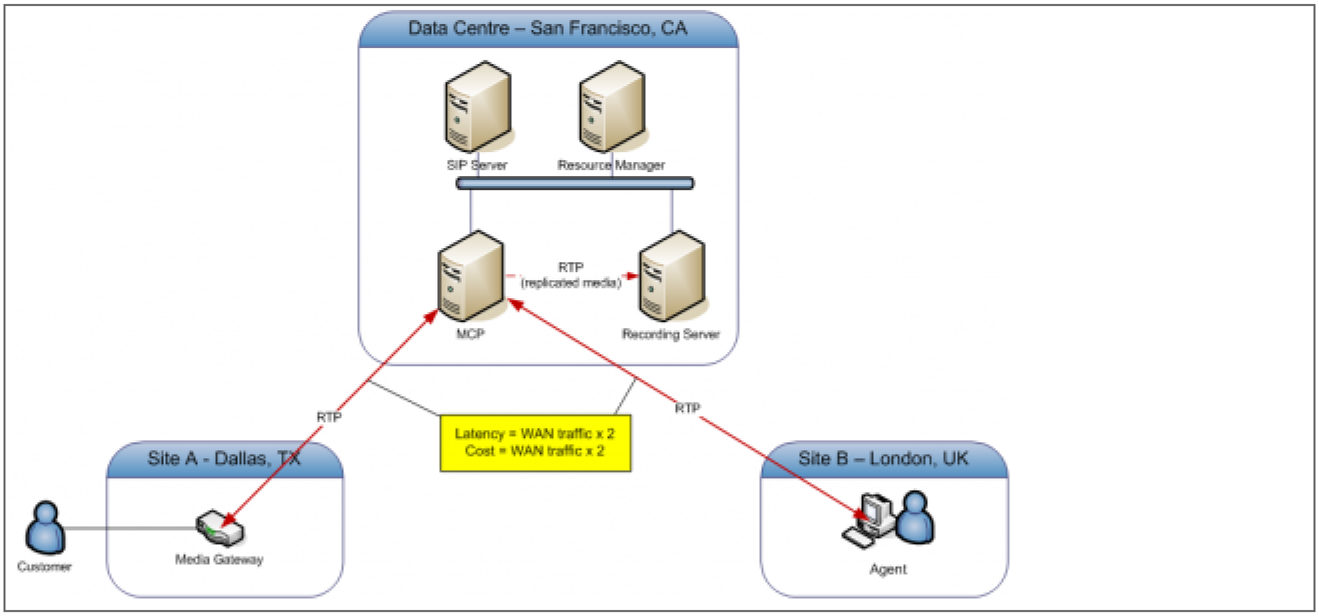
Customer and Agent Call Across the WAN with No Recording



In a typical scenario, the customer may be calling into a contact center site with a media gateway, and the agent is located in a different site as the agent.

Recording in Data Center

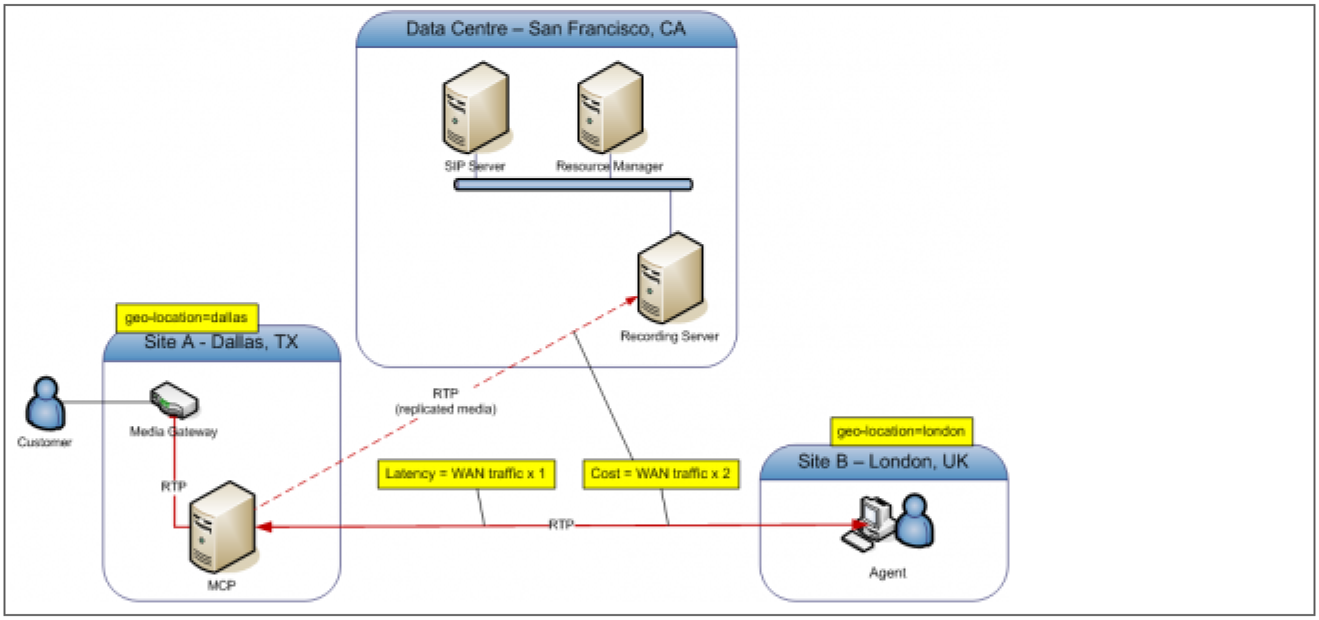
Customer and Agent Call Across the WAN with Recording in Data Center



When both Media Control Platform (MCP) and the recording server are deployed at the data center site, the deployment needs to double the WAN traffic since the media path needs to be bridged through the data center. This adds to the latency of the media path by doubling the WAN path.

Recording with Geo-location

Customer and Agent Call Across the WAN with Recording with Geo-location at the Customer Site



In order to minimize the latency, the geo-location feature has been introduced in SIP Server and Resource Manager. This feature allows MCPs to be deployed in a remote site that is close to one of the parties in the call. This diagram is a deployment that places MCP in Dallas as set in the `geo-location=dallas` parameter.

Geo-location for MCP and Recording Servers are considered separately by the Resource Manager. Geo-location is not deployed for the Recording Servers, and a single pool of Recording Servers is used at the data center.

Note: If you choose to deploy the Recording Servers according to the geo-locations, the same geo-location will be chosen for the same call for both MCP and Recording Server.

Audio Tones

In order to meet the regulatory requirements, some deployments require the system to periodically generate an audio tone to notify the participants in a call that the call is currently being recorded. The following sections describe how audio tones work:

- Applying Audio Tones
- Configuring Audio Tones
- Recording Conference Audio Tones

Applying Audio Tones

Audio tones can be generated either as all-party consent or one-party consent:

- All-party consent requires that all parties in the call being recorded to hear the audio tone periodically.
- One-party consent requires one of the parties in the call to hear the audio tone. The consent is configurable on Media Server.

There is a difference between all-consent/one-party-consent and applying the beep to certain calls:

- All-consent/one-party-consent setting is a global system setting on the MCP process.
- Applying a beep to certain calls can be done one of two ways:
    - Use a recording rule (for example, with Zoom) to add the beep src URI in the Tlib call to SIP Server to trigger the beep tone for a specific call.
    - Set up a separate tenant SIP Server that maps to a different GVP IVR Profile that controls the beep tone settings for that tenant.

When additional recording parameters are included when the MSML conference is instantiated, Media Server can apply the periodic audio tone. Additional states are kept in Media Server to keep track of the audio tone to be applied to the call.

Note: This is a custom Genesys extension to MSML conference to apply the audio tone.

The following parameters are configurable in the deployment and they follow the same conventions used for other parameters that can be applied to call recording. These parameters can be set either in the IVR Profile as service parameters, or as extensions in `RequestPrivateService`.

Parameter Name	Description
Audiosrc	The URI of the audio tone.If the URI resolves to a bad URI, then no audio tone is applied to the call, and recording error is notified to SIP server. If empty or not present then the recording will proceed without a tone. Only .wav files are supported. QTMF tones and files stored in sub-directories with multiple codecs that are supported by Media Server are not supported. For example, "music/beep" cannot be specified for this option, even though it is valid for other Media Server treatments.
Tonesilenceduration	The length of time, in milliseconds, between playing the audio tone. This is a mandatory parameter if the `audiosrc` parameter is defined, otherwise no audio tone is applied. The minimum accepted value is `1500` (if a smaller value is specified, `1500` will be used). Also, if the `tonesilenceduration` parameter is not present,

Parameter Name	Description
	MCP applies the default value of 30000 instead of not applying tone.

These parameters can be passed as additional parameters in RequestPrivateService (AttrExtensions). For example:

```
AttributeExtensions

      ‘rec ord’                        ‘source’

      ‘id’                                 ‘2134980asdf320990adsflkjag’

      ‘dest’                        ‘sip:10.0.0.101’

      ‘name’                        ‘value’

      ‘audiosrc’                        ‘http://example.com/tone.wav’

      ‘tonesilenceduration’        ‘30000’
```

The MSML code snippet would show as follows:

```
<msml>
   <createconference name=”recorder”>
     <gvp:recorder>
       <gvp:params>
         <gvp:param name=”record”>source</gvp:param>
         <gvp:param name=”id”>2134980asdf320990adsflkjag</gvp:param>
         <gvp:param name=”dest”>sip:10.0.0.101</gvp:param>
         <gvp:param name=”name”>value</gvp:param>
         <gvp:param name=”audiosrc”>http://example.com/tone.wav</gvp:param>
         <gvp:param name=”tonesilenceduration”>30000</gvp:param>
         <gvp:param name=”recordDN”>9000</gvp:param>
       </gvp:params>
     </gvp:recorder>
   </createconference>
   <join id1=”conf:recorder” id2=”conn:1234”/>
   <join id1=”conf:recorder” id2=”conn:2345”/>
</msml>
```

The same parameters can also be configured as service parameters in the IVR Profile. When the parameters are configured in the IVR Profile, they are treated as the default values for the IVR Profile, and the parameters can be overridden by AttrExtensions in the RequestPrivateService event on a per-call basis.

For example, the IVR Profile contains the gvp.service-parameters section with the following parameters:

- audiosrc=http://example.com/tone.wav
- tonesilenceduration=30000

Whenever Resource Manager passes the initial INVITE from SIP Server for instantiating the call recording, Resource Manager will insert the service parameters as additional request URI parameters:

```
INVITE
```

```
sip:msml=4e9d9e5700000003@mcp;dn=8403;record;audiosrc=http://example.com/
tone.wav;tonesilenceduration=30000
```

The parameters defined in the Request URI are considered as `<gvp:param>` in the MSML message for the conference. Media Server takes the parameters in Request URI first and then the parameters in `<gvp:param>` overrides the parameters in the Request URI. This means that the parameters defined in the `TrequestPrivateService` event has higher precedence than those defined in IVR Profile.

Other Media server considerations:

- When the recording is paused, no audio tone is generated.
- When the recording is resumed, the audio tone is applied.

Audio Tones Configuration

The following sections outline the general configuration for audio tones.

Media Server

The following table describes the options required for audio tones when using Media Server:

Section Name	Parameter Name	Description
Conference	record_recorddnhearstone	Specifies whether the RecordDN (Party A) hears the repeating tone.
Conference	record_otherdnhearstone	Specifies whether the OtherDN (Party B) hears the repeating tone.

Media Server allows you to configure whether the recording gets the audio tone as well. When the audio tone is injected into the call, Media Server distinguishes between what the participant hears and what the participant says. The above two configuration parameters affect what the participant hears.

Section Name	Parameter Name	Description
Conference	record_chan2source	Specifies the recorded media that represents the first participant (Record DN) in the recording session. • `recorddnsays` • `otherdnhears` If the Other DN is configured to receive consent and you want the consent to be recorded, set the value to `otherdnhear`.
Conference	record_otherdnhearstone	Specifies the recorded media that represents the second participant (Other DN) in the recording session. • `otherdnsays` • `recorddnhears`

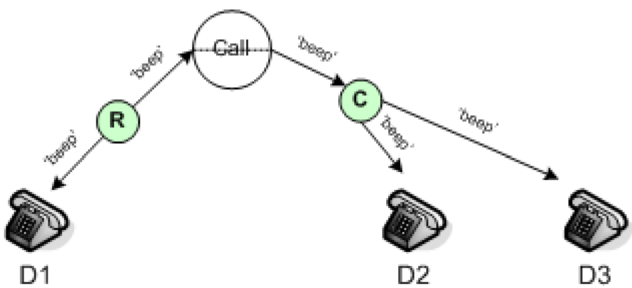
Section Name	Parameter Name	Description
		If the Record DN is configured to receive consent and you want the consent to be recorded, set the value to `recorddnhear`.

Recording Conference Audio Tones

When recording a conference, there are two Media Servers involved in the call:

- One for recording the recording DN.
- One for mixing media for other parties.

The audio tone is generated from the recording Media Server and are propagated to the conferencing Media Server. In order to ensure that all parties get the consent, set the `record_recorddnhearstone`, and the `record_otherdnhearstone` options in the `conference` section of the Media Server application to `true`.



When the recording is paused, no audio tone is generated. When the recording is resumed, the audio tone is applied.

The following parameters are configurable in the deployment and they follow the same convention used for other parameters that can be applied to a call recording. These parameters can be set in the following manner in order of precedence: - Extensions in RequestPrivateService; - IVR Profile for call recording service as service parameters

Parameter Name	Description
Audiosrc	The URI of the audio tone. If the URI resolves to a bad URI, then no audio tone is applied to the call, and recording error is notified to SIP server. If empty or not present then the recording will proceed without a tone
Tonesilenceduration	Length of time between playing the audio tone in milliseconds. Mandatory if audiosrc is defined, otherwise no audio tone is applied.

The above parameters can be passed as additional parameters in RequestPrivateService (AttrExtensions). For example:

```
AttributeExtensions
               ‘record’                        ‘source’
               ‘id’                                ‘2134980asdf320990adsflkjag’
               ‘dest’                        ‘sip:10.0.0.101’
```

```
‘name’                         ‘value’
‘audiosrc’                         ‘http://example.com/tone.wav’
‘tonesilenceduration’        ‘30000’
```

Genesys Active Recording System Setup

This section describes the steps required to configure the Genesys Active Recording Solution.

Prerequisites

- Install and Configure SIP Server
- Install and Configure Media Control Platform and Resource Manager. See the Genesys Voice Platform Deployment Guide for the steps required.
- Install and Configure the Utopy components. See the Genesys Interaction Analytics/SpeechMiner documentation for the steps required.
- Configure Media Server
- Enable MSR on Media Server

Configuring the Active Recording Ecosystem

1. Enable MSML services on SIP Server.

1. Configure the application level SIP Server. The following table describes the options to set:

Section Name	Parameter Name	Description
TServer	msml-support	Set to `true` to enable support of the call recording solution.
TServer	resource-management-by-rm	Set to `true` to enable support of the call recording solution. Resource monitoring and notification will be done by the Resource Manager. SIP Server will contact Media Server through Resource Manager.
TServer	msml-record-support	Set to `true` to enable SIP Server to engage GVP as a Media Server through the msml protocol for call recording.
TServer	record-consult-calls	Specifies whether to record consult calls:

Section Name	Parameter Name	Description
		• `true`—record consult calls • `false`—do not record consult calls
`TServer`	`Recording-filename`	Set to `$UUID$_$AGENTDN$_$ANI$_$DNIS$_$DATE$_$TIM`

2. Configure the monitored DN to enforce recording without a request from the third party vendor. f(see next table

- Set the `record` parameter to `true`.
  When set to `true`, call recording begins automatically when the call is established on the DN. Call recording stops when the DN leaves the call.

> **Important**
> This option changes behavior of the third party Vendor when EXTERNAL DATA recording rules are considered.

3. Configure the Trunk DN so that the third party recording vendor will not receive the number of the calling party in SIP signalling as a recorded DN.

- Set the `record` parameter to `false`.

2. Configure a DN for VoIP service.

1. Create a new MSML DN object and add the following parameters to the `General` tab:

- `Number` = The name of the MSML Server
- `Type` = `Voice over IP Service`

2. Add the following parameters to the Annex tab of the new DN:

Section Name	Parameter Name	Description
`TServer`	`Contact`	IP Address or FQDN? Set this to the Resource Manager IP address and port. Use the following format:

Section Name	Parameter Name	Description
		`sip: <Resource Manager_IP_address:Resource Manager_SIP_port>` Specifies the contact URI that SIP Server uses for communication with the treatment server.
TServer	service-type	Set to msml. VOIP service needs to be created with “service-type = msml” for MSML service.
TServer	prefix	Set to msml=.
TServer	subcription-id	Set to the name of the tenant to which this SIP Server belongs, using the following syntax <TenantName>
TServer	refer-enabled	Set to false.
TServer	make-call-rfc3725-flow	Set to 1.
TServer	ring-tone-on-make-call	Set to false. If multiple Resource Managers are configured, then create multiple VoIP DNs of the service-type=msml. In this case, SIP Server will balance the load between the multiple Resource Managers.
TServer	sip-hold-rfc3264	Set to true.
TServer	oos-check	Set to 10.
TServer	oos-force	Set to 2.

3. (Optional) Enable full-time recording.

To start recording based on static DN-level settings, set the record parameter to true in any of the following:

- Extension or ACD Position DN for agent-side recording
- Agent Login for agent-side recording
- Trunk DN for customer-side recording
- Trunk Group DN to record GVP interaction
- Voice Treatment Port DN to record GVP interaction

> **Important**
>
> Check with each recording vendor to verify that the above settings are supported. When Trunk DN recording is enabled, SIP Server does not generate all TLib events available for other DNs.

For more information about call recording, see Recording Using the T-Library Interface

4. (Optional) Enable selective call recording.

To enable selective recording, configure the following: In the routing strategy, configure the `TRouteCall` request to include the key record, with the values:

- `destination` for agent-side recording
- `source` for customer-side recording

You can also add the following optional key-value pairs in the extensions:

- `id`—A string used to add an identifier to the recording session. Must be globally unique. If not configured, Media Server constructs a unique identifier itself.
- `dest`—A string used to override the default location of the third party recording server.
- `params`—A string used to add additional parameters that can be passed as generic key-value pairs. These parameters will appear in the recording session.

> **Important**
>
> Full-time recording takes precedence over dynamic recording. SIP Server rejects any dynamic recording request that arrive while recording is already underway

.

For more information, see Dynamic Call Recording.

5. (Optional) Enable geo-location support.

1. In the `TServer` section of the Annex, set the `geo-location` parameter in any of the following objects:

- Inbound Trunk DN
- Routing Point DN
- Extension Attribute in `TRouteCall`
- Extension DN
- ACD Position DN

SIP Server selects and passes the `X-Genesys-geo-location` header using a different order of configuration precedence, depending on the call scenario. Refer to the *Genesys SIP Server Deployment Guide* for more details.
2. On the `Options` tab of the Resource Manager application, in the `rm` section, set the following option:

- `reject-recording-request-on-geo-location-nomatch`—When this parameter set to `true` (default value), Resource Manager rejects the recording request for non-matching geo-location while selecting LRGs for recording client MCP and recording server resources. If geo-location usage is not planned for recording, set this option to `false`.

3. (Optional) On the `Options` tab of all MCP's that are used for recording, in the `sip` section set following option:

- `mpc.copyheaders`—When this option contains the `X-Genesys-geo-location` value, the geo-location value is also taken into account when selecting third party recording server. If this option is set to an empty value, geo-location is ignored.

For more information about geo-location support, see Geo-location Configuration and Support.

6. Create a default IVR Profile.

Or, modify an existing IVR Profile.

> **Important**
> The default IVR profile is used by Resource Manager for all MSML requests; therefore, it may already be configured.

1. In Genesys Administrator, navigate to `Environment > Tenants`, select `Environment`, and go to the `Options` tab. Determine the default profile for the tenants (look under the section `gvp.general`, `gvp.general/default-application`. The value is set to the `Default` application).

Important

For multi-tenant configurations, configure the default profile in each tenant object.

2. Navigate to `Voice platform > IVR profiles > Default` application. On the `Options` tab, in the `gvp.service-parameters`, configure the following options:

- `recordingclient.recmediactl = fixed,2`. This value represents the number of invites. The number of invites varies for vendor:
  - Nice = 1
  - Zoom = 2
  - Verint = 2
- `Recordingclient.recdest = fixed,sip:[rm-ip]:[rm-port]`

7. Create a Recording Server application and provision a Resource Group.

1. Using Genesys Administrator, import `VP_CallRecordingServer_81x.apd` template file, and the corresponding `VP_CallRecordingServer_81x.xml` metadata file. These files are located on the Media Server installation CD, in the Resource Manager installation package.
2. Create one or more new Application object(s) using the template imported in step 1.

Important

The working directory and command line options can be set to any value, for example `.` (dot), as these are not used.

3. Add or modify the following options in the `gvp.rm` section:

- `aor=sip:<host|ip>:<port>`—Host and port are the FQDN or ip-address and listening SIP port of the recording server.
- `port-capacity`—Set this option according to recording server capacity.
- `redundancy-type = active`
- `recording-server = 1`

Important

Contact your third party recording vendor for details on setting the `port-capacity` option.

4. Using Genesys Administrator, create a new Resource Group for Recording Servers.

- When prompted in the Wizard, set the `Group Type` to `Recording` Server.
- When prompted, select valid values for the following options:
  - `Monitoring Method`—Set to `SIP OPTIONS`, or None.

Important

Consult with your recording vendor. If set to `SIP OPTIONS`, active recording will monitor the third party recording device. If set to None, monitoring is disabled.

- `Load Balancing Scheme`—Set to `parallel forking`, `round-robin` or `least used`.

Important

For NICE, select `parallel forking`.

- `Geo-location` (optional)
- Max Ports—Leave empty.
- When prompted, select the Recording Server Application that will be member of the Logical Resource Group (LRG), and configure:
  - `SIP Port`
  - `SIPS Port` (Optional)
  - `Max Ports`
  - Redundancy—Set to `active`.

Important

> Contact the third party recording vendor for the value of the SIPS Port, if the recording server supports Secure SIP.

8. Configure the MCP application and corresponding Resource Group.

1. Select the MCP Application that will be used as the recording client.

2. On the Options tab, in the vrmrecorder section, configure the following options:

- sip.routeset = sip:<[rm-ip or FQDN]:[rm-port];lr>, where [rm-ip or FQDN]:[rm-port] are the host and port of Resource Manager.
- sip.securerouteset = (optional)

3. If multiple MCPs are installed on same host, set the sip.transport.x option and ensure that port value used are not overlapping with each MCP.

> **Important**
> In environments with multiple MCP residing on same machine make sure that there are no port conflicts with settings in sip.transport.x options between each instances. If the MCP application is not used for recording, the recording module can be disabled by setting the vrmrecorder.enable option in the mpc section to false.

4. On the Options tab, in the mpc section, configure following options:

- rtp.multichantimeout = 0—For recording vendors that do not support sending RTCP packets for recording RTP sessions. If this value is set to a non-zero value, the recording session will be split into multiple segments.

9. (Optional) Configure audio tones during recording.

See Configuring and Applying AudioTones.

External Data Available from CIM

The data saved in the Call Recording external data table comes from various sources. The following information is available:

- Basic call-related data
- Call-related user data (attached data)
- Agent configuration data
- Extension Data
- Notification of recording
- Other Genesys Desktop Data (only for Genesys Driver)
- Other Call Recording Data (used internally by Call Recording)

The presence of specific data depends on the system configuration, routing design, network topology and on other conditions. Particular properties that must be stored in the Call Recording external data table must be configured during integration library implementation.

Configuring SIP Server for MSR

This topic describes the parameters to configure in SIP Server in order to enabled MSR recording.

Configuring the Application Level

See Enabling MSML Services on SIP Server.

Configuring the DN Level

1. Configure the Extension. Set the `record` parameter in the `Extension` object to `true` to enforce recording without a request from the third party vendor. This option changes the behavior of the Third Party Vendor when EXTERNAL DATA recording rules are considered.

2. Configuring the Trunk. Set the `record` parameter in the `Trunk` object to `false`. When set to `true`, the third party recording vendor will receive the number of the calling party in SIP signaling as a recorded DN.

3. Create a `TServer` section for msml in the VoIP Service DN, and add the following options:

- `service-type` = `msml`
- `sip-hold-rfc3264` = `true`

Configuring Media Server for MSR

This section outlines how to configure Media Server to enable MSR recording.

1. Creating a Resource Access point

Each recorder for the recording server is assigned as a resource access point. The `.apd` file, when provided with the IP Address of the resource manager, helps to create this and populates the parameters for the resource access point. The host part configuration is not as important as configuring the host for the recording server itself, because the host part configuration is a Resource Access Point (external), Management Framework is not going to ping or check its status. Configure the Resource Access Point in Configuration Manager or Genesys Administrator.
After creating the object, verify the following:

- That the `gvp.rm` section exists.
- That the `aor` parameter points to the recording server address.
- Whether the provision section has the parameter `recording-server=1` (this should be by default).

2. Configuring Media Control Platform Options

Configure the following Media Control Platform (MCP) parameters:
1. Navigate to `vrmrecorder > sip.routset`. Set `sip.routset` to `<sip:<[rm-ip or FQDN]:[rm-port];lr>`. This defines the route that MCP uses to access recording server. Set to the Resource Manager to allow the Resource Manager to invite the SLR servers from Call Recording. The syntax is very important, the expression must have < and > (without the < and > MCP would not invite the Resource Manager and recordings will fail).

2. Create the `recordingserver` resource group using the resource group wizard in Genesys Administrator. Select the service type for a recording server. GA finds the Resource Access Points with `provision.record-server=1`, and displays a list to choose from.

3. For the selected recorder resource, set the `port-capacity` and set the `redundancy-type` to `Active`.

4. Also, as for any resource group, select which Resource Manager should manage this group.

3. Configuring the Media Control Platforms

When assigning the Media Control platforms (MCPs) for handling call recording, the IP address and Port must match the details of the MCP. Set the `max ports` option to double the number of calls that you want to handle with the MCP. One port is used per stream in the call, one for the customer leg and one for the caller leg. If `max ports` is set to `1000`, the MCP can handle 500 calls.

4. Configuring the Recording Servers

Add recording servers to the resource group.

5. Configuring the Database Access Points

The access points are simply representations and do not include any configuration data for the SLR servers. The port is defined but is not used.

6. Configuring the Recording Server Group

1.In Genesys Administrator, navigate to `PROVISIONING > Voice Platform > Resource Groups`, and click `New`. This will start the Resource Group Wizard.

2. In Resource Manager Selection, select the Resource Manager.

3. In `Group Name` and Type, enter the group name and select `Recording Server` in Group Type.

4. Use the defaults for Tenant Assignment and Group Properties.

5. Add `recordingclient.recmediactl=fixed,2` to the service parameters.

7. Assigning Resources

1. Set the IP Address and SIP Port of the Recording Servers to those of the installed third party recording component.

2. Set the Max Ports option to double the number of calls you want to be able to record.

3. When configuring recording servers, all Recording servers must have the Redundancy set to Active. If not they will not be seen as an available resource to Resource Manager and calls will not be recorded.

4. In the `gvp.service-parameters` section, add the `recordingclient.recdest=fixed,sip:[rm-ip]:[rm-port]` option.

8. Configuring the IVR Profile

1. In Genesys Administrator, navigate to `Environment > Tenants`, select `Environment`, and go to the `Options` tab. Check what the default profile for tenants is (look under the section `gvp.general`, `gvp.general/default-application` .The value is set to the Default application).

2. Nagivate to `Voice platform > IVR profiles > Default` application. On the `Options` tab, under the `gvp.service-parameters` section, configure the following options:

- `recordingclient.recmediactl` = `fixed, 2`
- `recordingclient.recdest` = `fixed,sip:[rm-ip]:[rm-port]`

In case of RM active-active setup, for enabling duplicate recording, you may specify recdest2 parameter which will be configured similar to recdest parameter in the following way:

```
recordingclient.recdest = fixed,sip:[rm-ip-ha-1]:[rm-port]
recordingclient.recdest2 = fixed,sip:[rm-ip-ha-2]:[rm-port]
```

For example,

```
[gvp.service-parameters]
recordingclient.recdest=fixed,sip:172.27.166.76:5060
recordingclient.recdest2=fixed,sip:172.27.206.41:5060
```

9. Configuring a Recording Service

1.Create a new DN of type `Voice over IP`. Enter the recorder server name for the `Number`.

2. In the `Annex` section, create a `TServer` section, and configure the following options:

- `contact`—Set to `SIP URI`. This option specifies the contact URI that SIP Server uses for communication with the recorder server.
- `request-uri`—Set to `SIP URI`. This specifies the value of the Request-URI address to be used in the INVITE message, if that address is different from the address where the message will be sent.
- `service-type`—Set to `recorder`.

> **Important**
>
> SIP Server can also record a file name when emergency recording is initiated by an agent. See the `emergency-recording-filename` configuration option in the SIP Server Deployment Guide for more information.

10. Configuring a Treatment Service

1.Create a new DN of type `Voice over IP`. Enter the treatment server name for the `Number`.

2. In the `Annex` section, create a `TServer` section, and configure the following options:

- `contact`—Set to `SIP URI`. This option specifies the contact URI that SIP Server uses for communication with the treatment server.
- `service-type`—Set to `treatment`.

11. Configuring an MSML Service

1.Create a new DN of type `Voice over IP`. Enter the MSML server name for the `Number`.

2. In the `Annex` section, create a `TServer` section, and configure the following options:

- `contact`—Set this to the Resource Manager IP address and port. Use the following format: `sip:<Resource Manager_IP_address:Resource Manager_SIP_port>`. This option specifies the contact URI that SIP Server uses for communication with the MSML server.
- `msml`—Set to `msml`.
- `prefix`—Set to `msml`.
- `subscription-id`—Set to <TenantName> where <TenantName> is the name of the tenant to which this SIP Server belongs.
- `refer-enabled`—Set to `false`.
- `make-call-rfc3725-flow`—Set to 1.
- `ring-tone-on-make-call`—Set to `false`. If multiple Resource Managers are configured, then create multiple VoIP DNs of the `service-type=msml`. In this case, SIP Server will balance the load between the multiple Resource Managers.

Integrating Media Server for MSML

The following describes the basic steps required to integrate Genesys Media Server for MSML.

1. Configure SIP Server for MSML

See Configuring SIP Server for MSR.

2. Configure the MSML DN

Create a Voice over IP DN. See Configuring an MSML Service.

3. Configure the GVP components

Configure the following GVP components to their default settings:

- Resource Manager
- Media Control Platform

> **Important**
> SIP Server and Resource Manager use the same port `5060`. If both are deployed on the same host, you may have to change port numbers to avoid conflicts. Genesys suggests shifting the port numbers in the Resource Manager options up by 100—from `5060-5067` to `5160-5167`. For more information, see the *Genesys Media Server 8.1 Deployment Guide*.

4. Configure an MCP resource group for MSML services

1. Create a resource group for the MCP instances that will be used to provide MSML service.

2. Configure the resource group with the following minimum mandatory options:

- `load-balance-scheme`—Set to `round-robin`.

- `monitor-method`—Set to `option`.
- `port-usage-type`—Set to `in-and-out`.
- `resource-confmaxsize`—Set to `-1`.
- `service-types`—Ensure that `msml` is included in the list of servicetypes.

For more information, see the *Genesys Media Server 8.1 Deployment Guide*.

5. Ensure MCP is connected to Resource Manager

Ensure that the MCP instances are included on the `Connections` tab of the Resource Manager Application object.

6. Configure the Resource Manager application

1. In the `rm` section, configure the following parameters:

- `rm.conference-sip-error-respcode`—Set to `503`.
- `rm.resource-available-respcode`—Set to `603`.

2. In the `monitor` section, configure the following parameter:

- `sip.proxy.realeaseconfonfailure`—Set to `false`.

7. Create a default IVR Profile

Create a new IVR Profile to be used as the default for your particular tenant.

1. In the `Voice Platform Profiles` folder, create a new GVP IVRProfile.

2. On the Annex tab of the IVR Profile, create a `gvp.general` section, adding the following option:

- `service-type`—Set to `voicexml`.

8. Configure the Tenant object

Assign the IVR Profile as the default for your tenant.

1. Right-click your Tenant object, then select `Properties`.

2. On the Annex tab of the `Properties` window, create a `gvp.general` section.

3. Add the following options:

- `default-application`—Set to the name of the default IVR Profile.
- `service-type`—Set to `voicexml`.

Genesys Media Server

The Genesys Media Server is a module that provides MSML-based media services offered by the Genesys Voice Platform. When integrated with SIP Server, it supports MSML-based call recording, where the Genesys Media Server acts as a proxy, replicating the media stream in a new recording session with a third-party voice recorder that does the actual recording.

> **Important**
> Local file recording is not supported by Genesys Media Server for MSML-based call recording.

Feature Configuration The following provides an overview of the main steps that are required to configure the call-recording functionality.

1. Integrate SIP Server with Genesys Media Server

See Integrating Media Server for MSML.

2. Enable full-time call recording

To start recording based on static DN-level settings, set `record` to `true` in any of the following:

- Extension DN for agent-side recording.
- Agent Login for agent-side recording.
- Trunk DN for customer-side recording.

3. Enable dynamic call recording

To start recording during an ongoing conversation, configure either of the following:

- In the routing strategy, configure the `TRouteCall` request to include the key record, with the values:
  - `destination` for agent-side recording
  - `source` for customer-side recording
- In the T-library client, configure the `TRequestPrivateService` request to include the key record, with the values:

    - `source` for recording `ThisDN`
    - `destination` for recording `OtherDN`
- You can also add the following optional key value pairs:
    - `id` — A string used to add an identifier to the recording session. Must be globally unique. If not configured, Media Server constructs a unique identifier itself.
    - `dest` — A string used to override the default location of the third party recording server.
    - params — A string used to add additional parameters that can be passed as generic key value pairs. These parameters will appear in the recording session.

> **Important**
> Full-time recording takes precedence over dynamic recording. SIP Server rejects any dynamic recording request that arrive.

4. Enable mid-call recording control

To control the recording during an established session, configure `TRequestPrivateService` to include the key `AttrPrivateMsgID`, using one of the following values:

- `GSIP_RECORD_STOP (3014)`
- GSIP_RECORD_PAUSE (3015)
- GSIP_RECORD_RESUME (3016)

Troubleshooting

Calls are not being recorded

- If calls are not being recorded, start troubleshooting from SIP Server by checking if the following attached data keys are present:
    - `GSIP_RECORD`
    - `RECORDING_STATUS_GIM`
- If neither key is present, one of the following is likely:
    - Nothing is configured. Follow the instructions above to check if an attached data is present, and then enable recording on the Third Party Vendor side.
    - Only the Genesys side is configured. Enable recording on the Third party vendor side.
- If only the `RECORDING_STATUS_GIM` key is present with the value `RECORDING_NO`:
    - Check the SIP Server log to see if SIP Server received a `PrivateRequest` request from the Third Party Vendor.
        - If the request was received, but SIP Server returned an error code, recording is not configured on Genesys side.
        - If the request was not received, the recording rule did match a call.
    - If the `PrivateRequest` was received and handled correctly (SIP Server returned success response), there is mismatch between RTP and TLib events.
        - Check that the `record` option is set to `true` on the trunk, or if the `recording-filename` option has non empty value.
- If only the `GSIP_RECORD` key is present:
    - The Third Party Vendor has disconnected from SIP Server. Restart the core process and determine why there was a disconnection.
    - The Third Party Vendor is not configured, but the `record` option is set on the DN level. Enable recording on the Third Party Vendor side.
- If the `GSIP_RECORD` option is set to `OFF`, and the `RECORDING_STATUS_GIM` option is set to `RECORDING_NO`, there was an error during recording.
- Follow SIP logs in SIP Server, MCP and in Resource Manager and check for the following:
    - Whether both DNs were properly connected to the MCP. Check if both reINVITEs sent to the phone and the gateway are okay.
    - Whether the MCP is sending the correct INVITE record=<value> to the correct Resource Manager after parties are connected to MCP.
    - Whether the Resource Manager forwards the INVITE record to the external recording system.

- Whether the external system received an INVITE record= and whether it responded correctly to this INVITE.

- If the `GSIP_RECORD` option is set to `ON`, and the `RECORDING_STATUS_GIM` option is set to `RECORDING_NO`, there is a mis-configuration in the default profile. Check the section name, the option names, and the values.

SIP Server failure

If the `RequestPrivateService` message is submitted to start recording with any value containing an integer value instead of string value, SIP Server will fail.