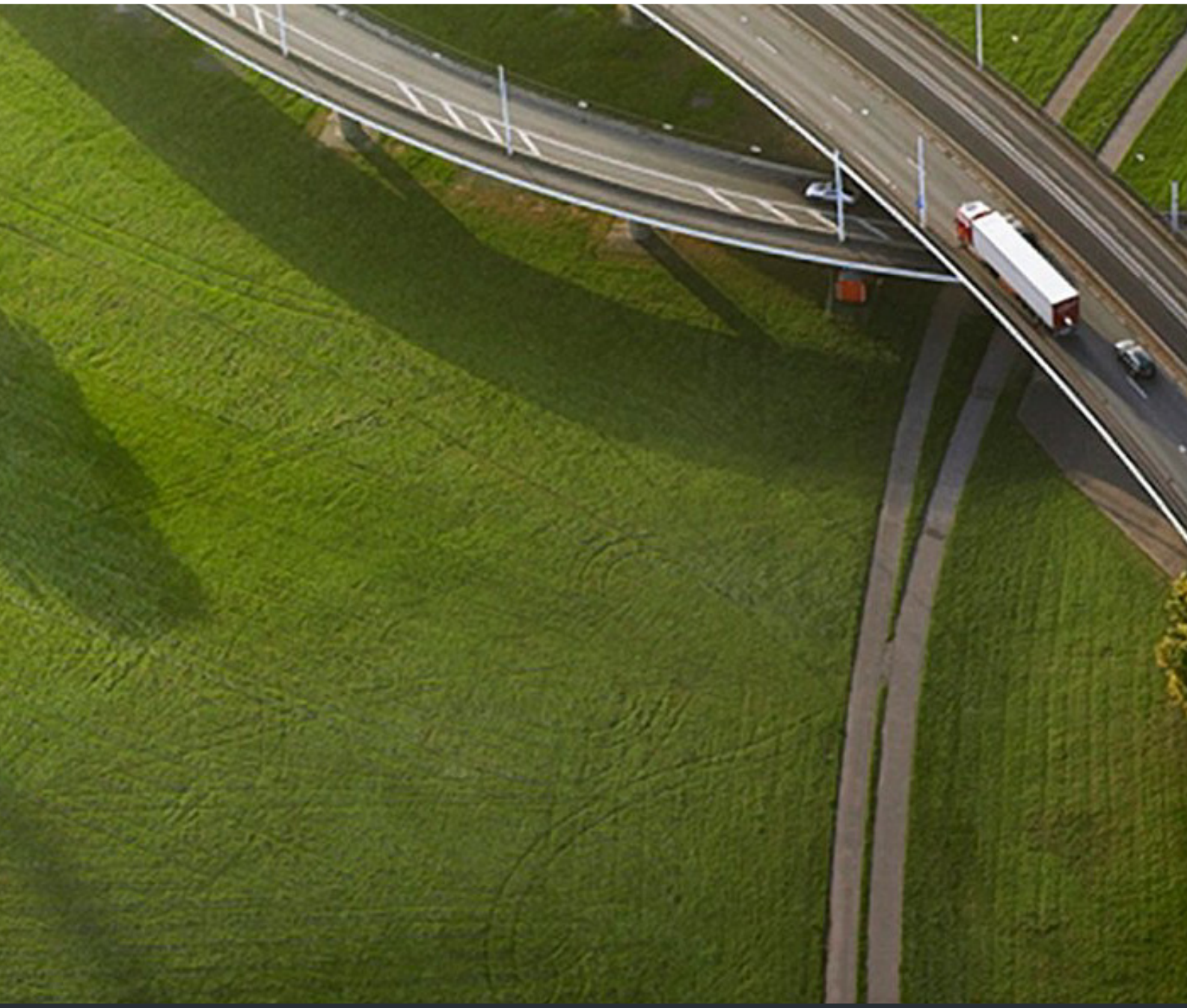
UNIFY Harmonize your enterprise



OpenScape Voice V6
Interface Manual: Volume 5, SIP Interface to Phones

Description

A31003-H8060-T101-05-7618

Our Quality and Environmental Management Systems are implemented according to the requirements of the ISO9001 and ISO14001 standards and are certified by an external certification company.

Reference No.: A31003-H8060-T101-05-7618

UNIFY Harmonize your enterprise

unify.com

Contents

History of Changes

Issue	Date	Changes
1	04/2011	V6 Issue 1 Following enhancements: • Support SDP up to 10 Kbytes • SDP Backward Compatibility for Best Effort SRTP for SDES • Best-effort SRTP with MIKEY enhancements • Updated description in Section 5.7, “Session Timers”
2	07/2011	V6 Issue 2: • Updated Section 6.7, “Key Management for Media Security”
3	11/2011	V6 Issue 3: • Updated Appendix J, “Quality of Service (QoS) Monitoring”
4	12/2011	V6 Issue 4: • Enhancement made to OpenScape Voice for Endpoint Controlled Recording when a SIP INVITE with X-Siemens-Call-Type header is received with value ‘recording’. OpenScape Voice forwards the SIP INVITE to the Recording Server with caller ID of the recorded parties. (Section 5.14.1, “X-Siemens-Call-Type” and Appendix K.3.2, “Actions at the SIP Server (B2BUA)”) • Correction updates to Section 8.16.1.2, “Interface to a non-recording client” and Section 8.16.1.3, “Server controlled recording” • Some client devices support providing their MAC addresses during registration. This information helps in identifying and assists locating the device on the network, e.g. for the purpose of emergency calls. Refer to Section 5.11.50, “X-Siemens-IID” and Section 5.14.7, “X-Siemens-IID”.

List of Figures

1 General Information

1.1 Warning and Disclaimer

Every effort has been made to make this document as complete and as accurate as possible, but no guarantee of 100% accuracy is implied. Siemens shall have neither liability nor responsibility to any person or entity with respect to the correctness of the information contained herein, other than to correct mistakes that are subsequently discovered in the text. Incorrect text shall not be construed as a promise or commitment on the part of Siemens to modify its products to achieve described operation, although Siemens is willing to work with its customers to provide requested functionality.

1.2 References

1.2.1 Normative References

Reference	Description
RFC 2119	"Key words for use in RFCs to Indicate Requirement Levels", BCP 14, Bradner, S
RFC 2245	"The TLS Protocol Version 1.0", T. Dierks, C. Allen
RFC 2246	"The TLS Protocol Version 1.0", T. Dierks, C. Allen
RFC 2543	"SIP: Session Initiation Protocol"; M. Handley / H. Schulzrinne / E. Schooler/ J. Rosenberg, March 1999 (Obsoleted by RFC3261, RFC3262, RFC3263, RFC3264, RFC3265)
RFC 2617	"HTTP authentication: Basic and Digest Access Authentication", Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, June 1999
RFC 3261	"SIP: Session Initiation Protocol"; J. Rosenberg / H. Schulzrinne / G. Camarillo / A. Johnston / J. Peterson / R. Sparks / M. Handley / E. Schooler
RFC 3262	"Reliability of Provisional Responses in the Session Initiation Protocol (SIP)"; J. Rosenberg / H. Schulzrinne
RFC 3263	"SIP: Locating SIP Servers"; J. Rosenberg / H. Schulzrinne
RFC 3264	“An Offer/Answer Model with SDP"; J. Rosenberg / H. Schulzrinne
RFC 3265	"Session Initiation Protocol (SIP)-Specific Event Notification", A. B. Roach
RFC 3311	“The Session Initiation Protocol (SIP) UPDATE Method"; J. Rosenberg

Reference	Description
RFC 3323	"A Privacy Mechanism for the Session Initiation Protocol (SIP", J. Peterson
RFC 3325	‘Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks"; C. Jennings / J. Peterson / M. Watson
RFC 3326	“The Reason Header Field for the Session Initiation Protocol (SIP)", H. Schulzrinne, D. Oran, G. Camarillo
RFC 3328	“Session Initiation Protocol (SIP) Extension for Instant Messaging", B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, D. Gurle
RFC 3515	“The Session Initiation Protocol (SIP) Refer Method", R. Sparks
RFC 3550	RTP: A Transport Protocol for Real-Time Applications. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. July 2003.
RFC 3551	RTP Profile for Audio and Video Conferences with Minimal Control. H. Schulzrinne, S. Casner. July 2003.
RFC 3725	Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP). J. Rosenberg, J. Peterson, H. Schulzrinne, G. Camarillo. April 2004.
RFC 3830	“MIKEY: Multimedia Internet KEYing", J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman
RFC 3840	"Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", J. Rosenberg, H. Schulzrinne, P. Kyzivat
RFC 3841	"Caller Preferences for the Session Initiation Protocol (SIP)", J. Rosenberg, H. Schulzrinne, P. Kyzivat
RFC 3842	“ A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", R. Mahy
RFC 3891	"The Session Initiation Protocol (SIP) "Replaces" Header", R. Mahy, B. Biggs, R. Dean
RFC 3892	“The Session Initiation Protocol (SIP) Referred-By Mechanism", R. Sparks
RFC 3903	"Session Initiation Protocol (SIP) Extension for Event State Publication", A. Niemi
RFC 3966	"The tel URI for Telephone Numbers", H. Schulzrinne
RFC 4028	“Session Timers in the Session Initiation Protocol (SIP ", S. Donovan, J. Rosenberg
RFC 4235	“An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", J. Rosenberg, H. Schulzrinne, R. Mahy.
RFC 4566	“Handley, et al., "SDP: Session Description Protocol", July 2006. Obsoletes RFC 2327
RFC 4567	"Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP)" , J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman
RFC 4568	“Session Description Protocol (SDP) Security Descriptions for Media Streams”, F. Andreasen, M. Baugher, D. Wing; July 2006

Reference	Description
RFC 4575	"A Session Initiation Protocol (SIP) Event Package for Conference State", J. Rosenberg, H. Schulzrinne, O. Levin.
RFC 4961	“Symmetric RTP / RTP Control Protocol (RTCP)”, D. Wing
RFC 5630	"The tel URI for Telephone Numbers", H. Schulzrinne, December 2004 "The use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", F. Audet

1.2.2 Informative References

Reference Number	Description
ICE	draft-ietf-mmusic-ice-08 - " Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", J. Rosenberg (work in progress)
SIP DIVERSION	“Diversion Indication in SIP”; S. Levy / J.R. Yang; RFC 5806, Historic
SIP Location conveyance	draft-ietf-sip-location-conveyance - "Location Conveyance for the Session Initiation Protocol". J. Polk, B. Rosen (work in progress)
STUN	draft-ietf-behave-rfc3489bis-03 - "Simple Traversal of UDP Through Network Address Translators (NAT) (STUN)", J. Rosenberg, C. Huitema, R. Mahy, D, Wing (work in progress)
TURN	draft-ietf-behave-turn-00 - "Obtaining Relay Addresses from Simple Traversal of UDP Through NAT (STUN)", J. Rosenberg, R. Mahy, C. Huitema (work in progress

1.3 Terminology

In this document, if the words “must”, “must not”, “should”, “should not”, and “may” are used then they are not intended to be “normative” (as described in RFC 2119) but rather they are used to indicate a capability or desired behavior. In addition, when the word “may” is used it generally implies that the capability is controlled via configuration or provisioning options.

1.4 Keyword / Descriptor

- SIP Interfaces
- SIP Client
- OpenScape Voice

2 Purpose

2.1 Scope

The goal of this document is to provide a specification for the SIP client interface, which is the interface by which clients (e.g., wired phones, wireless phones, soft clients, etc.) communicate using SIP with a Siemens OpenScape Voice SIP server.

This specification defines the use of existing protocols specified in IETF RFCs, and other documents. It does not define SIP protocol enhancements.

Clients conformant to this specification will be able to interoperate satisfactorily (from a signalling perspective) with a Siemens OpenScape Voice server.

The specification of the use of the Session Description Protocol (SDP) within SIP messages by SIP clients and SIP servers is outside the scope of this version of this specification.

The SIP interface for peer-to-peer (P2P) SIP is outside the scope of this version of this specification.

Section 4 identifies the SIP OpenScape Voice interface components and presents typical system scenarios that can be found in many customer projects.

Sections 5, 6 and 7 define the interface by introducing signalling building blocks, which are based on the SIP protocol and SIP protocol enhancements (see RFC 3261 and others) as well as other existing specifications.

Sections 8 and 9 describe telephony features using those signaling building blocks from 5, 6 and 7.

3 Conformance

In order to interoperate correctly, the OpenScape Voice and the SIP Client must conform to the requirements in this document.

3.1 Interoperability Testing

Adherence to the normative statements that are presented in this document does NOT guarantee full interoperability of OpenScape Voice with the various SIP Clients.

In many cases, the requirements from the SIP Clients are different from each other because there is/was no standardized interoperability specification available. There are several specifications from different stakeholders in the VoIP business available that vary in format and contents. Often, no description how telephony features are executed is provided. Therefore, in addition to this document, which specifies the default behavior, separate SIP Client specific annex documents may be provided to detail specific requirements that are different from this default behavior.

For these reasons, it is absolutely necessary to perform interoperability testing with each SIP Client and the OpenScape Voice in order to guarantee successful interoperation between OpenScape Voice and SIP Clients.

4 Architecture and SIP Deployment Scenarios

This chapter identifies the architectural landscape components and contains descriptions of possible OpenScape Voice deployment scenarios. These are limited to those scenarios which are considered as basic connection scenarios of SIP Clients to OpenScape Voice systems.

4.1 SIP Architectural Landscape Components

Figure 4.1 identifies the SIP components within the OpenScape Voice V5 landscape. This document describes the interfaces between the endpoints listed below and OpenScape Voice.

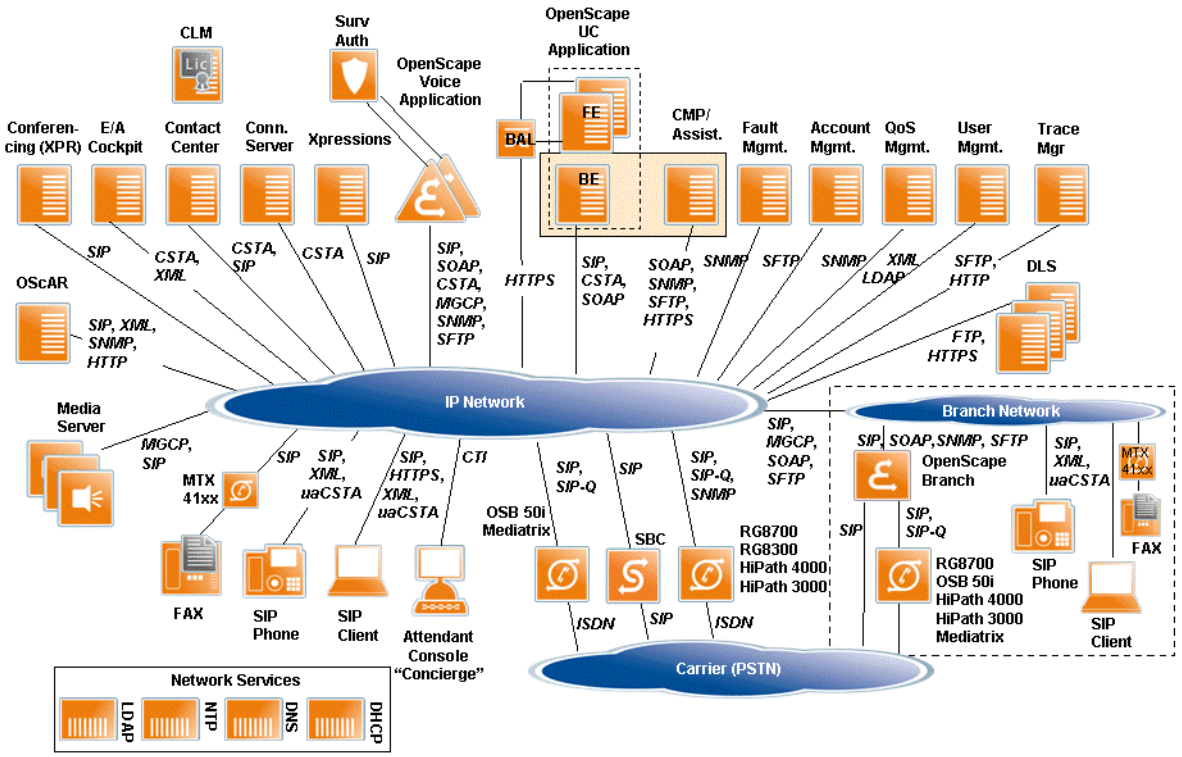


Figure 4.1 OpenScape Voice Solution Landscape

4.2 OpenScape Voice SIP Deployment Scenarios

OpenScape Voice is primarily deployed with a direct client interface. However, the following deployment options are supported:

- Clients interfacing directly to OpenScape Voice
- Edge proxy between client and OpenScape Voice
- Client with mobile appliance
- Client with mobile GSM Appliance

Also in some configurations there may be multiple intervening SIP-aware entities between the client and OpenScape Voice, e.g., edge proxy and SBC.

Throughout the remainder of this document the SIP client interface is specified as operating between a client and OpenScape Voice, but is equally applicable (with certain exceptions) between a client and an edge proxy (or mobile appliance when in WLAN mode) or between an edge proxy (or mobile appliance when in LAN mode) and OpenScape Voice.

4.2.1 SIP Client Functions

A client acts as a SIP User Agent (UA) and exchanges SIP messages with other SIP entities including:

- Other SIP clients
- Other types of SIP UA (e.g., gateways, media servers)
- Proxies, registrars, and redirect servers
- Back-to-back user agents (B2BUAs)
- Other types of SIP-aware entities such as session border controllers (SBCs) and application layer gateways (ALGs) that exhibit properties of proxies or B2BUAs.

Figure 4.2 shows an example configuration involving some of these entities.

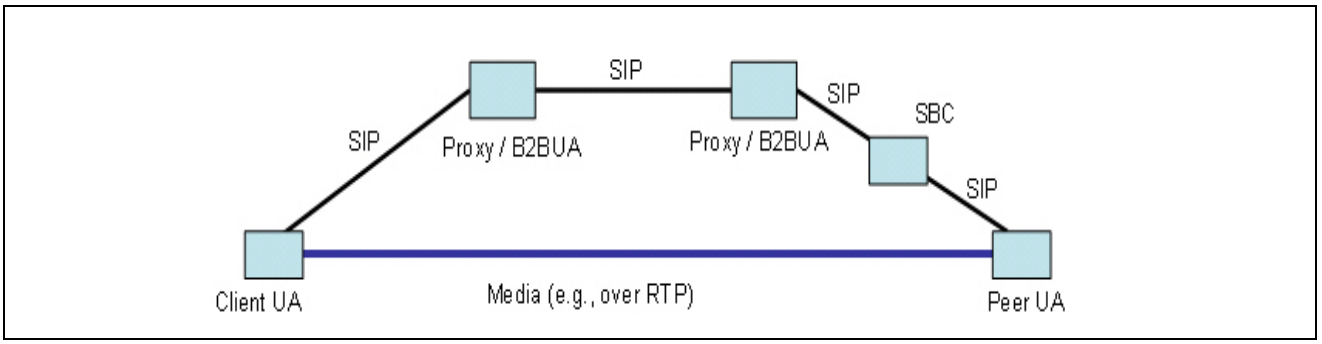


Figure 4.2 SIP Configuration

The SIP client interface is situated between the client and its SIP server. The SIP server incorporates SIP proxy and registrar functionality for the domain concerned, and therefore it is with this entity that the client registers and it is this entity that routes inbound SIP requests to clients. Also outgoing SIP requests from clients are routed via this entity. A SIP server typically behaves as a B2BUA rather than as a pure proxy and registrar.

Note: Although according to RFC 3261, SIP signalling does not always need to pass through a local proxy or B2BUA and can even occur directly between UAs (e.g., when directed to do so by a SIP redirect server or when proxies do not record-route), the descriptions within this document assume signalling always passes through the SIP server.

Some elements of SIP messages (all bodies and certain headers) are of end-to-end significance, passing transparently between any proxies and B2BUAs en route between the client and the peer UA. Although for certain messaging the SIP server may also act as the peer UA (e.g., when it provides interworking to MGCP or when it acts as subscriber or notifier for an event package) and in certain other situations might examine, modify, or generate information that a proxy would not be involved in, in other situations the SIP server merely provides a transparent pipe for end-to-end information. In these other situations, the peer UA is responsible for originating and terminating such information. Where such information is passed on across another instance of the client interface to another SIP client, that second SIP client is required to comply with this specification. When such information is passed across a trunking interface to a UA within another endpoint type such as a gateway, media server or conference unit, corresponding specifications for the trunking interface will place requirements on the endpoint for supporting this end-to-end information.

4.2.2 Clients Interfacing Directly to OpenScape Voice

In one common configuration, a client interfaces directly to OpenScape Voice. In SIP terms, OpenScape Voice acts as the client's outbound proxy. See Figure 4.3, Figure 4.4, and Figure 4.5 for examples of this configuration.

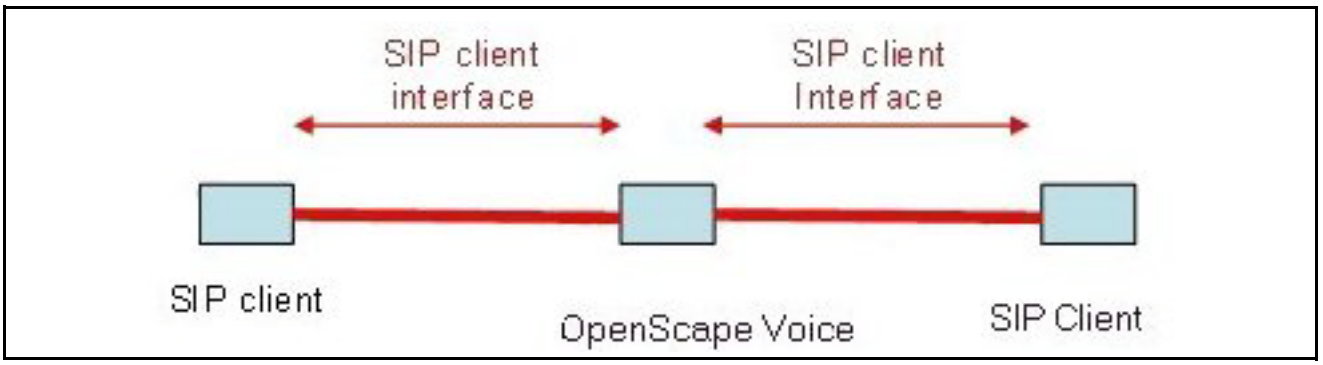


Figure 4.3 Client Interfacing Directly to OpenScape Voice: Peer UA to Same OpenScape Voice

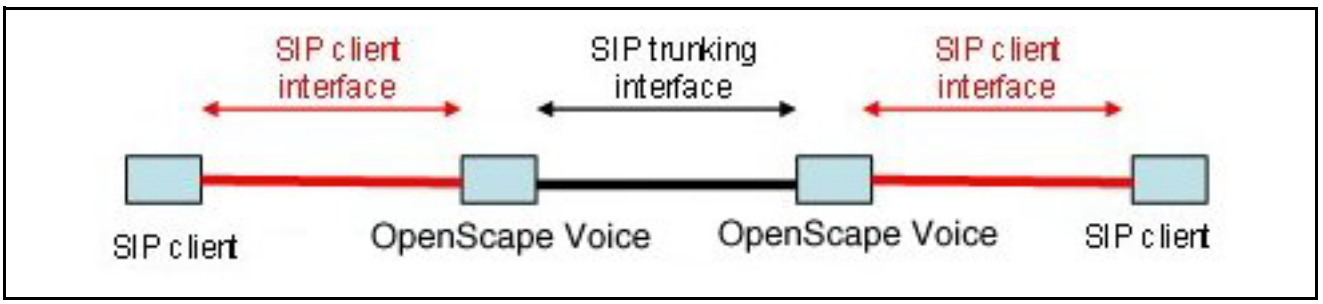


Figure 4.4 Client Interfacing Directly to OpenScape Voice: Peer UA to Another OpenScape Voice

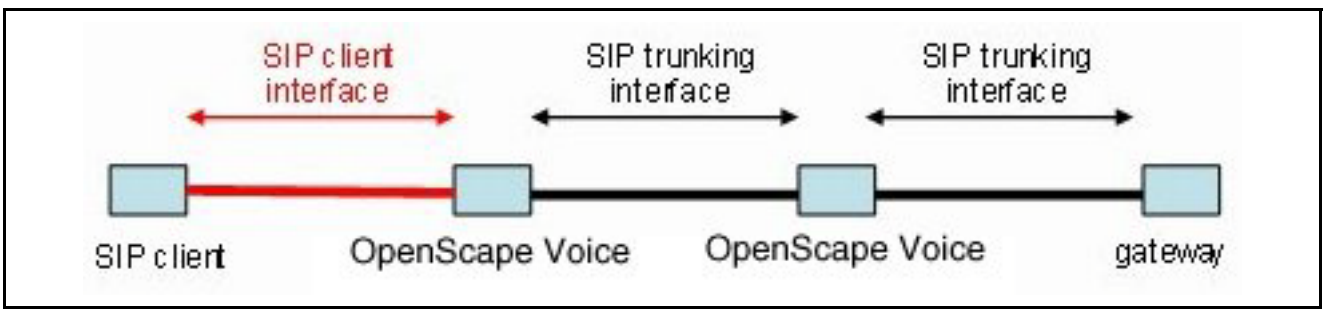


Figure 4.5 Client Interfacing Directly to OpenScape Voice: Peer UA in Gateway to Another OpenScape Voice

4.2.3 Edge Proxy between Client and OpenScape Voice

In other configurations an edge proxy may exist between the client and OpenScape Voice. For the purposes of this specification the term *edge proxy* extends to equipment types that exhibit B2BUA behavior rather than pure SIP proxy behavior, and to equipment types that include media relay capabilities. Examples include session border controllers (SBCs) and application layer gateways (ALGs). This edge proxy-, as opposed to the SIP server, acts as the outbound proxy, and thus the SIP client interface comprises two halves:

- Between the client and the edge proxy
- Between the edge proxy and OpenScape Voice, as shown in Figure 4.6

Technically the two halves will normally be very similar, differing mainly in the transport for SIP. Aspects of this specification relating to the transport of SIP apply between the client and the outbound proxy. Where the outbound proxy is an edge proxy, the transport of SIP between the edge proxy and OpenScape Voice is outside the scope of this specification.

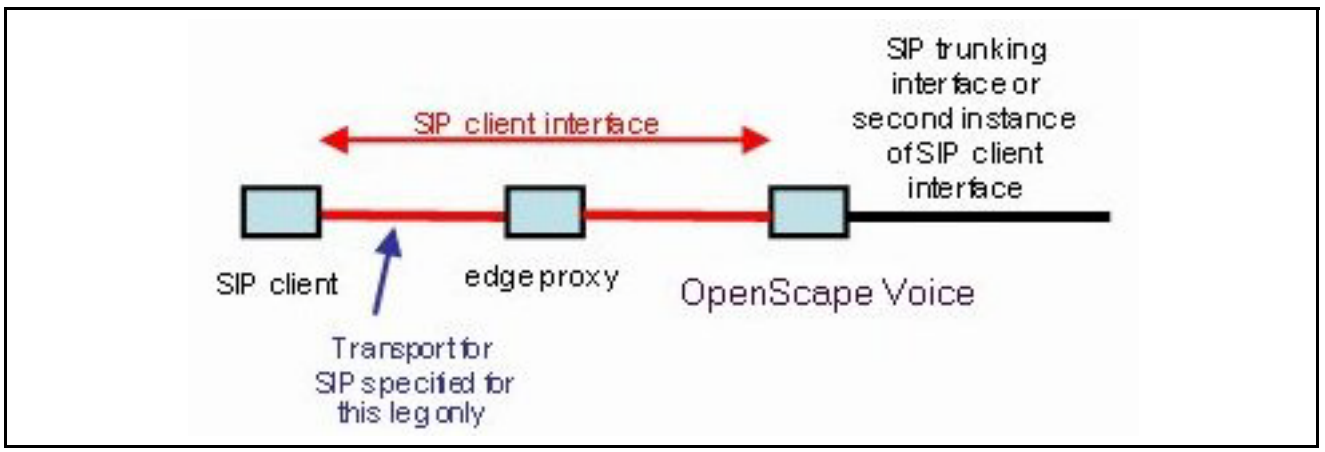


Figure 4.6 SIP Configuration with Edge Proxy

4.2.4 Client with Mobile Appliance

In yet another configuration a mobile appliance is involved. The mobile appliance supports a dual-mode handset that can roam between wireless LAN (WLAN) and cellular (GSM) networks with handover of calls. The mobile appliance acts as SIP client on behalf of the dual mode handset. When in WLAN mode, the dual mode handset communicates with the mobile appliance via a SIP-based interface that is outside the scope of this specification(see Figure 4.7).

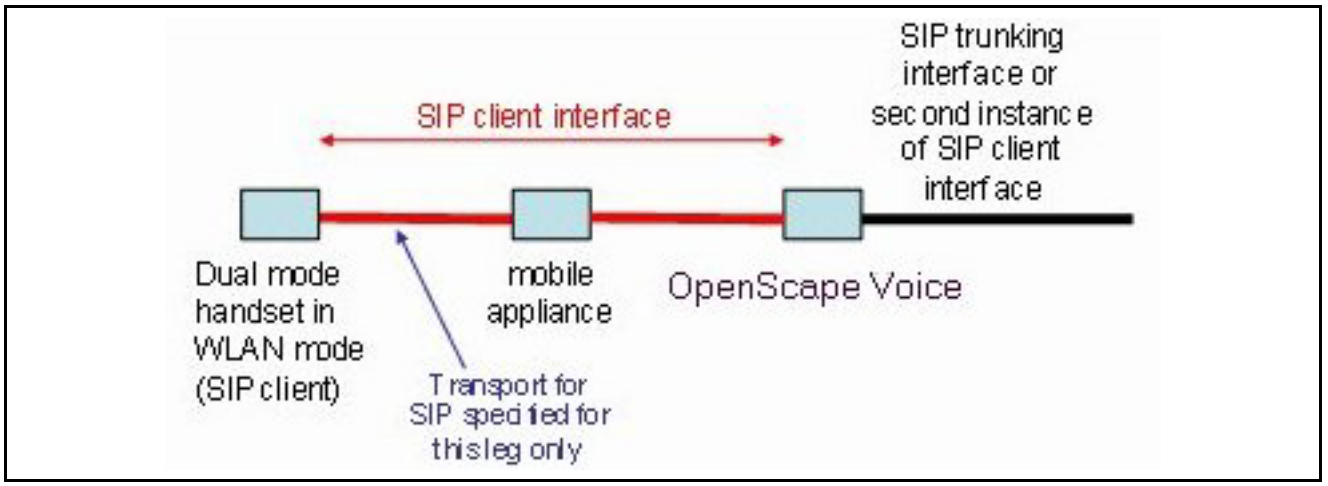


Figure 4.7 SIP Configuration with Mobile Appliance—Handset in WLAN Mode

4.2.5 Client with Mobile GSM Appliance

When in GSM mode, the dual mode handset communicates through the cellular network and the SIP network with the mobile appliance. In this mode the mobile appliance controls two calls back-to-back (signalling and media), i.e., a call leg between the mobile appliance and the handset via the SIP and cellular networks and a call leg between the mobile appliance and the communication partner in the SIP network. Both calls traverse the SIP client interface (see Figure 4.8).

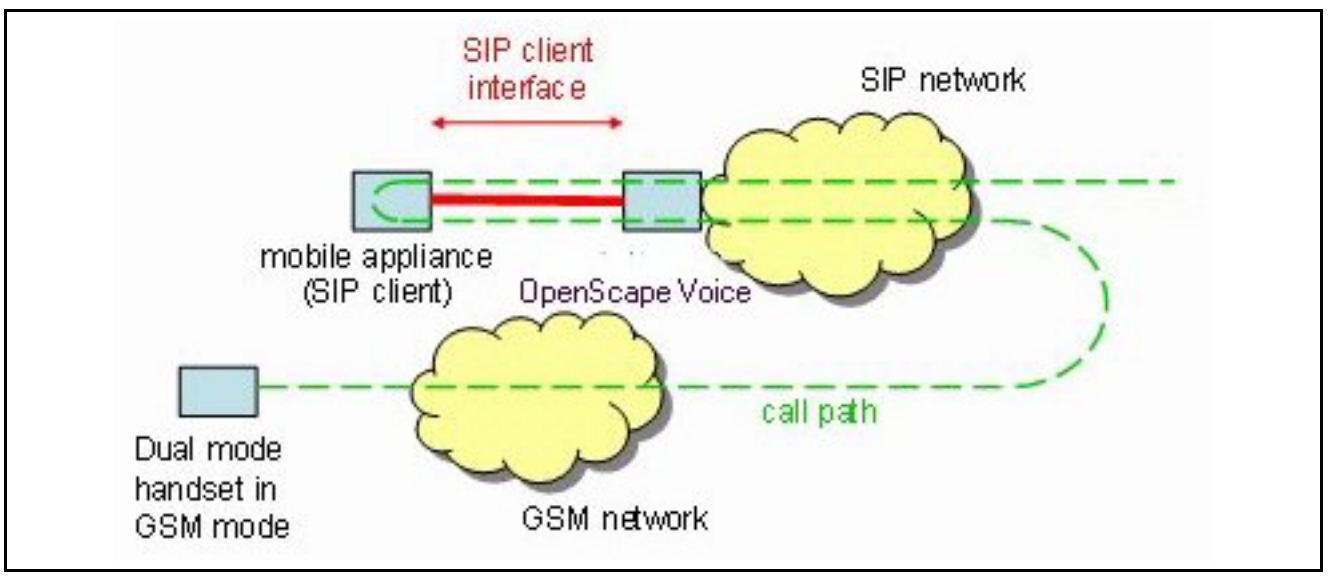


Figure 4.8 SIP Configuration with a Mobile Appliance—Handset in GSM Mode

4.3 Impact of NATs and Firewalls

The SIP client interface can pass through a NAT. This situation is supported only if an SBC or ALG intervenes between the client and OpenScape Voice in order to adjust IP addresses and ports signalled within SIP messages. The SBC or ALG would appear as an edge proxy as described in Sect. 4.1.

Likewise, there can be a NAT in a call path beyond OpenScape Voice. Again this is supported only through the use of SBCs or ALGs. Clients support no special capabilities to aid in NAT traversal.

Note: The support of techniques such as STUN or TURN can enable NAT traversal in certain circumstances. These techniques can be used unilaterally by a client without impacting the SIP client interface, the SIP server or the peer client. The support of ICE can assist more generally but would impact the SIP client interface and is outside the scope of this specification.

The SIP client interface can pass through a firewall provided the firewall is open for SIP signalling initiated in the direction client to server. Likewise media can pass through a firewall provided it is open for media packets flowing from the client and, as a result of this, is open for media packets in the opposite direction. For other firewall situations an SBC or ALG is required.

4.4 Standards Basis

The SIP client interface makes reference to SIP and SDP RFCs wherever possible. For a small number of signalling capabilities RFCs are not available and the SIP client interface uses SIP or SDP extensions specified in stable Internet Drafts or Siemens specifications.

Basic SIP and SDP signalling capabilities can be used to provide a large number of higher level features. The use of SIP for some of these features is described in a number of Informational or Best Current Practice RFCs. The use of SIP for other features is described in various Siemens specifications. This specification references these RFCs and Siemens specifications where appropriate.

The IETF has published a wide range of RFCs relating to SIP and SDP. However, OpenScape Voice supports only a subset of these RFCs. Future support by OpenScape Voice for additional RFCs will depend on market conditions and support by partners and competitors.

5 Requirements for Support of SIP Signaling Capabilities

Clients must support SIP in accordance with RFC 3261 as qualified in the sub-sections below.

OpenScape Voice currently supports IPv6 on the SIP client interface only as a pilot project, full support for IPv6 is planned for a future release.

Clients must behave in accordance with the appropriate standard (e.g., RFC 3261, RFC 3265) when encountering unsupported elements in received SIP messages. This includes, but is not limited to:

- Unsupported SIP methods
- Unsupported SIP header fields
- Unsupported SIP response codes
- Unsupported parameters in certain SIP header fields
- Unsupported URI schemes
- Unsupported URI parameters
- Unsupported body types
- Unsupported event packages
- Unsupported elements within notification bodies
- Unsupported SDP attributes
- Unsupported security options

In the case of OpenScape Voice, behavior in such cases depends on whether it is behaving as a UA or as a proxy in the context concerned. For example, in some cases OpenScape Voice may simply forward an unsupported element when forwarding the message concerned on the next hop (proxy behavior), whereas in other circumstances it may take action such as rejecting the message or discarding the element concerned.

5.1 Transport for SIP

Where an edge proxy is involved between the client and OpenScape Voice, the requirements below for OpenScape Voice relate instead to the edge proxy.

The following transports are specified for use at the SIP client interface: UDP, TCP, and TLS RFC 2246 (over TCP). Procedures for TLS are significantly different from UDP and TCP.

Note: UDP datagrams that exceed the size of the layer 2 maximum transmission unit (MTU), which for Ethernet is 1500 bytes, may or may not be received correctly . Although fragmentation and re-assembly at the IP layer can be used for long datagrams, packet loss or excessive delay can make this unreliable. Therefore RFC 3261 mandates that if a request is within 200 bytes of this maximum (i.e., 1300 bytes), it must be sent using a reliable transport, e.g., TCP. The additional 200 bytes allows the response to be slightly larger than the request. Switching between UDP and TCP on a per transaction basis, depending on the size of the request, is difficult and is not guaranteed to take account of oversize responses. Therefore UDP should not be used when oversize messages are a possibility. Oversize messages are likely when the following features or capabilities are used:

- large centralised conference;
- uaCSTA;
- keysets;
- call pick-up;
- video media;
- best effort SRTP;
- IPv6.

When interworking with other vendors' devices, oversize messages are possible due to the large numbers of payload types and attribute lines that may be present in SDP even for basic calls.

5.1.1 Client Behavior

A client must support UDP and TCP. A client must support TLS if it supports media security and may support TLS otherwise.

A client must have the ability to be configured with the transport type to be used.

A client must have the ability to be configured with the location (host name or IP address) of the SIP server.

Note: The SIP server location is used as the location of the SIP registrar and, when applicable, the location of the outbound proxy. The SIP server location may be that of an edge proxy rather than OpenScape Voice.

A client must have the ability to be configured with the port to be used at the SIP server and must have the ability to be configured with a local port for SIP signalling.

Note: In either case the port may need to change according to whether UDP/TCP or TLS is used. The default port for UDP/TCP is 5060 and for TLS is 5061.

Note: For clients that support user mobility, the configurable information above may be user-dependent and may change when a user logs on or off.

5.1.1.1 UDP and TCP

A client must have the ability to be configured to operate with outbound proxy on or off when configured for UDP or TCP transport.

When sending any SIP request to the server, a client must behave in accordance with RFC 3261 and in accordance with RFC 3263 for obtaining the SIP server IP address and port using DNS SRV records and A or AAAA records. There is no requirement to support Naming Authority Pointer (NAPTR) records.

Note: This involves DNS lookup if information available from configuration (SIP server location and port for registration or if outbound proxy on), from the topmost Route header field entry (for mid-dialog requests) or from the Request-URI is insufficient.

Note: These procedures can lead the client to use one of a number of redundant or load-sharing servers. This is not to be confused with the backup server concept.

When performing DNS lookup, a client should restrict itself to the configured transport type (UDP or TCP).

Note: Information obtained from DNS lookup will not impact the content of the SIP request, other than the transport type in the Via header field.

A client configured for TCP should retain a TCP connection for a reasonable time (e.g. a few minutes) for use for subsequent requests to the same SIP server.

A client configured for UDP or TCP must be capable of receiving incoming SIP requests using the configured transport and may be capable of receiving incoming SIP requests using the other transport. In the case of TCP, incoming requests can arrive on new TCP connections or on existing TCP connections.

Note: In future there may be a requirement to support draft-ietf-sip-outbound, thereby facilitating NAT traversal and reuse of TCP connections for requests in the reverse direction.

A client must use the same local port for sending and receiving UDP.

Note: This can help with NAT and firewall traversal by allowing a binding or hole to be valid for UDP packets in the reverse direction.

5.1.1.2 TLS

A client that supports TLS transport, and is configured for TLS transport for the server, must also have a SIP server Certification Authority (CA) certificate configured. A client must be able to use the following cipher suite:

"TLS_RSA_WITH_AES_128_CBC_SHA if configured for 128-bit encryption.

A client must be able to verify that a certificate received from the server during TLS handshake is signed by a CA whose certificate is configured.

A client must establish a TLS connection in order to send the initial REGISTER request and retain it for all future requests and responses to that server. A client must not accept a request or response received on any other transport (except in the context of survivability).

When sending an initial REGISTER request to establish a TLS connection to the SIP server a client must behave in accordance with RFC 3263 for deriving the SIP server IP address and port using DNS SRV records and A or AAAA records if not available from configured information. There is no requirement to support NAPTR

records. When establishing a TLS connection, a client must verify that the certificate received from the SIP server is valid and signed by the CA whose certificate is configured.

After establishing a TLS connection, a client must periodically check that the connection remains operational using the procedure in Appendix A, “TLS Connectivity Checking”.

5.1.2 OpenScape Voice Behavior

OpenScape Voice has the ability to use UDP, TCP, and TLS for communication with clients and has the ability to be configured with a private key and certificate for supporting clients that communicate using TLS. Each device (subscriber or endpoint) can be configured to accept a minimum transport type. See details in the following section.

OpenScape Voice has the ability to be configured with a local port for SIP signalling using TCP or UDP (default 5060) and the local port for TLS (default 5061).

5.1.2.1 Transport Type

If the subscriber or network endpoint is configured in OpenScape Voice to use a particular transport type, requests received with a “higher” transport type are also accepted. For example, when configured to use UDP, TCP requests are also accepted.

Requests using a “lower” protocol are rejected. For example, when configured to use TCP, UDP requests are rejected.

This determination is done when the device registers with the OpenScape Voice Server. If it attempts to register with a “lower” protocol, the registration request is rejected.

The accepted transport types are listed in Table 5.1.

Subscriber/Endpoint Configured For...	OpenScape Voice Server Will Accept...
UDP	UDP, TCP, or TLS
TCP	TCP or TLS
TLS	TLS only

Table 5.1 Accepted Transport Types

When using TCP, OpenScape Voice will establish a TCP connection to a client in order to send a request to a registered contact if no suitable TCP connection already exists. If an active TCP connection to the client exists, OpenScape Voice will reuse the connection.

Note: In OpenScape Voice it is recommended to use TCP transport for SIP subscriber endpoints and network endpoints. While OpenScape Voice does not enforce the RFC 3261 mandate for UDP mentioned in 4.2 above, to minimize possible side effects, the default transport type for new SIP clients added to the configuration becomes TCP. The selection for TCP transport type remains available and existing SIP endpoint configurations are not impacted.

5.1.2.2 TLS

OpenScape Voice is able to use the following cipher suite:

"TLS_RSA_WITH_AES_128_CBC_SHA if configured for 128-bit encryption.

If OpenScape Voice is configured for the use of TLS (private key and certificate available), it is able to accept a TCP connection from a client on the TLS port and establish TLS using the certificate concerned. OpenScape Voice does not require the client to supply a certificate.

Subject to successful SIP registration (including authentication of the client at the SIP level), OpenScape Voice will retain a TLS connection and use it for all subsequent SIP requests in either direction to and from the client concerned and will not accept SIP requests and responses received on any other transport.

After establishing a TLS connection, OpenScape Voice will respond to connectivity checks by the client as specified in Appendix A, “TLS Connectivity Checking”.

5.1.3 Survivability

In certain OpenScape Voice deployment scenarios, SIP clients may need to support use of a backup server to ensure survivability. These procedures are described in other Siemens specifications that are available upon request, but are outside the scope of this document.

5.2 SIP URIs

The following URI schemes are supported:

- SIP

- SIPS
- TEL (optional)

OpenScape Voice places the following restrictions on the size of URI elements:

- Display Name - max. 39 characters
- User Part - max. 128 characters
- Host Part - IP address or FQDN
- URI Parameters - max. 128 characters

5.2.1 SIP URIs

Clients must support SIP URIs (RFC 3261) in Request-URIs and in all fields of SIP headers and bodies in which a URI can occur, unless stated otherwise elsewhere in this specification.

Clients must support an IP address and an FQDN in the host part of a SIP URI in all Request-URIs and in all fields of SIP headers and bodies in which a SIP URI can occur, unless stated otherwise elsewhere in this specification.

For the purposes of this specification, a SIP URI in which the user part contains a telephone-subscriber string as defined in RFC 3966 is called a *type A SIP URI*. A type A SIP URI may or may not contain a user=phone parameter.

Note: The user part of a type A SIP URI will contain a telephone number (i.e., digits), either preceded by a '+' character (in the case of a global E.164 number) or followed by a phone-context parameter. However, OpenScape Voice does not currently support use of the phone-context parameter.

For the purposes of this specification, a SIP URI in which there is no user=phone parameter and in which the user part contains a string that is not a telephone-subscriber string is called a type B SIP URI.

Note: The user part of a type B SIP URI can be numeric (but not a telephone-subscriber string) or partly or wholly non-numeric.

For usages other than those specified in the subsections below, clients must support type A and type B SIP URIs.

Note: Where just transmitting a received URI it may not be possible for a server to determine whether a number is fully qualified, e.g., if the server has received it from a remote entity.

In accordance with RFC 3261, clients and servers must NOT generate SIP URIs having a transport parameter with value 'tls' (transport=tls).

Note: This does not preclude a client or server sending a SIP URI with transport=tls if merely passing on an opaque URI received from elsewhere.

Note: To express a requirement to be reached only via TLS, a SIPS URI should be used.

Clients must preserve any URI parameters in a received URI and forward them when forwarding the URI (including any unrecognised parameters).

Note: For a client this applies particularly to URIs received in the Record-Route header field that are forwarded in the Route header field, and to URIs received in the Contact header field of a 3xx response that are used in the Request-URI of a recursed request.

5.2.1.1 Additional Requirements for Client Addresses of Record (AoRs)

A client must support type B SIP URIs and should support type A SIP URIs as AoRs of its users. In the case of type B SIP URIs, a client must support both numeric and non-numeric user parts.

Note: A client will be configured with its AoRs, to match those configured at the server, and hence in a given deployment will only be required to use formats supported by the server.

5.2.1.2 Additional Requirements for Client Contact URIs

A client must support type B SIP URIs as its contact URIs. A client must be able to use numeric user parts in its contact URIs when operating with a server that does not support non-numeric user parts.

Note: Typically a client constructs its contact URIs algorithmically, e.g., by using the same user part as the AoR but changing the domain part. In such cases a numeric user part in an AoR will result in a numeric user part in the corresponding contact URI.

5.2.1.3 Additional Client Requirements for Transmitted Destination URIs

When a client issues a request that is not a mid-dialog request, the user part of a SIP Request URI or a To header field URI will depend on what is entered by the user and any manipulation by the client. The same applies to the Refer-To header field URI when issuing a REFER request. If the user enters a complete URI (e.g., from a phone book, web page or call log), the client should send this unchanged. Otherwise, if the user has just entered a character string, the client must convert this to a SIP URI by adding the server's FQDN or IP address (without the user=phone parameter). A client may have the capability to be configured with dial plan information that allows it to convert character strings representing partial telephone numbers to fully qualified telephone numbers, thereby allowing fully qualified telephone numbers to be sent to the server. In the absence of this, users will be required to enter character strings in a form acceptable to the server.

Note: OpenScape Voice will treat these as E.164 numbers for routing/translation purposes if the user=phone parameter is present or only characters 0-9, A-F,*,# are present, otherwise will be treated as an alias.

5.2.2 SIPS URIs

Clients and servers should support SIPS URIs (RFC 3261, updated by RFC 5630) and meet the same requirements as for SIP URIs.

Clients should support the receipt of SIPS URIs in appropriate fields (e.g., in the From header field or the Contact header field). Clients should be able to issue a mid-dialog request to a SIPS URI if a SIPS URI has been received in a Contact header field.

Note: Clients are not required to use SIPS URIs for other purposes, e.g., establishing a call to a SIPS URI or registering a SIPS URI as a contact. The SIPS URI scheme is not specified clearly in the RFCs concerned and is subject to ongoing discussion in the IETF. Clients should not draw any conclusions about the security of signalling from receipt of a SIPS URI or receipt of a positive response to a request to a SIPS URI.

Servers should support the sending and receipt of SIPS URIs but are not required to support registration using SIPS URIs as AoRs or contacts.

5.2.3 TEL URIs

Clients and servers may support TEL URIs (RFC 3966) in fields of SIP headers and bodies where they are permitted (e.g., not as contact URIs, not as To or From field URIs in a REGISTER request). There is no requirement for support of TEL URI parameters.

Note: The phone-context parameter for partial E.164 or private numbers is not currently supported by OpenScape Voice; however, support may be included in a future release.

The requirements of Sect. 5.2.1 concerning fully qualified telephone numbers apply equally to TEL URIs.

5.3 Registration

Clients and servers must support SIP registration (including refresh) in accordance with RFC 3261and should not use third-party registration. Clients must be able to be configured with the registration time-to-live value. A client should attempt a registration refresh a reasonable time (e.g., 10s) before expiry is due.

When sending an initial REGISTER request (as opposed to a refresh), the client should ensure that the Call-ID header field has a random value.

Note: This helps to avoid possible clash with a value used for a previous registration that is still current at the server. This allows the server to distinguish an initial registration from a refresh and to take relevant actions (e.g., to trigger another message waiting indication).

In the event of no response to a REGISTER request or a 500 or 503 response without a Retry-After header field to a REGISTER request, a client must retry after an interval of 0.5 seconds initially, doubling this interval for each subsequent retry up to an interval of 32 seconds. Thereafter a client must retry periodically, each retry following the previous retry by a random interval of between 1 minute and n minutes, where n is either fixed at 10 or configurable. In the event of a 500 or 503 response with a Retry-After header field to a REGISTER request, a client must act upon the Retry-After header field as specified in Sect. 5.11.33.

See Sect. 5.11.33 for additional requirements concerning the Retry-After header field.

If a client supports survivability (e.g., relating to the use of a back-up server) there are additional requirements documented in other Siemens specifications (available on request) that are outside the scope of this document.

5.3.1 Support of Backup Server for Survivability

If a client supports survivability (e.g., relating to the use of a backup server), there are additional requirements documented in other Siemens specifications (available on request) that are outside the scope of this document.

5.4 Digest Authentication

Clients and servers must support HTTP digest authentication in accordance with RFC 3261. User-to-user authentication is used between a client and the server, but clients are also required to support proxy authentication for use when proxies are involved.

5.4.1 Client Behavior

A single line (non-keyset) client must be able to be configured with the authentication identity, realm and shared secret for the AoR against which it registers as a contact. A multi-line (keyset) client must be able to be configured with said information separately for each line (primary line and secondary lines).

Note: For a line that appears on multiple clients, all clients will need to be configured with the same credentials.

On receipt of a challenge for user-to-user authentication (WWW-Authenticate header field in a 401 or 407 response), if the client has credentials for the realm concerned the client must re-issue the request containing appropriate authentication (Authorization header field), The client should also cache the nonce and use it to generate authentication in subsequent requests where appropriate.

A client may challenge certain incoming requests using the same realm and expecting the same credentials that it uses for responding to challenges on outgoing requests.

On receipt of a challenge for proxy authentication (Proxy-Authenticate header field in a 407 response), if the client has credentials for the realm concerned the client must re-issue the request containing appropriate authentication (Proxy-Authorization header field), The client should also cache the nonce and use it to generate authentication in subsequent requests on the same dialog.

5.4.2 OpenScape Voice Behavior

OpenScape Voice is able to be configured with the authentication identity, realm and shared secret for each AoR for which it accepts registrations.

Where TLS transport is not used, OpenScape Voice will require authentication on a REGISTER request and may require authentication on other requests from the client. Where TLS transport is used, OpenScape Voice will require authentication on the first REGISTER request after connection establishment and may require authentication on other requests over that same connection. OpenScape Voice will change the nonce periodically.

If authentication is required on a request and is not received, or is received with wrong parameters (e.g., wrong nonce or realm), OpenScape Voice will issue a challenge for user-to-user authentication (WWW-Authenticate in a 401 response). After a provisional number of attempts OpenScape Voice will reject a request with wrong credentials using a 403 Forbidden response.

5.5 Handling of Bodies

Clients and servers must support the use of SIP message bodies in accordance with RFC 3261. Servers and clients must support the use of multipart MIME for conveying multiple bodies.

Note: This specification does not include any features that require clients to support multipart MIME.

OpenScape Voice will not send to clients body types that have no relevance to clients, e.g., QSIG.

There is no requirement to support S/MIME.

5.6 INVITE-Initiated Dialogs

OpenScape Voice does not create multiple early dialogs with a client acting as a UAC as a result of forking, except where dictated by other requirements.

Note: Where a server performs forking, it should hide this from the client acting as UAC. Similarly, where an entity downstream from a server performs forking and exposes multiple early dialogs to the server, the server should hide these from the client. However, it is recognised that in some equipment types that become a server only in survivability mode, multiple early dialogs might be created. Also some server types may be required to operate in proxy mode in certain deployments, and hence will expose multiple early dialogs to the client.

Clients should be able to support the creation of multiple early dialogs per INVITE request when acting as a UAC and should undergo tests to determine the maximum number that can be supported.

OpenScape Voice supports at least two concurrent active sessions (i.e. sessions in which media flows).

Note: This is required by the mobile appliance and in support of phone-based conferences.

5.7 Session Timers

Session Timers are defined in RFC 4028. Clients capable of supporting session timers must be configured with session timing disabled. OpenScape Voice always performs session timing even in this configuration, able to detect stale sessions to be released.

5.8 SIP Event Framework

Clients and servers must support the event framework specified in RFC 3265 as both subscriber and notifier to the extent needed for those event packages supported (see Chapter 7).

Note: Some event packages do not use the SUBSCRIBE method but rely on implicit subscriptions.

Note: For some event packages a client or server will act as subscriber or notifier, but not both.

Note: For certain event packages and depending on policy or other circumstances, a server may forward SUBSCRIBE and NOTIFY requests and responses rather than acting as a subscriber or notifier itself.

5.9 SIP Methods

The methods discussed below are all defined in RFC 3261 unless another reference is given.

There is no requirement to support any of the methods not listed here. This means that:

- There is no requirement for a client to send such a method in a request.
- There is no requirement for OpenScape Voice to send such a method to a client in a request.
- There is no requirement for a client or OpenScape Voice to take special action on receipt of such a method in a request, other than the default behavior for unsupported methods (refer to Sect. 5).

5.9.1 ACK Method

Clients and servers must support sending and receipt of the ACK method in accordance with RFC 3261.

5.9.2 BYE Method

Clients and servers must support sending and receipt of the BYE method in accordance with RFC 3261.

5.9.3 CANCEL Method

Clients and servers must support sending and receipt of the CANCEL method in accordance with RFC 3261.

5.9.4 INVITE Method

Clients and servers must support sending and receipt of the INVITE method in accordance with RFC 3261, including the following:

- Sending and receipt of an INVITE request outside the context of an existing dialog to initiate a session
- Sending and receipt of an INVITE request within the context of an existing dialog (re-INVITE) to modify a session (e.g., for purposes such as call hold)

5.9.5 NOTIFY Method (RFC 3265)

5.9.5.1 Client Behavior

Clients that support event packages as notifier must support the sending of a NOTIFY request.

Clients that support event packages as subscriber must support the receipt of a NOTIFY request. A client that issues a SUBSCRIBE request must be prepared to receive NOTIFY requests on multiple dialogs as a result of the SUBSCRIBE request being forked.

Note: Some event packages use NOTIFY requests outside the context of a dialog, contrary to RFC 3265.

5.9.5.2 OpenScape Voice Behavior

OpenScape Voice supports event packages as a notifier and supports the sending of a NOTIFY request.

OpenScape Voice supports the receipt of a NOTIFY request.

Note: Some event packages use NOTIFY requests outside the context of a dialog, contrary to RFC 3265.

Note: For the basic capabilities described in this specification, OpenScape Voice is not required to support relaying of event packages or receipt of NOTIFY requests for relaying.

5.9.6 OPTIONS Method

Clients and servers must respond to a received OPTIONS request.

Note: A 2xx response with no header fields (other than those that are mandatory for any 2xx response) and with no bodies is sufficient. This allows the presence of a UA to be determined. There is no requirement to comply fully with RFC 3261.

Clients and servers may issue OPTIONS requests.

Note: When an OPTIONS request is received within a dialog OpenScape Voice treats it as just a 'ping' and sends a response without checking any headers such as Supported etc. SIP Session Timing procedures [Sect. 5.7] can be also used as a session keep alive mechanism.

5.9.7 REFER Method (RFC 3515)

5.9.7.1 Client Behavior

Clients must support the sending and receipt of REFER requests in which the method to be used for dereferencing is INVITE.

Note: OpenScape Voice acts as a UAS for any received REFER requests, and will not pass them on to clients. Therefore client support for receipt of REFER requests, if provided, is for use in other environments.

5.9.7.2 OpenScape Voice Behavior

OpenScape Voice supports receipt of REFER requests in which the method to be used for dereferencing is INVITE.

5.9.8 REGISTER Method

5.9.8.1 Client Behavior

Clients must support the sending of REGISTER requests in accordance with RFC 3261.

5.9.8.2 Server Behavior

Servers must support the receipt of REGISTER requests in accordance with RFC 3261.

5.9.9 SUBSCRIBE Method (RFC 3265)

5.9.9.1 Client Behavior

Clients that support as subscriber event packages that use the SUBSCRIBE method must support the sending of a SUBSCRIBE request.

Clients that support as notifier event packages that use the SUBSCRIBE method must support the receipt of a SUBSCRIBE request.

See Sect. 5.11.33 for additional requirements concerning the Retry-After header field.

5.9.9.2 OpenScape Voice Behavior

OpenScape Voice supports as subscriber event packages that use the SUBSCRIBE method and supports the sending of a SUBSCRIBE request.

OpenScape Voice supports as notifier event packages that use the SUBSCRIBE method and supports the receipt of a SUBSCRIBE request.

Note: For the basic capabilities described in this specification, OpenScape Voice is not required to support the sending of relayed SUBSCRIBE requests and the receipt of SUBSCRIBE requests for relaying.

5.9.10 The UPDATE method (RFC 3311)

5.9.10.1 Client behavior

Clients must support the receipt of UPDATE requests and should support the sending of UPDATE requests.

5.9.10.2 OpenScape Voice behavior

OpenScape Voice does not currently support the UPDATE method except in the special case of sending an UPDATE request without SDP, in an established dialog, to a client in order to update connected party display information.

5.10 Request Line

5.10.1 Client Behavior

When sending a request outside the context of an existing dialog where the required destination is selected by user input (e.g., by dialling or by keying into a phone book entry and then using that for dialing), a client must be able to send as Request-URI a SIP URI containing a telephone number in the user-info part and the FQDN or IP address of the server in the host part, may be able to send other forms of SIP URI (e.g., name@domain) and may be able to send a TEL

URI. When sending any request outside the context of an existing dialog where the required destination is obtained from elsewhere (e.g., from the caller identity of a previous incoming call), a client should be able to send any form of URI listed in Sect. 5.2 as the Request-URI. However, this will be constrained by the particular request type (e.g., RFC 3261 lays down particular requirements for REGISTER requests and the destination of a SUBSCRIBE request will depend on the event package concerned).

Note: When sending a request within an existing dialog, RFC 3261 requires the client to use the received remote target URI as the Request-URI.

A client must be able to accept in a received request a Request-URI that matches any URI given as contact when registering.

5.10.2 OpenScape Voice Behavior

When receiving a request outside the context of an existing dialog, OpenScape Voice is able to route on a SIP Request-URI containing a telephone number in the user-info part and the FQDN or IP address of the server in the host part. OpenScape Voice is also able to route on other forms of Request-URI listed in Sect. 5.2.

Note: When receiving a request within an existing dialog, RFC 3261 requires the server (OpenScape Voice) to accept any URI that it has sent to the client as a contact URI.

OpenScape Voice will send as a Request-URI only a URI that the client has registered as a contact.

5.11 SIP Header Fields

The header fields discussed below are all defined in RFC 3261 unless another reference is given.

There is no requirement to support any of the header fields not listed here. This means that:

- There is no requirement for a client to send such a header field.
- There is no requirement for OpenScape Voice to send such a header field to a client.

- There is no requirement for a client or server(OpenScape Voice) to take special action on receipt of such a header field, other than the default behaviour for unsupported header fields.

For each header field listed below, unless otherwise stated, the requirements to use it in particular requests and responses are as stated in RFC 3261 or the RFC for the method or header field concerned (whichever was published later).

5.11.1 Accept

Clients and servers must support sending of the Accept header field in relevant requests and 2xx responses when able to accept body types other than the default in the same transaction or dialog and in 415 responses when able to accept body types other than the default in a repeat request.

Note: For example, in a SUBSCRIBE request it is only necessary to indicate any body types acceptable in the SUBSCRIBE response or NOTIFY requests relating to that subscription.

Clients and servers must support receipt of the Accept header field in requests and 2xx responses and 415 responses and should avoid sending unacceptable body types in subsequent messages in the same transaction or dialog or in repeat requests.

5.11.2 Alert-Info

5.11.2.1 Client Behavior

A client must be capable of being configured with a set of strings (case-insensitive) identifying special types of audible alerting. By default, a client must be configured with at least the following strings and meanings, but these can be modified or removed for particular deployments (e.g., outside the USA):

- ""<Bellcore-dr1>" - normal (internal) alerting
- ""<Bellcore-dr2>" - external alerting
- ""<Bellcore-dr3>" - recall alerting (e.g., following transfer or call-back)
- ·"<alert-emergency>" - emergency / urgent alerting.

A client that is configured in this way must be able to apply a special type of audible alerting if it receives an INVITE request, or an UPDATE request while already alerting, containing an Alert-Info header field whose URI either has the

corresponding string as its value or has parameter 'info' with the corresponding string as its value (e.g. Alert-Info: <Bellcore-dr1> or Alert-Info: uri;info=<Bellcore-dr1>).

Note: OpenScape Voice does not use the **info**=format.

Likewise, a client that is configured in this way may be able to apply a special ring-back tone if in response to an INVITE request it receives a 180 response containing an Alert-Info header field whose URI meets the conditions above, subject to conditions for playing ringback tone being met.

A client must be capable of auto-answering a call or automatically retrieving from hold (see Sect. 8.2.1) if the INVITE or re-INVITE request contains an Alert-Info header field whose URI has an info parameter with the specific value "alert-autoanswer". In this case, if the URI also has a delay parameter with a value greater than 0, the client must answer or retrieve the call after that number of seconds (e.g., Alert-Info: <uri>;info=alert-autoanswer;delay=5). Otherwise the client must answer or retrieve the call immediately (e.g., Alert-Info: <uri>;info=alert-autoanswer).

There is no requirement for a client to contact the URI concerned.

5.11.2.2 OpenScape Voice Behavior

A server may support the sending of an Alert-Info header field in an INVITE request containing the 'info' and/or the 'delay' URI parameter.

5.11.3 Allow

Clients and servers must support sending and receipt of the Allow header field in accordance with RFC 3261.

Clients and servers should avoid sending requests for unsupported methods to the peer UA.

Note: In the case of OpenScape Voice, the Allow header field will normally indicate the OpenScape Voice's own capabilities, rather than that of the peer UA.

5.11.4 Allow-Events (RFC 3265)

Clients and servers must send an Allow-Events header field in 489 responses. The use of the Allow-Events header field in other messages is optional except where stated otherwise for specific event packages. OpenScape Voice supports the call completion to busy subscriber (CCBS) and call completion on no reply (CCNR) features (see Sect. 8.11), and may send an Allow-Events header field indicating 'ccbs' in a 486 response to an INVITE request and 'ccnr' in a 180 response to an INVITE request to indicate the availability of CCBS and CCNR respectively.

Clients and servers may make use of information in a received Allow-Events header field. In particular, clients may make use of event 'ccbs' or 'ccnr' in an Allow-Events header field as an indication that the feature concerned is available to the destination concerned.

Note: The 'ccbs' and 'ccnr' event packages are not available to a client (they operate only across the trunking interface), so CCBS and CCNR features have to be invoked by other means.

5.11.5 Authentication-Info

Clients and servers should support receipt of an Authentication-Info header field in a 2xx response and may support sending of an Authentication-Info header field in a 2xx response, in accordance with RFC 3261.

5.11.6 Authorization

5.11.6.1 Client Behavior

Clients must support sending an Authorization header field in a request in accordance with RFC 3261.

5.11.6.2 OpenScape Voice Behavior

OpenScape Voice supports receipt of an Authorization header field in a request in accordance with RFC 3261.

5.11.7 Call-ID

Clients and servers must support sending and receipt of the Call-ID header field in accordance with RFC 3261.

5.11.8 Call-Info

5.11.8.1 Client behavior

If the client is unable to support uaCSTA for CSTA Answer call-back recalls, OpenScape Voice is able to provide the necessary indications using the Call-Info header field "info-param" value generic-param syntax of RFC 3261.

The syntax extensibility of "info-param" allows definition of a new "answer-after" generic-param. When present in the Call-Info header of an incoming INVITE request, it indicates how many alert cycles should be applied by the UAS before the call is automatically answered. For example,

```
Call-Info: answer-after=1
```

indicates that the call should be automatically answered after 1 alert cycle.

5.11.8.2 OpenScape Voice behavior

For the client call offer feature, OpenScape Voice supports the reception and forwarding of the Call-Info header field within the 486 response between the called and calling client.

For CSTA Answer call-back recalls, OpenScape Voice is able to provide an "answer-after" indication within the Call-Info header field for clients which are unable to support the uaCSTA interface,

5.11.9 Contact

Clients and servers must support sending and receipt of the Contact header field in accordance with RFC 3261 for communicating targets in the context of dialogs. Refer to Sect. 5.2 for the use of a contact URIs in this header field.

Clients and servers must support the use of this header field in a 3xx response.

Note: In this case the URI in the Contact header field is not necessarily a contact URI but could be an AoR.

5.11.9.1 Additional Behavior for Clients

A client must take account of the 'expires' parameter in the Contact header field of a 200 response to a REGISTER request in order to refresh registrations at appropriate intervals.

A client that supports conferencing (see Sect. 8.7) must support receipt of the 'isfocus' media feature tag as defined in RFC 3840 and, if able to host a local conference, must support sending this tag.

5.11.9.2 Additional Behavior for OpenScape Voice

OpenScape Voice takes account of any 'expires' parameter in the Contact header field of a REGISTER request. OpenScape Voice includes the 'expires' parameter in the Contact header field of a 200 response to a REGISTER request.

OpenScape Voice supports sending and receipt of the "isfocus" media feature tag.

5.11.10 Content-Disposition

Clients and servers must support receipt of the Content-Disposition header field in accordance with RFC 3261 and must support the default value of 'session' (for SDP bodies) and 'render' (for other bodies, including defaulting to these when the header field is missing. Clients and servers must support sending the Content-Disposition header field in accordance with RFC 3261] if values other than defaults are used. These requirements apply also to this header field within a multi-part MIME body indicating the content disposition of a body part.

5.11.11 Content-Length

Clients and servers must support sending and receipt of the Content-Length header field in accordance with RFC 3261].

5.11.12 Content-Type

Clients and servers must support sending and receipt of the Content-Type header field in accordance with RFC 3261 and must support the value 'application/SDP' and any types used by supported event packages. This also applies to this header field within a multi-part MIME body indicating the content type of a body part.

5.11.13 CSeq

Clients and servers must support sending and receipt of the CSeq header field in yaccordance with RFC 3261.

5.11.14 Diversion (SIP DIVERSION)

5.11.14.1 Client Behavior

Clients must support the sending of a Diversion header field in a 3xx response to an INVITE request and must include the client's AoR as the diverting URI. Clients must support receipt of a Diversion header field containing any form of URI listed in Sect. 5.2 in an INVITE request.

5.11.14.2 OpenScape Voice Behavior

OpenScape Voice supports receipt of a Diversion header field containing the client's AoR as the diverting URI in a 3xx response to an INVITE request. OpenScape Voice supports the sending of a Diversion header field in an INVITE request with any form of URI listed in Sect. 5.2.

5.11.15 Event (RFC 3265)

Clients and servers must support sending and receipt of the Event header field in accordance with RFC 3265 and with any event packages that are supported (see Chapter 7)

5.11.16 Expires

A server should take account of an Expires header field in a REGISTER request.

Clients and servers must support sending and receipt of the Expires header field in SUBSCRIBE requests and 200 responses to SUBSCRIBE requests in accordance with RFC 3265 and with any event packages that are supported.

Note: OpenScape Voice provides system wide configuration options to enable randomization of the Expires time sent in 200 OK responses to REGISTER and SUBSCRIBE requests.

5.11.17 From

Clients and servers must support sending and receipt of the From header field in accordance with RFC 3261.

Clients should include an epid parameter [Sect. 5.14.6] when sending a REGISTER or INVITE request on behalf of its sole line (single line or non-keyset client) or its primary line (keyset client) and must NOT include this parameter when sending such requests on behalf of a secondary line.

Note: In the case of a REGISTER request, this assists the server in determining whether a registration with a new contact URI is to replace an existing registration or co-exist with an existing registration. The inclusion of the epid parameter in an INVITE request is for future use.

5.11.18 Geolocation

Not currently supported by OpenScape Voice.

5.11.19 Max-Forwards

Client must support sending the Max-Forwards header field in accordance with RFC 3261 and should use the initial value recommended in RFC 3261. There is no requirement for a client to act on this header field on receipt.

Servers must support sending and receipt of the Max-Forwards header field in accordance with RFC 3261].

5.11.20 Min-SE (RFC 4028)

Clients and servers must support sending and receipt of the Min-SE header field.

5.11.21 P-Asserted-Identity (RFC 3325)

5.11.21.1 Client Behavior

Clients should support receipt of the P-Asserted-Identity header field with any form of URI listed in Sect. 5.2 in INVITE and UPDATE requests and in 1xx (except 100) and 2xx responses to those requests. This includes receipt of an UPDATE request on an incoming early dialog to reflect the result of transfer before answer

(semi-attended transfer). A P-Asserted-Identity header field received from the OpenScape Voice should only be used to provide name and/or number display information to the user; it must not be relied upon for asserted identity information.

A P-Asserted-Identity header field received in requests and responses from a client will be ignored by the OpenScape Voice.

5.11.21.2 OpenScape Voice Behavior

OpenScape Voice sends the P-Asserted-Identity header field in all SIP INVITE requests and may send the P-Asserted-Identity header field in other requests and responses above with a SIP or SIPS URI (not TEL URI). This capability may be used to deliver name and/or number display information to the user.

5.11.22 Privacy (RFC 3323)

5.11.22.1 Client Behavior

OpenScape Voice ignores a Privacy header field received in INVITE requests and responses from clients.

5.11.22.2 OpenScape Voice Behavior

OpenScape Voice will not send a Privacy header field to a client.

5.11.23 Proxy-Authenticate

Clients must support receipt of a Proxy-Authenticate header field in a 407 response.

Note: This is in support of working with proxies. There is no requirement on OpenScape Voice to send this.

5.11.24 Proxy-Authorization

Clients must support sending of a Proxy-Authorization header field in a request in accordance with RFC 3261.

Note: This is in support of working with proxies. There is no requirement on OpenScape Voice to receive this.

5.11.25 Proxy-Require

Servers and clients should NOT send this header field. There is no requirement for OpenScape Voice to act upon this header field if received, even when acting as a proxy for the request.

5.11.26 Reason (RFC 3326)

Clients must support receipt of the Reason header field in a BYE request and in a CANCEL request and, depending on the particular reason conveyed, should provide an appropriate indication to the user. OpenScape Voice may send the Reason header field in a BYE request or in a CANCEL request. For the particular case where the OpenScape Voice wishes to indicate in a CANCEL request that the call concerned has been answered elsewhere, it will include a Reason header field containing SIP response code 200 and Reason-Phrase "Call completed elsewhere".

5.11.27 Record-Route

Clients and servers must support receipt of a Record-Route header field in a dialog-forming request or response as specified in RFC 3261 for a UAC and for a UAS. Clients and servers must support sending a Record-Route header field in a response to a dialog-forming request based on what is received in the request.

Note: This header field can occur as a result of an edge proxy.

Note: The requirements of Sect. 5.2.1 concerning transport=tls apply when a server generates a URI for use in this header field.

5.11.28 Refer-To (RFC 3515)

Clients and servers that support the REFER method (see Sect. 5.9.7) must support the Refer-To header field accordingly.

5.11.29 Referred-By (RFC 3892)

Note: This is needed for various forms of call transfer and for conference.

5.11.29.1 Client Behavior

A client must support the sending and receipt of a Referred-By header field in a REFER request and the sending of a Referred By header field in an INVITE request arising from a REFER request. A client may support receipt of a Referred-By header field in an INVITE request.

5.11.29.2 OpenScape Voice Behavior

OpenScape Voice supports the receipt of a Referred-By header field in a REFER request. A Referred By header field may be included within an INVITE or re-INVITE request sent to a transfer-target if the message is received from a trusted interface or is being sent as a result of an OpenScape Voice Single Step or Blind Call Transfer.

Note: Sending of a Referred-By header to an insecure interface is controlled by an OpenScape Voice attribute that is provisioned for these interfaces.

5.11.30 Replaces (RFC 3891)

Note: This is needed for attended and semi-attended call transfer.

5.11.30.1 Client Behavior

A client must support the sending and receipt of a Replaces header field as a parameter of the URI in a Referred-To header field in a REFER request. A client must support sending and receipt of a Replaces header field in an INVITE request.

Note: Sending in an INVITE request can arise from receipt of a REFER.

5.11.30.2 OpenScape Voice Behavior

OpenScape Voice supports the receipt of a Replaces header field as a parameter of the URI in a Referred-To header field in a REFER request. For the capabilities described in this specification OpenScape Voice does not send a Replaces header field in an INVITE request.

5.11.31 Require

Servers and clients must support receipt of a Require header field in a request in accordance with RFC 3261.

In the case of a server, handling of a received Require header field will depend not only on whether the options concerned are supported by the server, but also whether they are recognised as options where support needs to be provided by the downstream UAS, e.g., options relating to SDP offer/answer. The following cases apply:

- If the server recognises an option tag as an option that the server is able to provide in the context of this request, the server must NOT return a 420 response code on account of this option tag and should NOT pass this option tag on if the request is forwarded (unless the same option is to be required of the downstream UAS).
- If the server recognises an option tag as an option that, in the context of this request, should be provided by the downstream UAS, the server must forward the option tag in a Require header field if the request is passed on.
- If the server recognises an option tag as an option that, in the context of this request, neither the server nor the downstream UAS is able to provide, or if the server does not recognise an option tag, the server must indicate this option tag in an Unsupported header field in a 420 response and must NOT pass the request on.

Servers must be able to send a Require header field in a request and clients may be able to send a Require header field in a request.

Note: This requirement on servers reflects requirement 2 above concerning forwarding an option tag in a Require header field of a forwarded request. servers and clients should avoid unnecessary use of the Require header field in other situations, because of its detrimental impact on performance if a modified request has to be submitted on receipt of a 420 response.

5.11.32 Request-Disposition

OpenScape Voice does not provide any support for the Request-Disposition header field.

5.11.33 Retry-After

5.11.33.1 Client Behavior

A client must observe the contents of the Retry-After header field in a 500 or 503 response to a REGISTER or SUBSCRIBE request by deferring a repetition of the request for at least the number of seconds indicated.

5.11.33.2 OpenScape Voice Behavior

OpenScape Voice may include the Retry-After header field when sending a 500 or 503 response to a REGISTER or SUBSCRIBE request.

5.11.34 Route

Clients and servers must support sending a Route header field in mid-dialog requests as specified in RFC 3261 for a UAC.

Note: The need to send this header field can arise as a result of receiving a Record-Route header field from an edge proxy.

5.11.35 Server

5.11.35.1 Client Behavior

A client that supports TLS must take account of the relevant tokens (see Appendix A) if present in this header field in a 2xx response to a REGISTER request.

5.11.35.2 Server Behavior

If a server sends this header field in a response, it must encode it in accordance with Appendix B.

When sending a 2xx response to a REGISTER request, a server must include this header field if support for TLS connectivity checking is applicable and must include the relevant token(s) in accordance with Appendix A.

5.11.36 Session-Expires (RFC 4028)

Clients and servers must support sending and receipt of the Session-Expires header field.

5.11.37 Subscription-State (RFC 3265)

Clients and servers must support sending and receipt of the Subscription-State header field in accordance with RFC 3265 and with any event packages that are supported.

5.11.38 Supported

Clients and servers must support sending of the Supported header field in relevant requests and 2xx responses when able to support specific options within the context of that request or dialog.

Clients and servers must support receipt of the Supported header field in requests and 2xx responses.

5.11.39 To

Clients and servers must support sending and receipt of the To header field in accordance with RFC 3261.

5.11.40 Unsupported

Clients and servers must be able to include an Unsupported header field when sending a 420 response, in accordance with RFC 3261. Servers and clients that send a Require header field in a request must be able to receive an Unsupported header field in a response.

5.11.41 User-Agent

5.11.41.1 Client Behavior

If a client sends this header field in a request, it should encode it in accordance with Appendix B, “ABNF Definition of SIP Headers Server and User-Agent”.

5.11.41.2 OpenScape Voice Behavior

For the capabilities described in this document, there are no OpenScape Voice requirements for this header field.

5.11.42 Via

Clients and servers must support the Via header field in accordance with RFC 3261.

5.11.43 Warning

5.11.43.1 Client Behavior

For the capabilities described in this document, there are no client requirements for this header field.

5.11.43.2 OpenScape Voice Behavior

OpenScape Voice may include a Warning: header with a warning code of 399 in some SIP response messages. The intent of the Warning header is to facilitate the debugging of complex scenarios; there is no need for any device to take any special action based on this header field e.g. Warning: 399 <OpenScape Voice IP> "No License"

5.11.44 WWW-Authenticate

5.11.44.1 Client Behavior

Clients must support receipt of a WWW-Authenticate header field in a 401 response.

5.11.44.2 OpenScape Voice Behavior

OpenScape Voice supports sending a WWW-Authenticate header field in a 401 response.

5.11.45 X-Siemens-Call-Type

See Sect. 5.14.1.

OpenScape Voice may send this header field with value 'recall-transfer' and clients may support receipt of this header field in accordance with Appendix E.

Clients and servers that support media security (see Sect. 8.15) must support sending and receipt of this header field with values 'ST-secure' and 'ST-insecure'.

Clients and servers that support directed call Pick-UP (see Sect. 8.5) must support sending and receipt of this header field with value 'DIR_PICK'.

Clients that support group call Pick-UP (see Sect. 8.6) must support receipt of this header field with value 'GROUP_PICK'. Servers that support group call Pick-UP must support sending this header field with value 'GROUP_PICK'.

Clients that support endpoint controlled recording as a recording client (see Sect. 8.16) must support sending of this header field with value 'recording' and MAY support sending this header field with value 'recorded' . Servers that support endpoint controlled recording must support receipt of these values.

Clients that support endpoint controlled recording or server controlled recording (see Sect. 8.16.1 and Sect. 8.16.1.3) must support receipt of this header field with value 'recorded'. Servers that support one or both of these features must support sending this value.

Clients and servers that support 1-way speaker call must support sending and receipt of this header field with value 'speaker-1-way'. Clients that support this 1-way speaker call must auto-answer and carry out special procedures for this feature when this value of this header field is received in an INVITE request.

Clients and servers that support 2-way speaker call must support sending and receipt of this header field with value 'speaker-2-way'. Clients that support this 2-way speaker call must auto-answer and carry out special procedures for this feature when this value of this header field is received in an INVITE request.

Note: Special client procedures for 1-way/2-way speaker call involve initially playing a warning tone to both parties and then periodically playing a reminder tone to both parties.

5.11.46 X-Siemens-CDR

See Sect. 5.14.2.

OpenScape Voice may be configured to accept billing information received in this SIP header field of a SIP INVITE or REFER request and to store such information in the CDR.

Clients may use this header field, if received in an INVITE request, to correlate the incoming call with some other entity (e.g., a CSTA call). In order to receive it, clients must include the x-siemens-cdr option tag when sending a REGISTER request.

5.11.47 X-Siemens-Proxy-State

OpenScape Voice supports sending this header field and clients may support receipt of this header field in accordance with OpenScape Voice survivability procedures. These procedures are described in other Siemens specifications that are available upon request, but are outside the scope of this document.

5.11.48 X-Siemens-RTP-stats

See Appendix J, “Quality of Service (QoS) Monitoring”.

Clients may include this header field to report session quality statistics for audio when sending a BYE request or a 200 response to a BYE request under both of the following conditions:

- Immediately prior to sending the message the call had in a session that includes audio.
- The session duration was not less than the Minimal Session Length specified in Appendix J.

In this situation, the report period comprises the entire session duration, either from the start of the call or from the most recent session change (e.g., remote IP address change, codec change), whichever is later, and relates to the sole or first audio medium in the SDP. The client must include in the header field one instance of each of the parameters specified in Appendix J for this header field, but omitting any parameters for which information is not available.

Notes:

- The provision of statistics for video is outside the scope of this version of this specification.
- The provision of statistics when a session changes during a call (e.g., owing to call transfer by the remote party) is outside the scope of this version of this specification. The main purpose is to provide statistics in the event that a user terminates the call because of bad audio quality, and this will generally result in sending or receipt of a BYE request.
- Where two SDP m-lines are used to represent alternatives for the same medium (e.g., with and without SRTP), the first audio medium might not correspond to the first m=audio line.

OpenScape Voice supports receipt of this header field in a BYE request or a 200 response to a BYE request and interprets the contents as relating to the sole or first audio medium in the SDP. If more than one of instance of this header field is present, OpenScape Voice will use information only from the first one.

5.11.49 X-Siemens-Original-Called-Identity

For Group Call Pick-UP or Directed Call Pick-UP, OpenScape Voice sends an X-Siemens-Original-Called-Identity header field to the picking-up UA containing the identity of the party that is being picked up.

It is supported for the SIP Subscriber (keyset and non-keyset) interface only. If the identity of the party that is picking up the call has its presentation restricted, the X-Siemens-Original-Called-Identity header field is anonymized.

5.11.50 X-Siemens-IID

The purpose of this header field is the conveyance of a network interface identifier (e.g. MAC address) in the course of client registration. This information helps in identifying a device in the network and can assist in locating the device, e.g. for the purpose of emergency calls.

A client that can determine its network interface identifier(s) must support sending this header field in every REGISTER request it sends to OpenScape Voice.

OpenScape Voice supports receipt of this header field in REGISTER requests.

The header field is defined in Section 5.14.7, “X-Siemens-IID”.

5.12 SIP Response Codes

The response codes discussed below are all defined in RFC 3261 unless another reference is given.

There is no requirement to support any of the response codes not listed here. This means that:

- There is no requirement for a client to send such a response code.
- There is no requirement for OpenScape Voice to send such a response code to a client.
- There is no requirement for a client or OpenScape Voice to take special action on receipt of such a response code, other than the default behaviour for unsupported response codes in a given range (see Sect. 5).

Similarly for a response code for which only sending requirements are stated for a client or OpenScape Voice, there are no special requirements on receipt other than the default behaviour for the range concerned. For a response code for which only receiving requirements are stated for a client or OpenScape Voice there is no requirement to send.

5.12.1 SIP 1xx Response Codes

5.12.1.1 100 Trying

Clients and OpenScape Voice must support sending and receipt of this.

5.12.1.2 180 Ringing

Clients and OpenScape Voice must support sending and receipt of this. Clients should use receipt of this to provide appropriate audio and visual indication to the user that the called user is being alerted.

5.12.1.3 181 Call is Being Forwarded

Clients must support receipt of this and OpenScape Voice must support sending this in the context of call diversion (see Sect. 8.8.1).

5.12.1.4 182 Queued

Clients must support receipt of this and should provide appropriate audio and visual indication to the user that the call is being queued, including the identity of the queued-at party if available. OpenScape Voice will only send this response if received from a peer UA.

5.12.1.5 183 Session Progress

Clients must support receipt of this. OpenScape Voice supports sending and receipt of this.

5.12.2 SIP 2xx Response Codes

5.12.2.1 200 OK

Clients and OpenScape Voice must support sending and receipt of this.

5.12.2.2 202 Accepted

Clients must support receipt of this in a response to a REFER request and clients and OpenScape Voice must support receipt of this in response to a SUBSCRIBE request.

Clients and OpenScape Voice must support sending this in a response to a REFER request. Clients may support sending this in response to a SUBSCRIBE request. OpenScape Voice does not currently support sending this in response to a SUBSCRIBE request.

5.12.3 SIP 3xx Response Codes

Clients need not take action on contacts other than the first contact in a received 3xx response.

5.12.3.1 302 Moved Temporarily

Clients and OpenScape Voice must support receipt of this. OpenScape Voice supports sending this if the diversion is to be performed by the client rather than OpenScape Voice. Clients that support client-based diversion must support sending this.

5.12.4 SIP 4xx Response Codes

5.12.4.1 400 Bad Request

Clients and OpenScape Voice must support sending this.

5.12.4.2 401 Unauthorized

Clients must support receipt of this. OpenScape Voice must support sending this.

5.12.4.3 402 Payment Required

OpenScape Voice will only send this response if received from a peer UA.

5.12.4.4 403 Forbidden

Clients should support receipt of this by providing an appropriate indication to the user. OpenScape Voice supports sending this.

5.12.4.5 404 Not Found

Clients should support receipt of this by providing an appropriate indication to the user. Clients must support sending this. OpenScape Voice will only send this response if received from a peer UA.

Note: Clients should send this when a received Request-URI fails to match a registered contact URI.

5.12.4.6 405 Method Not Allowed

Clients and OpenScape Voice must support sending this.

5.12.4.7 406 Not Acceptable

Clients and OpenScape Voice must support sending this.

5.12.4.8 407 Proxy Authentication Required

Clients may support receipt of this and resending the request with appropriate authentication. OpenScape Voice does not currently send this response.

5.12.4.9 408 Request Timeout

OpenScape Voice supports sending this.

5.12.4.10 415 Unsupported Media Type

Clients and OpenScape Voice must support sending this.

5.12.4.11 416 Unsupported URI Scheme

Clients must support sending this. OpenScape Voice does not currently send this response, if received URI scheme is not SIP, SIPS, or TEL OpenScape Voice will return 400 response.

5.12.4.12 420 Bad Extension

Clients and OpenScape Voice must support sending this.

5.12.4.13 422 Session Interval Too Small (RFC 4028)

Client and OpenScape Voice must support receipt of this.

5.12.4.14 480 Temporarily Unavailable

Clients may support sending this to indicate a DND condition and in this case must include the value "Do Not Disturb" in the reason phrase of a Reason header. OpenScape Voice makes use of "Do Not Disturb" in the reason phrase to determining future handling of the call.

5.12.4.15 481 Call/Transaction Does Not Exist

Clients and OpenScape Voice must support sending this.

Note: This can also arise because of failure to match a dialog with the contents of the Replaces header field. For example, during call-pick-up this can mean that the target call is no longer available to be picked up.

Note: If a Notify is received by the OpenScape Voice that has no subscription associated with it, a "481 Call/Transaction Not Found" response will be sent back to the sender. When a client receives a 481 to a Notify, the client should reSubscribe to the event.

5.12.4.16 484 Address Incomplete

Clients should support receipt of this by providing an appropriate indication to the user. OpenScape Voice will only send this response if received from a peer UA.

5.12.4.17 486 Busy Here

Clients and OpenScape Voice must support sending this. Clients should support receipt of this by providing an appropriate indication to the user and enabling appropriate features. OpenScape Voice supports receipt of this and may invoke appropriate features.

5.12.4.18 487 Request Terminated

Clients and OpenScape Voice must support sending this.

5.12.4.19 488 Not Acceptable Here

Clients and OpenScape Voice must support sending this.

5.12.4.20 489 Bad Event (RFC 3265)

Clients and OpenScape Voice must support sending this.

5.12.4.21 491 Request Pending

Clients and OpenScape Voice must support sending this.

5.12.5 SIP 5xx Response Codes

5.12.5.1 500 Server Internal Error

Clients must support receiving this and must honour any Retry-After header [Sect. 5.11.33] included. OpenScape Voice supports sending this.

5.12.5.2 501 Not Implemented

Clients must support sending this. OpenScape Voice will only send this response if received from a peer UA.

5.12.5.3 503 Service Unavailable

Clients must support receiving this and must honour any Retry-After header [Sect. 5.11.33] included. OpenScape Voice supports sending this.

5.12.5.4 504 Server Time-out

Clients must support receiving this and OpenScape Voice supports sending this.

5.12.5.5 505 Version Not Supported

Clients must support sending this. OpenScape Voice will only send this response if received from a peer UA.

5.12.6 SIP 6xx Response Codes

5.12.6.1 600 Busy Everywhere

A client should *not* send this. A client should treat this the same as 486 if received.

OpenScape Voice will only send this response if received from a peer UA.

5.12.6.2 603 Decline

A client must send this to indicate that a call has been rejected by the user. Clients must support receiving this. OpenScape Voice may send this—e.g., if Resource Management determines that requested bandwidth is not available/authorized.

5.12.6.3 604 Does Not Exist Anywhere

A client should *not* send this. OpenScape Voice will only send this response if received from a peer UA.

5.12.6.4 606 Not Acceptable

A client should *not* send this. OpenScape Voice may return this response code to indicate that the call has been blocked by call admission control (bandwidth limitation). OpenScape Voice may also use this code in a Reason header of a BYE message to indicate that a call has been blocked by call admission control.

5.12.7 Client Handling of 4xx, 5xx and 6xx Responses to an INVITE

In the event of a client receiving an irrecoverable response to an INVITE request, indicating that a session could not be established, the client will normally provide some indication to the user (e.g. tone, display, context menu etc.). The following is intended to provide guidance so that clients conforming to this specification can provide a consistent response to the user for a given response from a server.

For this purpose the following categories of irrecoverable response are specified.

- Busy: The result of calling a user who is either busy or in a state of unwillingness or inability to accept a call (e.g. DND).
- Temporarily Unavailable: The result of attempting to access a feature or network resource that is "temporarily" not available (e.g., due to network congestion).
- Permanently Unavailable: The result of attempting to access a feature, subscriber or destination that is not obtainable unless configuration of the environment is changed to allow access (e.g., no such number, not in service, not authorized, etc.).
- Terminated: The result of terminating a request by normal means (e.g., as a result of the client sending a CANCEL request).

An irrecoverable response is any 4xx, 5xx or 6xx response to an INVITE request except those for which recovery is possible by immediately resubmitting a modified INVITE request (e.g., with credentials for digest authentication, if the client has been challenged).

Table 5.2 lists those irrecoverable SIP response codes that fall into the busy or temporarily unavailable category. All other irrecoverable 4xx, 5xx or 6xx response codes fall into the permanently unavailable category.

Response Code	Category	Additional Requirements
408 Request Timeout	Temporarily Unavailable	—
480 Temporarily Unavailable	Busy	See Sect. 5.12.4.14.
486 Busy Here	Busy	
503 Service Unavailable	Temporarily Unavailable	
606 Not Acceptable	Temporarily Unavailable	See Sect. 5.12.6.4.
487 Request Terminated	Terminated	

Table 5.2 Irrecoverable SIP Response Codes—Busy or Temporarily Unavailable Categories

---

Note: 606 Not Acceptable is in a different category from that of 488 Not Acceptable Here, although both relate to an unacceptable SDP offer. The reason for the different categories is that 606 is used by OpenScape Voice when a call is rejected by call admission control because of temporary conditions.

On receipt of an irrecoverable response, a client SHOULD provide indications to the user in accordance with the category to which that response belongs. For example, the choice of tone should at least indicate the category, even if the range of tones available is insufficient to indicate the precise reason for call failure. A display may be able to provide a more precise reason. In the case of the Terminated category, the client SHOULD NOT provide a tone, since the user should already be aware of the reason for termination (e.g., because the user initiated call clearing).

5.12.8 Response Codes When OpenScape Voice Provides Tones and Announcements

When OpenScape Voice provides a tone or announcement (from a media server) rather then immediately returning a failure response, it MAY generate a 183 provisional response at the start of the tone or announcement and include an SDP answer. Following completion of the tone or announcement, OpenScape Voice MAY generate a final response that is consistent with the failure condition.

Note: For example, in the case of DND a 480 response may be generated in accordance with Sect. 5.12.4.14.

5.13 SIP Option Tags

Clients and OpenScape Voice must support (send when appropriate and recognise on receipt) the following option tags:

- replaces RFC 3891
- timer RFC 4028
- x-oscar-opu (only if Group Pick-UP is supported in accordance with Appendix F, in which case usage must be in accordance with Appendix F)

The Supported header field is extended with a new token value.

The x-siemens-cdr option tag is case-sensitive and is defined as:

x-siemens-cdr

Example:

Supported: x-oscar-opu

The x-siemens-cdr option tag means that the sending UA supports receipt of the X-Siemens-CDR header field.

For the basic capabilities described in this document there is no requirement to support any option tags not listed above. This means that:

- There is no requirement for a client to send such an option tag in a Require, Proxy-Require or Supported header field.
- There is no requirement for a server to send such an option tag to a client in a Require, Proxy-Require or Supported header field, although it may pass such option tags on when acting as a proxy.

- Other than the default behaviour for unsupported option tags received in Require and Proxy-Require header fields, there is no requirement for a client or server to take special action on receipt of such option tags.

5.14 Nonstandard Data

5.14.1 X-Siemens-Call-Type

This proprietary header is used to convey the call type information unless a feature specification defines another protocol element for its specific call type usage.

Call types must be encoded using the formal definition given below. There can be a list of values if more than one call type applies to the present call. The sequence of values is not relevant.

X-Siemens-Call-type = "X-Siemens-Call-Type" HCOLON call-type-list

call-type-list = call-type *(COMMA call-type)

call-type = token

The call type values *must* be treated as case insensitive.

Entities receiving an unrecognized or unsupported call type *must* ignore it.

Values currently used within this document:

call-type = "DIR_PICK" / token

call-type = "GROUP_PICK" / token

call-type = "recall-transfer" / token

call-type = "recorded" / token

call-type = "recording" / token

call-type = "ST-secure" / token

call-type = "ST-insecure" / token

Other call types are used with Keyset procedures but these are outside the scope of this document.

5.14.2 X-Siemens-CDR

This proprietary header is used to convey call correlation and billing information.

X-Siemens-CDR = "X-Siemens-CDR" HCOLON id-set

id-set = idtoken 0*4(COMMA idtoken)

idtoken = gidgen | gidseq | tidgen | tidseq | chargenum

gidgen = "gid-gn" EQUAL <string>

gidseq = "gid-seq" EQUAL <integer>

tidgen = "tid-gn" EQUAL <string>

tidseq = "tid-seq" EQUAL <integer>

chargenum = "charge" EQUAL <string>

Values currently used within this document:

idtoken = "charge"

5.14.3 X-Siemens-Proxy-State

The header field *X-Siemens-Proxy-State* is included in OPTIONS and REGISTER response is defined by the following ABNF Syntax.

X-Siemens-Proxy-State = ("X-Siemens-Proxy-State") HCOLON xsps-value *(COMMA xsps-value)

xsps-value = "survivable" / "normal" / other-xsps-value

other-xsps-value = token

with token as defined in RFC 3261

5.14.4 X-Siemens-RTP-stats

The proprietary *X-Siemens-RTP-stats-list* header field is included in the SIP BYE message is defined by the following ABNF syntax.

x-siemens-rtp-stats-list = "X-Siemens-RTP-stats-list" COLON x-siemens-rtp-stats-params-lists

x-siemens-rtp-stats-params-lists = x-siemens-rtp-stats-params-list *(COMMA x-siemens-rtp-stats-params-list)

x-siemens-rtp-stats-params-list = mt "SEMI" x-siemens-rtp-stats-param *(SEMI x-siemens-rtp-stats-param)

X-siemens-rtp-stats-param = tb / te / ipl / ... / os / or / generic-param

5.14.5 X-Siemens-Original-Called-Identity

The proprietary X-Original-Called-Identity header field is used much like the *P-Asserted-Identity* header field and follows the same ABNF syntax rules – See RFC 3325

5.14.6 epid parameter

The client shall use the MAC address of the phone as the EPID as this uniquely identifies the device and is persistent over a reboot of the phone.

The format for the epid will be an 8 character ASCI string encoded as follows

1	2	3	4	5	6	7	8
"XX"		<last 6 digits in MAC address>					

where "XX" is a two character string identifying the manufacturer of the device, and "SC" is reserved for use by Siemens devices.

The client shall include the epid in the From: header of every request generated by the phone (e.g. REGISTER, INVITE, SUBSCRIBE etc.) for non-keyset lines and for keyset primary lines, i.e. epid shall not be included for calls from keyset secondary lines.

Example:

From:15615299502<sip:15615299502@10.232.3.110:5061>;tag=0470c66cd6;epid=SCBCC441

5.14.7 X-Siemens-IID

The ABNF definition of this header field extends the message-header production of RFC 3261 as follows:

message-header =/ X-Siemens-IID

X-Siemens-IID = "X-Siemens-IID" HCOLON IID-type *(COMMA IID-type)

IID-type = 802MAC / generic-param

802MAC = “802MAC=”12*12HEXDIG

Note: generic-param as defined in RFC 3261

Examples:

a) X-Siemens-IID: 802MAC=0123456789AB

b) X-Siemens-IID: 802MAC=0123456789AB, extensiontype=coolnewtype

This header field may be sent in every REGISTER request.

6 Requirements for Support of SDP-Related Capabilities

6.1 SDP

Clients and OpenScape Voice must support SDP in accordance with RFC 4566. OpenScape Voice must support audio and video. Clients must support audio and may support video.

6.2 SDP Size

The maximum SDP size supported by OpenScape Voice in V6 is 10 Kbytes. Prior to V6 the maximum size was 3072 bytes.

6.3 Offer-Answer Exchange

Clients and OpenScape Voice must implement the offer-answer model in accordance with RFC 3261 and RFC 3264.

Clients must be tolerant of receiving an SDP offer in which the sequence of media lines differs from that in the last SDP offer or answer sent or received on the dialog concerned, including the removal of media lines and the re-ordering of media lines. Clients must treat missing media lines as indicating that the media concerned are no longer available. Clients must adopt the new sequence of media lines in the subsequent SDP answer.

Note: During certain third-party call control (3PCC) operations (Sect. 8.14) a server can forward an SDP offer or answer from one remote UA and later forward an SDP offer from a different remote UA. The SDP offer from the second remote UA will not necessarily maintain the same sequence of media lines as the previous SDP offer or answer sent to the client. If the server forwards this second SDP offer to the client without re-arranging the media lines, it will be in violation of RFC 3264, which requires a subsequent SDP to have the same media lines in

the same order as previously (although additional media lines may be added). OpenScape Voice may not remedy this and can be expected to deliver SDP offers that fail to comply in this respect.

Note: OpenScape Voice does not currently fully support multiple media streams within a dialog (except for the special case of media security Sect. 6.6). It is recommended that clients should not include multiple media lines in SDP offers except for the special case of media security.

6.4 Codec Change "On the Fly"

Clients and OpenScape Voice are not required to support codec change "on the fly". If an offer-answer exchange has resulted in two or more codecs being agreed for a given medium as a result of the client or server offering two or more, and if the client or server wishes to receive only one of these, the client or server should initiate a further offer-answer exchange to reduce the number of codecs to one. The client or server can either use its own preferred codec (out of those so far negotiated) or can wait to see which codec is used in the first received RTP packet and drop other codecs.

A client or server must transmit using a single codec even when an offer answer exchange has negotiated two or more codecs for a given medium.

Note: The above requirements impact OpenScape Voice only when the server is the peer UA (e.g., when interworking with MGCP).

6.5 Media Connection

At a minimum, clients must support media connection and the playing of ringback tone in accordance with the following requirements. Additional media connection requirements are described in other Siemens specifications that can be supplied upon request.

6.5.1 Port Handling

A UA *must* support symmetric RTP in accordance with RFC 4961.

A UA *must* be able to rotate the RTP receive port for a given medium between calls.

Note: This is less important if SRTP is used to authenticate the sender of an RTP stream, although it is still good practice.

However, a UA should be able to be configured to suppress this behaviour if needed for some firewall or NAT environments.

6.5.2 Validating Received RTP and SRTP Packets

A UA must treat as invalid any received SRTP packet that fails SRTP validation checks.

6.5.2.1 Correlation

A UA must treat as invalid any RTP or SRTP packet received prior to receipt of SDP with a non-zero port for the medium concerned.

A UA may have a means of configuration control over whether it uses IP address and port matching for RTP, i.e., whether it matches the source IP address and port of a received RTP stream with the IP address and port received in SDP. If there is no configuration control, a UA must always use IP address and port matching for RTP. When IP address and port matching is used for RTP, a UA may also use IP address and port matching for SRTP.

Note: IP address and port matching for SRTP is less useful, since SRTP authentication provides a superior means of selecting only valid packets.

6.5.2.2 Correlation When Using IP Address and Port Matching

When using IP address and port matching, a UA must treat as invalid any received RTP/SRTP packet whose source IP address and/or port fails to match corresponding information in received SDP.

Note: In some circumstances a UA may receive SDP containing a dummy port, i.e., a port that will not emit RTP/SRTP and will discard any received RTP/SRTP. The UA will not be aware that this is a dummy port. In such circumstances RTP/SRTP should not be received from that port, and therefore valid RTP/SRTP should not be detected.

A change of SDP from the peer UA can occur (e.g., in a re-INVITE during a confirmed dialog). Although in violation of RFC 3264, a change of SDP can also occur in a subsequent response to an INVITE request on the same early dialog, in which case a UA must treat this as a change of SDP. If the new SDP has a different IP address or port, the UA must treat as invalid any further RTP/SRTP packets received from the old IP address and port.

6.5.2.3 Correlation When Not Using IP Address and Port Matching

When not using IP address and port matching, a UA must treat as invalid any received RTP (not SRTP) packet with a different source IP address and/or port from that of the most recently received valid packet (if any) for the medium concerned, unless a time of at least 3 seconds has elapsed since the last valid packet was received. A UA may make this time a configurable value.

Note: This prevents unwanted switching between two or more sources of received RTP packets, unless one source stops transmitting, in which case another valid source can take over. In particular, in the case of two or more called users simultaneously answering a forked call, this gives an opportunity to switch to the correct stream if the wrong stream was selected to start with. The elapsed time specified above is a compromise between the need to switch quickly when a stream stops (in order to rectify a wrong choice of stream) and the need to avoid switching from the correct stream when there is a natural period of silence.

6.5.3 Rendering Valid RTP Packets

A UA must render received RTP/SRTP packets that are considered valid (see Sect. 6.5.2), except where prevented by the rules for provision of local ringback tone (see Sect. 6.5.4) or local conditions. Local conditions can include the state of the user interface (e.g., whether a call is being held, whether the speaker is muted, whether the video window is obscured), state of the TDM network in the case of a gateway, state of a conference in the case of a conference bridge, etc..

A UA must discard (without rendering to the user) any invalid received RTP or SRTP packets.

6.5.4 Provision of Local Ringback Tone

The following requirements apply only to audio. The possible application of similar procedures to video is outside the scope of this document.

A UA must start local ringback tone (if not already started) on receipt of a 180 response (with or without SDP). If the UA is already rendering RTP/SRTP packets when a 180 response is received, it must stop rendering RTP/SRTP packets. A UA must not start local ringback tone on receipt of a 18x response other than 180.

A UA must continue providing local ringback tone if any 18x response (with or without SDP) is received while local ringback tone is being played.

Note: If valid RTP/SRTP is subsequently received this will cause local ringback tone to be stopped, in accordance with the next paragraph.

A UA must stop providing local ringback tone and render received RTP/SRTP packets in accordance with Sect. 6.5.3 as soon as a valid RTP or SRTP packet (see Sect. 6.5.2) is received. A UA must not restart local ringback tone if the received RTP/SRTP stream appears to stop.

The table below shows how the above rules apply in different situations. In the event of a discrepancy between the table and the rules above, the rules above apply.

State	Event	Action
Not rendering	180 received	Start local ringback tone
Not rendering	18x (not 180) received	No change
Not rendering	Valid RTP/SRTP received (SDP must have been received)	Start rendering RTP/SRTP
Playing local ringback tone	180 received	No change
Playing local ringback tone	18x (not 180) received	No change
Playing local ringback tone	Valid RTP/SRTP received (SDP must have been received)	Stop local ringback tone / start rendering RTP/SRTP
Rendering RTP/SRTP	180 received	Stop rendering RTP/SRTP / start local ringback tone
Rendering RTP/SRTP	18x (not 180) received	No change

6.6 Negotiation of Media Security

Clients and servers that support media security (see Sect. 8.15) *must* support negotiation of the use of the Secure Real Time Protocol (SRTP) for media security, with fallback to RTP, in accordance with Siemens Security and Payload Encryption specifications (that can be supplied upon request). In the case of a server that does not handle media, this requirement applies only to the extent of not hindering negotiation between UAs that do handle media.

As a minimum requirement to ensure interoperability, clients must support the following procedure:

If multiple media lines are received indicating the same codec or set of codecs but different profiles, i.e. RTP/AVP and RTP/SAVP, the client should interpret these as proposed alternatives. The SIP UA SHALL accept only one of the media lines and reject the others by indicating port 0 in its SDP answer.

6.7 Key Management for Media Security

Clients and servers that support media security (see Sect. 8.15) *must* support key management for SRTP in accordance with “Siemens Security and Payload Encryption” specifications (that can be supplied upon request) using either MIKEY (RFC 3830) or SDES (RFC 4568). Specifications can be provided upon request.

If MIKEY is supported, it must be done using unprotected key distribution (referred to in Siemens specifications as MIKEY option 0) and using Key Management Extensions to SDP (RFC 4567) for transporting MIKEY initiator and responder messages in SDP offers and answers respectively. In the case of a server that does not handle media, this requirement applies only to the extent of not hindering key management between UAs that do handle media.

Clients supporting Security Descriptions key management in accordance with “Siemens Security and Payload Encryption” specifications using SDES MUST be capable of being configured to use either MIKEY or SDES. When configured to use MIKEY, a client MUST NOT propose SDES in an SDP offer and MUST NOT accept SDES in a received SDP offer. When configured to use SDES, a client MUST NOT include MIKEY in an SDP offer and MUST NOT accept MIKEY in a received SDP offer.

OpenScape Voice provides interworking between clients that support best-effort SRTP (1 RTP + 1 SRTP media in the offer) and clients that do not support it, e.g. would reject the above offer. Interworking is provided by removing the SRTP media from the offer, hence offering only RTP media to the called party. If one client supports best-effort SRTP and the other supports SRTP-only, but does not accept the offer, there is no interworking and the call shall fail. OpenScape Voice provides interworking between a best-effort SRTP client and an RTP-only client.

7 Requirements for Support of SIP Event Packages

Except where otherwise stated below or in Section 5.11.4, “Allow-Events (RFC 3265)”, there is no requirement to use the Allow-Events header field.

7.1 server-mode-backup and server-mode-normal event packages

These event packages are used by a survivable proxy (refer to the *OpenScape Voice Configuration Manual: Volume 1, System Configuration and Administration*) for notifying a client that the proxy is operating in back-up or normal mode. These event packages use implicit subscriptions (a client is always considered to be subscribed) and NOTIFY requests are sent outside the context of any dialog. A NOTIFY request indicating one of these event packages in the Event header field does not contain a body.

Clients must support receipt of NOTIFY requests indicating these events. The procedures for use of these events is described in the Siemens Survivability specification (available upon request) and is outside the scope of this document.

Note: At present no security mechanism is defined for these NOTIFY requests to prevent denial of service attacks.

7.2 refer event package (RFC 3515)

Clients must support this event package as subscribers and notifiers. OpenScape Voice supports this event package as a notifier.

Note: This event package uses bodies of type 'message/sipfrag'.

This event package uses implicit subscriptions. A subscription begins when a 2xx response to a REFER request is sent and the subscription shares the dialog on which the REFER request was sent. The subscription implicitly ends when a NOTIFY request containing a final response to the referred request has been sent and responded to or when the subscription expires. A client or server acting as notifier should choose a value for the expires parameter of the Subscription-State header field that is long enough to avoid expiry in most reasonable circumstances. There is no requirement to support subscription refresh.

Subscriptions will normally be lost on switching to survivability mode.

Note: OpenScape Voice currently only supports implicit subscription to the 'refer' event when the REFER request is used to add a participant to a Station Controlled Large Conference. OpenScape Voice currently does not send NOTIFY messages to report the progress of the REFER request as part of normal (i.e. not related to Station Controlled Large Conference) call transfer procedures. OpenScape Voice support for these capabilities is planned for release V5.0. and this document will be updated at that time to describe the procedures.

7.3 message-summary event package (RFC 3842)

Clients must support this event package if they support the MWI feature. OpenScape Voice supports this event package (Sect. 8.10.)

Note: This event package uses bodies of type 'application/simple-message-summary'.

This event package uses implicit subscriptions (a client is always considered to be subscribed) and NOTIFY requests are sent outside the context of any dialog.

Clients and servers that support this event package must use the body type and format specified in RFC 3842.

OpenScape Voice includes the msg-status-line (indicating whether or not there are messages waiting), and does not currently include a Voice-Message msg-summary-line.

A client must understand the msg-status-line and the Voice-Message msg-summary-line.

7.4 dialog event package (RFC 4235)

Clients should support this event package as subscribers and OpenScape Voice will support this event package as notifier for the Group Pick-Up feature (Sect. 8.6).

Note: This event package uses bodies of type 'application/dialog-info+xml'.

The procedures for use of these events when a backup server is utilized is described in the Siemens Survivability specification (available upon request) and is outside the scope of this document.

7.5 conference event package (RFC 4575)

A client MAY support this event package as subscriber in support of centralized conference (Sect. 8.7.1).

A server must support this event package as notifier in support of centralized conference.

A client that supports local conference as host MAY support this event package as notifier.

A server must be transparent to SUBSCRIBE and NOTIFY requests and responses relating to this event package when used in support of local conference.

7.6 talk event package

This event package is used to request a client to auto-answer an incoming call. This event package uses implicit subscriptions. NOTIFY requests are sent outside the context of any dialog. NOTIFY requests indicating this event package in the Event header field do not contain a body.

A client may support this event package as subscriber.

A client that supports this event package as subscriber must include an Allow-Events header field indicating 'talk' when sending a 180 response to an INVITE request in order to indicate that a call can be auto-answered in this way.

Note: This is non-standard use of the Allow-Events header field, because normally it is sent by a UA capable of being a notifier, rather than a UA capable of receiving a NOTIFY request.

A client that supports this event package as subscriber must support receipt of NOTIFY requests indicating this event in the Event header field and auto-answer a call if one exists in a suitable state. A client that is a keyset must auto-answer only calls alerting the primary line. A client must send a 200 response to a NOTIFY request whether or not it is successful in auto-answering a call.

Note: At present no security mechanism is defined for these NOTIFY requests to prevent denial of service attacks.

A server may support this event package as notifier.

8 SIP Support of Higher-Level Features

8.1 Identification Services

Clients and servers must support SIP User Identification Services as described in Appendix C, “SIP—Identification Services (Display Services)”.

8.2 Call Hold

For the purposes of this specification, call hold is a means by which a user can suspend the bidirectional flow of media during a call (hold) in order to perform some other action (e.g., participate in a different call) and can subsequently restore the flow of media (retrieval). Normally the remote participant is made aware of having been placed on hold (e.g., by means of a displayed message and/or the playing of music or an announcement. SIP signaling in support of this is specified in Appendix D, “SIP—Media Hold”. This mechanism can be used for various purposes (e.g., to make a consultation call, swap between lines).

8.2.1 Client Behavior

Clients must support provisions for UAs in Appendix D for holding and retrieving calls as qualified below. This applies to the use of hold for various purposes, in particular consultation hold and manual hold.

When holding and retrieving a call, a client must support procedures for hold and may support the transmission of media on hold. A client that supports transmission of media on hold must NOT do so while known to be in a conference call.

Note: If local media on hold is provided, the client should send a=sendonly, otherwise a=inactive, in the SDP offer.

When a call is being held, a client must have the capability to apply local media on hold (e.g., music) for appropriate media if not provided remotely.

Note: If a=sendonly is received in the SDP offer, media on hold is being provided remotely, either from the hold UA or from a media server introduced by the server. If a=inactive is received, media on hold is not being provided remotely.

8.2.2 OpenScape Voice Behavior

OpenScape Voice supports corresponding procedures in Appendix D, “SIP—Media Hold” on the interface to the holding client and on the interface to the held client. OpenScape Voice has the capability of introducing a media server to provide media on hold if inactive or sendonly is received and must be configurable to do so if inactive or sendonly is received, only if inactive is received, or not at all. The signaling by which a SIP server introduces a media server for this purpose is outside the scope of this specification.

Note: Introduction of a media server can mean that the SDP offer to the held client contains a=sendonly whereas the SDP offer from the holding client contained a=inactive.

8.3 Consultation

Clients can use hold and retrieve for consultation hold, allowing the making of a consultation call, alternating between a consultation call and an existing call, reverting to a held call after clearing another call, etc. No additional signalling at the SIP client interface is involved.

8.4 Call Transfer

Call transfer includes blind (unattended) transfer, attended transfer and semi-attended transfer, as defined in Appendix E, “SIP—Call Transfer”.

8.4.1 Blind (Unattended) Transfer

Clients and servers must support blind transfer in accordance with Appendix E as qualified below.

This feature requires use of the REFER method (see Sect. 5.9.7), the Refer-To header field (see Sect. 5.11.28) and the Referred-by header field (see Sect. 5.11.29).

8.4.2 Interface to a Client Acting as a Transferor

8.4.2.1 Client Behavior

A client must be able to perform the role of transferor in accordance with Appendix E, “SIP—Call Transfer”.

Note: This includes the sending of a REFER request with method INVITE in the Refer-To header field URI and optionally with a Referred-By header field and acting on responses and notifications.

8.4.2.2 OpenScape Voice Behavior

OpenScape Voice will support signaling to the transferor in accordance with Appendix E.

Note: This includes receipt of a REFER request with method INVITE in the Refer-To header field URI and optionally with a Referred-By header field.

8.4.3 Interface to a Client Acting as a Transferee

8.4.3.1 Client Behavior

A client must be able to act as a transferee in accordance with Appendix E, “SIP—Call Transfer” for the case where the server has terminated the REFER request from the transferor.

Note: This just involves processing a received re-INVITE request.

In this case, the client should display the identity of the transfer target if received in a P-Asserted-Identity header field in the re-INVITE request

8.4.3.2 OpenScape Voice Behavior

OpenScape Voice will *not* forward a REFER request from the transferor to the transferee. OpenScape Voice will support signaling to the transferee in accordance with Appendix E, “SIP—Call Transfer” for the case where the server terminates the REFER request.

Note: This just involves sending a re-INVITE request.

A server should include the identity of the transfer target in a P-Asserted-Identity header field in the re-INVITE request.

8.4.4 Interface to a Client Acting as a Transfer Target

8.4.4.1 Client Behavior

A client must be able to perform the role of transfer target in accordance with Appendix E, “SIP—Call Transfer”.

Note: This just involves processing a received INVITE request, which may contain a Referred-By header field.

A client may make use of the Referred-By header field to provide more information to the user.

8.4.4.2 OpenScape Voice Behavior

OpenScape Voice will support signaling to the transfer target in accordance with Appendix E, “SIP—Call Transfer”.

Note: . This involves sending re-INVITE requests and indicating the identity of the transferee in a P-Asserted-Identity header field.

8.4.5 Attended Transfer

Clients and servers must support attended transfer in accordance with Appendix E, “SIP—Call Transfer” as qualified below.

This feature requires use of the REFER method (see Sect. 5.9.7), the Refer-To header field (see Sect. 5.11.28), the Replaces header field (see Sect. 5.11.30) and optionally the Referred-by header field (see Sect. 5.11.29).

8.4.6 Interface to a Client Acting as a Transferor

8.4.6.1 Client Behavior

A client must be able to perform the role of transferor in accordance with Appendix E.

Note: This includes the sending of a REFER request with method INVITE and a Replaces header field parameter in the Refer-To header field URI and acting on responses and notifications.

8.4.6.2 OpenScape Voice Behavior

A server must support signalling to the transferor in accordance with Appendix E.

Note: This includes receipt of a REFER request with method INVITE and a Replaces header field parameter in the Refer-To header field URI.

8.4.7 Interface to a Client Acting as a Transferee

8.4.7.1 Client Behavior

A client must be able to act as a transferee in accordance with Appendix E for the case where the server has terminated the REFER request from the transferor.

Note: This just involves processing a received re-INVITE request.

In this case a client should display the identity of the transfer target if received in a P-Asserted-Identity header field in the re-INVITE request.

8.4.7.2 OpenScape Voice Behavior

OpenScape Voice will NOT forward a REFER request from the transferor to the transferee. A server must support signalling to the transferee in accordance with Appendix E for the case where the server terminates the REFER request.

Note: This just involves sending a re-INVITE request.

A server should include the identity of the transfer target in a P-Asserted-Identity header field in the re-INVITE request.

8.4.8 Interface to a Client Acting as a Transfer Target

8.4.8.1 Client Behavior

A client must be able to act as a transfer target in accordance with Appendix E for the case where the server has terminated the REFER request from the transferor.

Note: This just involves processing a received re-INVITE request.

In this case the client should display the identity of the transferee if received in a P-Asserted-Identity header field in the re-INVITE request.

8.4.8.2 OpenScape Voice Behavior

OpenScape Voice will NOT generate an INVITE request containing a Replaces header field towards the transfer target. A server must support signalling to the transferee in accordance with Appendix E for the case where the server terminates a REFER request from the transferor.

Note: This just involves sending a re-INVITE request.

A server should include the identity of the transferee in a P-Asserted-Identity header field in the re-INVITE request.

8.4.9 Semi-Attended Transfer

Clients and servers must support semi-attended transfer in accordance with Appendix E as qualified below.

This feature requires use of the REFER method (see Sect. 5.9.7), the Refer-To header field (see Sect. 5.11.28), the Replaces header field (see Sect. 5.11.30) and optionally the Referred-by header field (see Sect. 5.11.29).

8.4.10 Interface to a Client Acting as a Transferor

8.4.10.1 Client Behavior

A client must be able to perform the role of transferor in accordance with Appendix E.

Note: This includes the sending of a REFER request with method INVITE and a Replaces header field parameter in the Refer-To header field URI and acting on responses and notifications. This is no different from attended transfer except that it occurs before the transfer target has answered.

8.4.10.2 OpenScape Voice Behavior

A server must support signalling to the transferor in accordance with Appendix E.

Note: This includes receipt of a REFER request with method INVITE and a Replaces header field parameter in the Refer-To header field URI.

8.4.11 Interface to a Client Acting as a Transferee

8.4.11.1 Client Behavior

From a client perspective, semi-attended transfer is not distinguishable from attended transfer. Therefore the requirements of Sect. 8.4.7 apply.

8.4.11.2 OpenScape Voice Behavior

OpenScape Voice will NOT forward a REFER request from the transferor to the transferee. A server must support signalling to the transferee in accordance with Appendix E for the case where the server terminates the REFER request.

Note: . This just involves sending a re-INVITE request, but not until the transfer target has answered.

A server should include the identity of the transfer target in a P-Asserted-Identity header field in the re-INVITE request.

8.4.12 Interface to a Client acting as a Transfer Target

8.4.12.1 Client Behavior

A client must be able to act as a transfer target in accordance with Appendix E for the case where the server has terminated the REFER request from the transferor.

Note: This just involves processing a received UPDATE request before answer and a received re-INVITE request after answer.

A client should display the identity of the transferee if received in a P-Asserted-Identity header field in an UPDATE request or a re-INVITE request.

8.4.12.2 OpenScape Voice Behavior

OpenScape Voice will *not* generate an INVITE request containing a Replaces header field towards the transfer target. A server must support signalling to the transferee in accordance with Appendix E for the case where the server terminates a REFER request from the transferor.

Note: This involves sending a re-INVITE requests and indicating the identity of the transferee in a P-Asserted-Identity header field.

8.5 Directed Call Pick-UP

Clients and servers may support Directed Call Pick-UP in accordance with Appendix I as qualified below. Requirements in the remainder of this sub-section apply only to a client or server that supports this feature.

This feature requires use of the dialog event package RFC 4235, the Replaces header field (see Sect. 5.11.30) and the X-Siemens-Call-Type header field indicating Directed Call Pick-UP (value 'DIR_PICK').

8.5.1 Interface to a Client Acting as a Picking-up User

8.5.1.1 Client Behavior

On request from a user to perform Directed Call Pick-UP, a client *must* act in accordance with Appendix I.

Note: This involves sending a SUBSCRIBE request to subscribe to the dialog event package at the target, i.e., a user at which a call is believed to be alerting. Then, if the resulting NOTIFY request identifies a suitable call, it involves sending an INVITE request containing a Replaces header field to carry out the pick-up. The X-Siemens-Call-Type header field in the SUBSCRIBE and INVITE requests indicates that directed call pick-up is being performed.

Additional requirements related to survivability are provided in the Siemens Survivability specification (available upon request), but are outside the scope of this document.

8.5.1.2 OpenScape Voice Behavior

OpenScape Voice behaves in accordance with Appendix I.

Note: This involves accepting subscriptions to the dialog event package and acting as notifier. Information given in notifications can be filtered to reflect the fact that it is needed only for directed call pick-up purposes. It also involves receiving INVITE requests containing a Replaces header field and an X-Siemens-Call-Type header field indicating directed call pick-up.

8.5.2 Interface to a Client Acting as a Target User

8.5.2.1 Client Behavior

A client must behave in accordance with Appendix I.

Note: This involves receipt of a BYE or CANCEL request containing an X-Siemens-Call-Type header field indicating directed call pick-up and a Reason header field indicating response code 200. The directed call pick-up indication can be used to indicate to the user that the alerting or held call has been picked up. The Reason header field can be used to adjust call records to reflect the fact that the call was answered elsewhere.

Note: There is no requirement to support the optional procedure in Appendix I whereby the target user's client acts as notifier for the dialog event package subscription from the picking-up user's client.

8.5.2.2 OpenScape Voice Behavior

OpenScape Voice behaves in accordance with Appendix I.

Note: This involves sending a BYE or CANCEL request (depending on whether the original INVITE transaction has completed) containing an X-Siemens-Call-Type header field indicating directed call pick-up and a Reason header field containing response code 200 (indicating that the call has been answered elsewhere).

OpenScape Voice does not use the optional procedure in Appendix I whereby a SUBSCRIBE request for the dialog event package is forwarded to the target user's client.

8.5.3 Interface to a Client acting as a Wanted User

8.5.3.1 Client Behavior

A client *must* behave in accordance with Appendix I.

Note: In the case of an alerting call, this just involves receipt of an X-Siemens-Call-Type header field indicating directed call pick-up in the 200 response to the original INVITE request and processing a received re-INVITE request to perform a new offer/answer exchange. In the case of a held call, this just involves processing a received re-INVITE request containing an X-Siemens-Call-Type header field indicating directed call pick-up. In either case the directed call pick-up indication can be used to indicate to the user that the call has been picked up,

---

Note: There is no requirement to support the optional alternative procedure in Appendix I whereby the target user's client receives an INVITE request containing a Replaces header field and an X-Siemens-Call-Type header field indicating directed call pick-up.

8.5.3.2 OpenScape Voice Behavior

OpenScape Voice behaves in accordance with Appendix I.

Note: When picking up an alerting call, this involves sending an X-Siemens-Call-Type header field indicating directed call pick-up in the 200 response to the original INVITE request and, following receipt of the ACK request, sending a re-INVITE request to initiate a new offer/answer exchange. When picking up a held call, this involves sending a re-INVITE request containing an X-Siemens-Call-Type header field indicating directed call pick-up.

OpenScape Voice does not use the optional alternative procedure in Appendix I whereby it forwards an INVITE request containing a Replaces header field and an X-Siemens-Call-Type header field indicating Directed Call Pick-UP from the picking-up user's client to the wanted user's client

8.6 Group Call Pick-UP

Clients and servers may support Group Pick-UP in accordance with Appendix F as qualified below. Requirements in the remainder of this subsection apply only to a client or server that supports this feature.

This feature requires use of the dialog event package and the Group CPU access code in the Request-URI and To headers. (see Sect. 5.11.30).

8.6.1 Interface to a Client Acting as a Pick-UP Group Member

8.6.1.1 Monitoring Pick-UP Groups

Client Behavior

A client must be capable of being configured as a member of a Pick-UP Group and may be capable of being configured as a member of more than one Pick-UP Group. A client configured as a member of a Pick-UP Group must be configured with the Group Pick-UP UR.

For each configured Pick-UP Group, a client must establish and maintain a subscription to the dialog event package of the Pick-UP Group service in accordance with Appendix F and should make use of resulting notifications to make the user aware of the presence of calls eligible for Group Pick-UP.

OpenScape Voice Behavior

OpenScape Voice is capable of being configured for provision of a Pick-UP Group service on behalf of a number of Pick-UP Groups. OpenScape Voice will accept subscriptions from group members to the Group Pick-UP URI to the dialog event package in accordance with Appendix F.

8.6.1.2 Picking Up Calls

Client Behavior

On request from a user to performGroup Pick-UP on a call that has been notified as eligible for Group Pick-UP, a client must issue an INVITE request to the Group Pick-UP URI and act upon responses in accordance with Appendix F.

OpenScape Voice Behavior

On receipt of an INVITE request containing a Request-URI field indicating the Group Pick-UP URI, OpenScape Voice will act in accordance with Appendix F.

8.6.2 Interface to a Client Acting as a Calling (Wanted) User

8.6.2.1 Client Behavior

A client must behave in accordance with Appendix F.

Note: In the case of an alerting call, this just involves receipt of a 200 response to the original INVITE request and processing a received re-INVITE request to perform a new offer/answer exchange.

Note: There is no requirement to support the optional alternative procedure in Appendix F whereby the target user's client receives an INVITE request containing a Replaces header field and an X-Siemens-Call-Type header field indicating directed call pick-up

---

8.6.2.2 OpenScape Voice Behavior

OpenScape Voice behaves in accordance with Appendix F.

Note: When picking up an alerting call, this involves sending a 200 response to the original INVITE request and, following receipt of the ACK request, sending a re-INVITE request to initiate a new offer/answer exchange.

Note: OpenScape Voice does not use the optional alternative procedure in Appendix F whereby it forwards an INVITE request containing a Replaces header field and an X-Siemens-Call-Type header field indicating Directed Call Pick-UP from the picking-up user's client to the wanted user's client.

8.7 Conferencing

8.7.1 Centralized Conference

Clients and servers must support centralized conferences in accordance with Appendix H, “Conferencing”, as qualified below.

This feature requires use of the REFER method (see Sect. 5.9.7) and the Referred-by header field (see Sect. 5.11.29). It also involves use of the conference event package (see Sect. 7.5) for support of conference membership display.

8.7.2 Interface to a Client Acting as a Conference Creator

8.7.2.1 Client Behavior

A client *must* be able to perform the role of conference creator in accordance with Appendix H, “Conferencing”.

Note: This includes the sending of an INVITE request to the conference factory URI and, if successful, sending REFER requests on any existing dialog whose remote party is to become a conference member.

A client *must* be able to be configured with a conference factory DN that can be combined with the address of the OpenScape Voice to compose a conference factory URI to which requests to create a conference can be sent. The conference factory DN forms the user part of the URI and the OpenScape Voice address forms the host part e.g. "xxxxx@yyyyyy" where:

- xxxxx: shall be "Conference Factory DN"
- yyyyyy: shall be an OpenScape Voice IP Address or domain name.

Note: The absence of a configured conference factory DN can be used to determine that a local conference should be invoked instead of a centralized conference.

8.7.2.2 Server Behavior

A server *must* support conference creation in accordance with Appendix H, “Conferencing”

Note: This includes receipt of an INVITE request to the conference factory URI and, if successful, receipt of REFER requests on any existing dialogs referring the remote party to the conference focus.

8.7.3 Interface to a Client That Is a Conference Member (Including Conference Creator)

8.7.3.1 Client Behavior

A client *must* be able to perform the role of conference member in accordance with Appendix H, “Conferencing”.

Note: This includes the ability to receive the isfocus parameter in a Contact URI and the ability to add another member by sending a REFER request to that member's UA asking it to send an INVITE request to the conference focus.

A client that is a conference member (as determined by receipt of the isfocus parameter) may subscribe to the conference event package in accordance with Appendix H, “Conferencing”, and in this case should display conference membership details based on notifications received.

8.7.3.2 Server Behavior

A server *must* support the interface to a conference member.

Note: This includes the ability to send the isfocus parameter in a Contact URI and the ability to add another member by receiving a REFER request asking a member's UA to send an INVITE request to the conference focus.

A server *must* support receipt of subscriptions to the conference event package in accordance with Appendix H and provide membership details as notifications.

8.7.4 Local Conference

Local conference is implemented entirely within the client device, and is therefore outside the scope of this document.

8.8 Call Diversion

Clients and servers must support call diversion in accordance with Appendix G as qualified below.

Clients and servers must support the Diversion header field (see Sect. 5.11.14) and must support as a minimum the following reasons for diversion:

- Call forwarding unconditional
- Call forwarding busy
- Call forwarding no reply
- Call deflection

8.8.1 Interface to Client Acting as a Calling User

8.8.1.1 Client Behavior

A client must support receipt of diversion indications as detailed in Appendix G in response to an INVITE request and should make information available to the user as appropriate.

A client must support receipt of a 3xx response in accordance with RFC 3261 and must support the handling of a Diversion header field in a 3xx response in accordance with Appendix G.

8.8.1.2 OpenScape Voice Behavior

OpenScape Voice will supply diversion indications in response to an INVITE request as detailed in Appendix G.

Note: Indications include the 181 response and its P-Asserted-Identity header field indicating the diverted-from identity. A P-Asserted-Identity header field in a subsequent response to the INVITE request identifies the diverted-to user.

OpenScape Voice may send a 3xx response to a client in accordance with RFC 3261 and if so may include a Diversion header field in the 3xx response in accordance with Appendix G.

8.8.2 Interface to Client Acting as a Diverted User

8.8.2.1 Client Behavior

A client must support receipt of diversion indications as detailed in Appendix G in an INVITE request and should make information available to the user as appropriate.

8.8.2.2 OpenScape Voice Behavior

OpenScape Voice will supply diversion indications in an INVITE request as detailed in Appendix G.

Note: Indications are contained in zero or more Diversion header fields.

8.8.3 Interface to Client Acting as a Served (Diverting) User

8.8.3.1 Client Behavior

A client may support client-based diversion. A client that supports client-based diversion must support the sending of a 302 response to an INVITE request when conditions for diversion apply and inclusion of a Diversion header field as detailed in Appendix G.

8.8.3.2 OpenScape Voice Behavior

OpenScape Voice will support receipt of a 302 response to an INVITE request, with or without a Diversion header field as detailed in Appendix G. In the absence of a Diversion header field OpenScape Voice will assume the reason is call forwarding no reply if a 180 response has been received and call forwarding unconditional otherwise.

8.8.3.3 OpenScape Voice-Based Diversion

OpenScape Voice may act as a Diverting Proxy/B2BUA as described in Appendix G.

8.8.3.4 Client Behavior

No requirements other than those in Sect. 8.8.3.1 above.

8.8.3.5 OpenScape Voice Behavior

As described in Appendix G.

8.9 Do Not Disturb (DND)

DND is a feature whereby a user can indicate that all incoming calls are to be rejected or handled at an alternative destination (e.g., at a different member of a hunting group).

8.9.1 Client-Based DND

8.9.1.1 Client Behavior

A client that rejects an incoming call because DND is active must use the 480 response code in accordance with Sect. 5.12.4.14.

8.9.1.2 OpenScape Voice Behavior

OpenScape Voice may take special action (e.g. divert the call) on receipt of a 480 response indicating DND.

8.9.2 OpenScape Voice-Based DND

OpenScape Voice acting as a Proxy/B2BUA may provide DND services.

Support for OpenScape Voice based diversion with uaCSTA, in which the client acts as the CSTA CF, is outside the scope of the current version of this document.

8.9.2.1 Client Behavior

No requirements.

8.9.2.2 Server Behavior

OpenScape Voice may take special action (e.g. divert the call) when a subscriber is provisioned with the DND service.

8.10 Message Waiting Indication (MWI)

This feature allows a message server (e.g., for voice messages) to notify a user via his/her client that one or more messages are waiting. The means for accessing those messages is outside the scope of this specification.

Clients should support and OpenScape Voice does support MWI.

This feature requires use of the message-summary event package (see Sect. 7.3).

8.10.1 Client Behavior

The following requirements apply to a client that supports MWI.

- A client should indicate support for the message-summary event package in the Allow-Events header field when sending a REGISTER request.
- A client should be able to receive NOTIFY requests indicating the message-summary event package in the Event header field and containing a message summary body, such requests arriving outside the context of a dialog. A client should use the information contained in the body to advise the user of messages waiting.

8.10.2 OpenScape Voice Behavior

OpenScape Voice is able to provide notifications to a client based on the presence of the Allow-Events header field indicating support for the message-summary event package in REGISTER requests.

After initial registration and whenever message summary information changes for a user, OpenScape Voice will send a NOTIFY request outside the context of a dialog to the registered appearance of the user's line, with the Event header field indicating the message-summary event package and with message summary information in the body.

8.11 Call Completion

Call completion to busy subscriber (CCBS) allows a user who has encountered a busy situation while trying to establish a call to request to have the call established later when the applicable resources become free.

Call completion on no reply (CCNR) allows a user who has encountered a no reply situation while trying to establish a call to request to have the call established later when the called user is believed to be present.

The Allow-Events header field in certain responses to INVITE requests can indicate the availability of CCBS and CCNR features at the OpenScape Voice as an aid to reaching the required destination. Other than this, the SIP client interface currently requires no specific support for CCBS and CCNR. If these services are available at the OpenScape Voice, they can be invoked using an access code, which means sending a normal INVITE request in which the Request-URI contains the access code. The access code can be programmed on a feature key, for example, for ease of use. The client need not be aware of the feature being invoked. 3PCC techniques (see Sect. 8.14) are used to initiate a further call when the destination becomes available.

8.11.1 Client Behavior

A client may have the capability to interpret the Allow-Events header field to determine the availability of CCBS and CCNR.

8.11.2 OpenScape Voice Behavior

OpenScape Voice can support CCBS and CCNR, including indication of availability in the Allow-Events header field.

8.12 Call Waiting

Call waiting is a feature that allows an incoming call to wait at a client even though the user is busy on another call at that client. The user is advised that a call is waiting and can answer the waiting call (after holding or clearing the existing call), reject the waiting call or deflect the alerting call.

A client may support call waiting, in which case the procedures below apply. There is no impact on the server.

When a client is busy and receives an INVITE request that in other respects is acceptable, and if configured for call waiting to be applied, the client must send a 180 response. When the existing call is cleared, the client can alert the user on behave of the call that has been waiting and allow the user to answer it. Also the client should allow the user to hold the existing call in order to answer the waiting call. In either case, when the user answers the call the client must send a 200 response to the INVITE request.

8.13 Call offer

Call offer is a feature that allows a caller to request that a call wait at a busy destination. The called user is advised that a call is waiting and can answer the waiting call (after holding or clearing the existing call), reject the waiting call or deflect the waiting call.

Clients MAY support call offer as a calling client and MAY support call offer as a called client. OpenScape Voice MAY support call offer for a calling client and MAY support call offer for a called client.

This feature requires use of the Call-Info header field (see Sect. 5.11.8) and the Request-Disposition header field (see Sect. 5.11.32).

8.13.1 Interface to calling client

Requirements in this sub-section apply only to clients that support call offer as a calling client and servers that support call offer for a calling client.

8.13.1.1 Client behavior

If, in response to an INVITE request, a client receives a 486 Busy response containing a Call-Info header field with a purpose parameter with value 'queue', the client must cache the URI from that header field for a period long enough to allow the user to request call offer to the busy destination and SHOULD indicate to the user that call offer is possible.

When sending an INVITE request for which the user has requested call offer, a client must include a Request-Disposition header field containing directive 'queue-directive' with value 'queue'. If the user request was for call offer to a destination that had recently responded with a 486 response containing a Call-Info header field as above, the client must place the cached URI in the Request-URI.

8.13.1.2 OpenScape Voice behavior

When delivering a 486 response to an INVITE request, if the server is aware that call offer is available at the busy destination, it must include a Call-Info header field with a purpose parameter with value 'queue' and a suitable URI representing the busy destination.

Note: The server should ensure that the URI delivered in the Call-Info header field is suitable for the client to use in an INVITE request.

On receipt of an INVITE request containing a Request-Disposition header field with directive 'queue-directive' with value 'queue', the server must handle this as a call offer request.

8.13.2 Interface to called client

Requirements in this sub-section apply only to clients that support call offer as a called client and servers that support call offer for a called client.

8.13.2.1 Client behavior

When a client sends a 486 Busy response to an INVITE request, if the state of the client is such that it would be able to accept an offered call as a waiting call, the client must include in the response a Call-Info header field containing the client's contact URI and the purpose parameter with value 'queue'.

Note: A client configured for call waiting (see Sect. 8.12) will not normally be in this situation.

Note: A client will typically have a limit of one waiting call or a small number of waiting calls, and if this limit has already been reached should not include a Call-Info header field with purpose=queue.

When a client is busy and receives an INVITE request containing a Request-Disposition header field containing directive 'queue-directive' with value 'queue':

- if the client is able to accept the offered call as a waiting call, the client must behave as specified for acceptance of a call as a waiting call in Sect. 8.12;
- if the client is unable to accept the offered call, the client must send a 486 Busy response without a Call-Info header field containing 'purpose=queue'.

Note: In accordance with Sect. 8.12, a 180 response will be returned for a successfully offered call. Therefore the server cannot distinguish a successfully offered call (where the client was busy when the INVITE request with queue-directive 'queue' arrived) from a normal alerting call (where the client was free when the INVITE request arrived). A 180 response is preferred to a 182 response in this situation because UAC behavior is better defined (playing of ringback tone).

8.13.2.2 OpenScape Voice behavior

When a server receives a 486 Busy response to an INVITE request, if a Call-Info header field is present with a purpose parameter with value 'queue', the server MAY use this to indicate to the caller the availability of call offer.

Note: Before forwarding the Call-Info header field URI towards the caller, the server should ensure that the URI is meaningful, e.g., by converting a URI based on a local IP address to a globally routable URI.

When sending an INVITE request to the client, if the server is aware that the caller has requested call offer, the server must include a Request-Disposition header field containing directive 'queue-directive' with value 'queue'.

8.14 Third-Party Call Control (3PCC)

RFC 3261, in combination with RFC 3264 allows a server to conduct various aspects of third party call control. In particular it can initiate call establishment, transfer two calls together, deflect a call, etc., typically under the control of a separate application. Use of uaCSTA with 3PCC is outside the scope of the current version of this document.

8.14.1 Third-Party Call Establishment

8.14.1.1 OpenScape Voice Behavior

OpenScape Voice may use third-party call establishment. To initiate call establishment, the OpenScape Voice will send an INVITE request to the "calling" client without an SDP. Following receipt of a 200 response containing an SDP offer, and subject to successfully completing the call to the destination, the OpenScape Voice will return an SDP answer in an ACK request in accordance with RFC 3261.

8.14.1.2 Client Behavior

A client should support third-party call establishment by being able to accept an INVITE request without SDP in accordance with RFC 3261 and by being able to perform auto-answer based on the contents of the Alert-Info header field (subject to authorization).

8.14.2 Third-Party Call Clearing

OpenScape Voice may initiate call clearing by sending a BYE request towards each client (or a CANCEL request in the case of a client that has not answered). No special SIP signaling is involved.

8.14.3 Third-Party Call Hold

OpenScape Voice may send an INVITE request without an SDP offer to a client to solicit an offer in a 2xx response and then insert a=inactive when forwarding that offer to the other client, thereby giving the appearance that it has been placed on hold. Similarly for retrieval it can do likewise but insert a=sendrecv when forwarding the offer to the second client. No special SIP signaling is involved beyond that specified in Sect. 8.2.

8.14.4 Third-Party Consultation Call

OpenScape Voice cannot initiate a consultation call. If the procedures for third party call establishment in Sect. 8.14.1.1 were to be used while a call already exists, the client would regard itself as busy to the incoming call and would send a 486 response or invoke a feature such as call waiting or call forwarding. Even if the existing call could somehow be placed on hold, this would not prevent the client regarding itself as busy.

Note: OpenScape Voice can achieve a form of consultation call by connecting the remote party of the original call to a media server, establishing a call to a third party, and then connecting the client to the third party (using a re-INVITE request). However, the result is only a single call at the client, which the OpenScape Voice can toggle between the two remote parties. From the client's perspective there is no consultation call.

8.14.5 Third-Party Call Transfer

OpenScape Voice may initiate third party blind transfer, attended transfer and semi-attended transfer. Signaling at the SIP client interface to the transferee and the SIP client interface to the transfer target is as specified in Sect. 8.4. Signaling at the SIP client interface to the transferor only involves sending a BYE request to clear the existing call or calls. No special SIP signaling is involved.

8.14.6 Third-Party DTMF Sending

The sending of DTMF by a client is not possible under third-party control.

Note: An OpenScape Voice can send DTMF using a media server. This has no impact on the SIP client interface.

8.14.7 Speaker Volume Adjustment and Microphone Mute

Speaker volume adjustment and microphone mute are currently not possible via SIP signaling and must be controlled locally at the phone..

8.15 Media Security

Media security is a feature that provides encryption and integrity protection of real-time media (audio, video) while transported through the IP network.

Clients may support and OpenScape Voice does support media security for audio and, if applicable, video, which involves the use of SRTP instead of RTP.

8.15.1 Client Behavior

The following requirements apply only to clients that support payload security:

- A client must support TLS transport in accordance with Sect. 5.1.1.2.
- A client must support negotiation of media security in accordance with Sect. 6.7.
- A client must support key management for media security in accordance with Sect. 6.6.

8.15.2 OpenScape Voice Behavior

Note: Servers are required to support TLS (see Sect. 5.1.2.2).

- A server must support negotiation of media security in accordance with Section 6.7, “Key Management for Media Security”.
- A server must support key management for media security in accordance with Section 6.6, “Negotiation of Media Security”.

Note: These requirements are limited to the passing on of information in SDP offers and answers, except where the server also acts as a SIP UA.

8.15.3 Non-Standard Data Considerations

A new option tag is defined for use in the Supported header field, and hence extends the Supported header field with a new token value: `x-siemens-cdr`. This value is case-sensitive.

Example:

```
Supported: x-siemens-cdr
```

The x-siemens-cdr option tag means that the sending UA supports receipt of the X-Siemens-CDR header field.

8.16 Media recording

Media recording includes endpoint controlled recording and server controlled recording, as described in Appendix K.

8.16.1 Endpoint controlled recording

Clients and servers MAY support endpoint controlled recording as qualified below. The requirements below relate only to clients and servers that support this feature.

This feature requires use of the X-Siemens-Call-Type header field (see Sect. 5.11.45) with values 'recording' and 'recorded'.

8.16.1.1 Interface to a recording client

8.16.1.1.1 Client behavior

A client MAY support endpoint controlled recording as a recording client. A client that supports endpoint controlled recording as a recording client must behave as specified in Appendix K for a recording client.

Note: This includes the ability to establish a second call to the recording server, carry out the necessary media mixing, send the X-Siemens-Call-Type header field with value 'recording' in the INVITE request to the recording server and optionally send the X-Siemens-Call-Type header field with value 'recorded' in a message on the recorded call's dialog.

8.16.1.1.2 OpenScape Voice behavior

A server must support endpoint controlled recording, as specified in Appendix K for a SIP server.

Note: This includes receipt of an X-Siemens-Call-Type header field with value 'recording' in an INVITE request and with value 'recorded' on a message related to a recorded call.

8.16.1.2 Interface to a non-recording client

8.16.1.2.1 Client behavior

A client must support endpoint controlled recording, as specified in Appendix K for a (non-recording) SIP endpoint.

Note: This includes receipt of an X-Siemens-Call-Type header field with value 'recorded'.

8.16.1.2.2 OpenScape Voice behavior

A server must support endpoint controlled recording on the interface to a non-recording client, as specified in Appendix K for a SIP server.

~~**Note:** This includes sending an X-Siemens-Call-Type header field with value 'recorded' towards a non-recording client, subject to policy~~

8.16.1.3 Server controlled recording

Clients and servers MAY support server controlled recording in accordance with Appendix K as qualified below. The requirements below relate only to clients and servers that support this feature.

This feature requires use of the X-Siemens-Call-Type header field (see Sect. 5.14.1) with value 'recorded' and the use of conventional 3PCC techniques.

8.16.1.3.1 Client behavior

A client must support server controlled recording. as specified in Appendix K for a (non-recording) SIP endpoint.

Note: Requirements are as for a non-recording client for endpoint controlled recording (see Sect. 8.16.1), with the additional use of conventional 3PCC techniques, such as receipt of (re-)INVITE requests without SDP offer.

8.16.1.3.2 OpenScape Voice behavior

A server must support server controlled recording, as specified in Appendix K for a SIP server.

Note: Requirements are as for the interface to a non-recording client for endpoint controlled recording (see Sect. 8.16.1.2), with the additional use of conventional 3PCC techniques, such as sending (re-)INVITE requests without SDP offer. OpenScape Voice does not send the X-Siemens-Call-Type header field with value ‘recorded’ for Server Controlled Recording.

---

9 Support of ISDN Supplementary Services

Table 9.3 lists ISDN supplementary services and where applicable indicates support for corresponding features at the SIP client interface by reference to the appropriate sections in this specification.

ISDN Supplementary Service	Support at SIP Client Interface
Call Waiting	See Section 8.12, “Call Waiting”.
Message Waiting Indication.	See Section 8.10, “Message Waiting Indication (MWI)”.
Call Hold	See Section 8.2, “Call Hold”.
Completion of Calls to Busy Subscriber (CCBS).	See Section 8.11, “Call Completion”.
Completion of Calls on No Reply (CCNR)	See Section 8.11, “Call Completion”.
Anonymous Call Rejection (ACR)	A client or OpenScape Voice can reject a call because the caller is anonymous. No specific SIP signalling is specified for this, although response code 403 Forbidden might be an appropriate choice.
Rejection of Forwarded Calls (RFC)	A client or OpenScape Voice can reject a call because it has been forwarded. No specific SIP signalling is specified for this, although response code 403 Forbidden might be an appropriate choice.
Direct Dialling In (DDI)	This has only an indirect effect on the SIP client interface, in that DDI numbers may map to SIP AoR URIs that clients register against.
Calling Line Identification Presentation (CLIP)	This can be achieved by using the P-Asserted-Identity header field in an INVITE request, or in its absence the From header field.
Calling Line Identification Restriction (CLIR)	This feature should be performed by a OpenScape Voice. A client can receive an INVITE request with no P-Asserted-Identity header field and an anonymous URI value in the From header field.
Connected Line Identification Presentation (COLP)	This can be achieved by using the P-Asserted-Identity header field in a 18x or 200 response to an INVITE request.
Connected Line Identification Restriction (COLR)	This feature should be performed by OpenScape Voice. A client can receive an INVITE response with no P-Asserted-Identity header field.
Malicious Call Identification (MCID)	The SIP client interface provides no support for this. It can be performed at OpenScape Voice.
Calling Name Identification Presentation (CNIP)	As for CLIP.
Calling Name Identification Restriction (CNIR)	As for CLIR.
Connected Name Identification Presentation (CONP)	As for COLP.
Normal Call Transfer (CT)	See Section 8.2, “Call Hold”.
Single Step Call Transfer (SSCT).	See Section 8.1, “Identification Services”.
Call Forward Busy (CFB)	See Section 8.8, “Call Diversion”.
Call Forward No Reply (CFNR).	See Section 8.8, “Call Diversion”.

Table 9.3 ISDN Supplementary Services (Seite 1 von 2)

ISDN Supplementary Service	Support at SIP Client Interface
Call Forward No Logged In (CRNL)	This is not explicitly supported at the SIP client interface. However, it can be performed within OpenScape Voice, in which case it will appear as call forwarding unconditional when signalled at the SIP client interface.
Call Forward Unconditional (CFU)	See Section 8.8, “Call Diversion”.
Call Deflection (CD)	See Section 8.8, “Call Diversion”.
Line Hunting (LH)	This feature has no impact on the client interface. It can be performed within OpenScape Voice.
Advice of Charge (AOC)	Not supported.
Reverse Charging (REV)	Not supported.
Conference Call (CONF)	See Section 8.7, “Conferencing”.
Three Party Service (3PTY)	See Section 8.7, “Conferencing”.
Meet Me Conference (MMC)	No special capabilities are required at the SIP client interface to allow users to call a meet-me conference.

Table 9.3 ISDN Supplementary Services (Seite 2 von 2)

A TLS Connectivity Checking

This procedure will apply when support is indicated in the response to the REGISTER request by means of the presence of the string "connectivity-check" in the Server header field e.g.

Server: OpenScape Voice_v3.0 connectivity-check

or

Server: connectivity-check

Note: In particular this should deal with SIP client control switchover to another node. The TLS connection to the old node will be found to be broken, and the subsequent connection attempt should result in connection to a different node.

A SIP Client should be able to be configured with a time interval (T) that determines the time between connectivity checks. Following successful completion of a connectivity check, a SIP Client should conduct the next check at time T-R later, where R is a random number between 0 and 20% of T. The Client should use an initial value of R that is seeded by MAC address.

Note: If T is 120s, then the interval between one check and the next will be a random value between 96s and 120s. This randomness will help to even the load.

Note: Seeding in this way should ensure that if many endpoints start up at the same time they will not all perform the first connectivity check at the same time.

To perform a check, the SIP Client sends a two-octet message, where the first octet has the value 0x00 and the second octet contains a sequence number in the range 0 to 255, which is incremented cyclically for each check. The check is successful if the SIP Client receives an appropriate response or a SIP message within 5 s.

Note: The two-octet message is chosen not to conflict with any SIP message. It can therefore be handled before passing to the SIP stack.

Note: The value of 5s should normally be sufficient for at least one TCP retransmission.

If the SIP UA fails to receive the expected message within that time, it shall repeat the same message (without changing the sequence number). The SIP client discards any messages that are not SIP messages and are not expected connectivity check responses.

If after five attempts the connectivity check has not passed, the SIP client should consider the TLS connection and/or its underlying TCP connection to have failed and shall attempt to establish a new TCP connection and a new TLS connection. On successful establishment of a new TLS connection the endpoint shall attempt to register again.

B ABNF Definition of SIP Headers Server and User-Agent

The Server header field contains information about the software used by the UAS to handle the request.

The User-Agent header field contains information about the UAC originating the request.

The definition of these headers is (RFC 3261):

```
token           =  1*(alphanum / "-" / "." / "!" / "%" / "*"
                   / "_" / "+" / "`" / "'" / "~" )
Server          =  "Server" HCOLON server-val *(LWS server-val)
User-Agent      =  "User-Agent" HCOLON server-val *(LWS server-val)
server-val      =  product / comment
product         =  token [SLASH product-version]
product-version =  token
```

This procedure will apply when support is indicated in the response to the REGISTER request by means of the presence of the string "connectivity-check" in the Server header field e.g.

Server: OpenScape Voice_v3.0 connectivity-check

or

Server: connectivity-check

Note: In particular this should deal with SIP client control switchover to another node. The TLS connection to the old node will be found to be broken, and the subsequent connection attempt should result in connection to a different node.

A SIP client should be able to be configured with a time interval (T) that determines the time between connectivity checks. Following successful completion of a connectivity check, a SIP Client should conduct the next check at time T-R later, where R is a random number between 0 and 20% of T. The Client should use an initial value of R that is seeded by MAC address.

Note: If T is 120s, then the interval between one check and the next will be a random value between 96s and 120s. This randomness will help to even the load.

Note: Seeding in this way should ensure that if many endpoints start up at the same time they will not all perform the first connectivity check at the same time.

To perform a check, the SIP client sends a two-octet message, where the first octet has the value 0x00 and the second octet contains a sequence number in the range 0 to 255, which is incremented cyclically for each check. The check is successful if the SIP Client receives an appropriate response or a SIP message within 5 s.

Note: The two-octet message is chosen not to conflict with any SIP message. It can therefore be handled before passing to the SIP stack.

Note: The value of 5s should normally be sufficient for at least one TCP retransmission.

If the SIP UA fails to receive the expected message within that time, it shall repeat the same message (without changing the sequence number). The SIP Client discards any messages that are not SIP messages and are not expected connectivity check responses.

If after five attempts the connectivity check has not passed, the SIP Client should consider the TLS connection and/or its underlying TCP connection to have failed and shall attempt to establish a new TCP connection and a new TLS connection. On successful establishment of a new TLS connection the endpoint shall attempt to register again.

C SIP—Identification Services (Display Services)

C.1 Originating User Identification (aka Calling Line Identity)

The following procedures describe inclusion of identity information in an INVITE request.

The respective headers can also be included in other requests, e.g. BYE. Although not explicitly covered here, this additional capability is analogous to the procedures for INVITE.

C.1.1 UAC (Originating Client) Procedures

Every INVITE request must contain a (syntactically) valid From: field: If a display name is present, its contents are ignored. If an OpenScape Voice name is configured it is used instead.

- The client must always provide a sip: or sips: URI with the AoR which matches the OpenScape Voice subscriber's identity.
- If identity presentation is allowed, the OpenScape Voice subscriber's Permanent Presentation Status must not restrict the identity presentation, otherwise the user must utilize the OpenScape Voice Calling Identity Delivery Suppression - Suppression (CIDS) per-call feature to override the restriction or use the appropriate Calling Number Delivery Blocking (CNDB), or Calling Line Identity Restriction (CLIR) feature to toggle the identity presentation restriction for the call.
- If identity presentation is restricted, the OpenScape Voice subscriber's Permanent Presentation Status must either restrict the identity presentation or the user must utilize the OpenScape Voice Calling Identity Delivery Suppression - Suppression (CIDS) per-call feature to restrict or suppress the identity presentation or use the appropriate Calling Number Delivery Blocking (CNDB), or Calling Line Identity Restriction (CLIR) to toggle the identity presentation restriction for the call.
- The presentation of the display name may be allowed or restricted on a per-call basis using the OpenScape Voice Calling Name Delivery Block feature, or in combination with the Calling Number by using the Calling Identity Delivery Suppression - Suppression (CIDS) feature.

C.1.2 Outbound OpenScape Voice Procedures

When sending an INVITE request for a new dialog OpenScape Voice (the server that is able to authenticate the called user) shall add a P-Asserted-Identity header field if the called user is configured to receive public identities in the From header field. The P-Asserted-Identity header field will contain a sip: or sips: URI with the public identity of the calling user for an external call or the OpenScape Voice External Caller ID of the calling user if configured.

Examples of From header field sent to a SIP Subscriber:

- Name and Number Allowed

  ```
  From: "John Michaels"
  <sip:12345@10.0.0.100>;tag=snl_J9IFm9h881
  ```

- Name and Number Restricted (or Name Unavailable and Number Restricted)

  ```
  From:
  <sip:anonymous@anonymous.invalid>;tag=snl_J9IFm9h881
  ```

- Name Restricted or Name Unavailable

  ```
  From: <sip:12345@10.0.0.100>;tag=snl_J9IFm9h881
  ```

- Number Restricted

  ```
  From: "John Michaels"
  <sip:anonymous@anonymous.invalid>;tag=snl_J9IFm9h881
  ```

- Name and Number Unavailable

  ```
  From: <sip:10.0.0.100>;tag=snl_J9IFm9h881
  ```

- Number Unavailable

  ```
  From: "John Michaels"
  <sip:10.0.0.100>;tag=snl_J9IFm9h881
  ```

Note: These examples are also applicable for a P-Asserted-Identity header field.

For calls to a SIP subscriber, if the terminating SIP UA indicates support for the SIP UPDATE method and the calling party identity changes before the call is answered, OpenScape Voice sends a SIP UPDATE request without an SDP including a P-Asserted-Identity header field containing the new calling party identity. This happens, for instance, during Semi-Attended Transfer scenarios:

- A calls B; B answers
- B consults to C
- The SIP INVITE request to C contains B's identity in the From header field

- C is ringing
- B transfers A to C

IF C supports the SIP UPDATE method, OpenScape Voice sends a SIP UPDATE request to C with A's identity in the P-Asserted-Identity header field.

C.1.3 Inbound OpenScape Voice Procedures

On receipt of an INVITE request including a P-Asserted-Identity header but no Privacy:id header, the OpenScape Voice serving the called user may pass on the P-Asserted-Identity header in the INVITE sent to the UAS, dependent on policy, if it trusts the preceding (upstream) element. However, if a Privacy:id header is present a P-Asserted-Identity header will not be passed on to the UAS unless the called user has the privilege (assigned via OpenScape Voice configuration) to override the presentation restriction (e.g. an emergency centre).

In general, OpenScape Voice will ignore any P-Asserted-Identity header received from a client. See special exception case described in section Sect. 5.11.21.2.

C.1.4 UAS (Terminating Client) Procedures

The UAS may render calling user identity information - display name if present and URI - to the called user according to local policy. The following sources may be available for this purpose from the received INVITE request; how the sources are used is however a matter of implementation or policy rules (including the handling of conflicting information):

- If a P-Asserted-Identity header header is present it is providing a public identity of the caller which may be rendered to the user, otherwise:
- The content of the From: header of the INVITE message can be used, possibly indicating that this is unauthenticated information.

For actual presentation the UAS may also make use of a matching local directory entry (e.g. display a locally stored name rather than a URI).

C.2 Terminating User Identification (aka Connected Line Identity)

The following sub-clauses describe inclusion of identity information in 200 OK responses for INVITE requests.

C.2.1 Inbound OpenScape Voice Procedures

On receipt of a 200 OK response to an INVITE request, an OpenScape Voice serving the called user may add a P-Asserted-Identity header if it can authenticate the answering user, e.g. if the response arrives via a TLS connection over which the user has previously been authenticated, and it trusts the next element upstream (w.r.t the direction of the request). The P-Asserted-Identity header will contain a sip: or sips: URI with the AOR of the answering user. The URI may be preceded by an optional display name if available. If a Privacy:id header was received in the response then the connected number will be anonymized. A privacy header is ignored if received from an interface which is not trusted.

Note: For proper security (prevention of Man-in-the-Middle attacks) TLS should be used between the UAS and the inbound OpenScape Voice.

OpenScape Voice will not include any P-Asserted-Identity header in the final response if the called user requested privacy and the next element upstream is not trusted.

C.2.2 Outbound OpenScape Voice Procedures

OpenScape Voice shall include a P-Asserted-Identity (PAI) header field within the following SIP response codes before the call is answered:

- 180 Ringing: PAI contains the alerting party identity (name and number).
- 181 Call Is Being Forwarded: PAI contains the diverting party identity.
- 183 Session Progress: PAI contains the called party identity.

On receipt of a final (INVITE) 200 OK response including a P-Asserted-Identity header but no Privacy:id header, OpenScape Voice serving the calling user MAY pass on the P-Asserted-Identity header in the final response sent to the UAC. However, if a Privacy:id header is present a P-Asserted-Identity header will not be passed on to the UAC unless the calling user has the privilege to override the presentation restriction (e.g. an emergency centre).

If the call is terminated with a final 4xx/5xx/6xx response, it may include a P-Asserted-Identity header field containing the releasing party identity (e.g. busy party) if received from a trusted interface or the call terminates before answer in the same OpenScape Voice serving the calling user.

C.2.3 UAC (Originating Client) Procedures

The UAC may render connected user identity information - display name if present and URI - to the calling user according to local policy. The only source available for this purpose according to the present specification is a P-Asserted-Identity header, if present and supported and the preceding element is trusted; if a Privacy:id header is also present and supported identity information from P-Asserted-Identity must not be displayed unless the user has the privilege to override privacy (otherwise the P-Asserted-Identity header should not have been received at all);

For actual presentation the UAC may also make use of a matching local directory entry (e.g. display a locally stored name rather than a URI).

C.3 Mid-Dialog Requests

The procedures of Sect. C.1 and Sect. C.2 above will also apply to target refresh requests and responses within an established dialog, for instance as a result of a call transfer performed by OpenScape Voice or a participant in another network. In other words, Sect. C.1 will also apply to re-INVITE requests and Sect. C.2 to re-INVITE responses.

The headers P-Asserted-Identity and Privacy:id ought to be interpreted depending on the context. For example they will identify the connected user rather than the calling user of the original call if included in a re-INVITE request sent "backwards" compared to the original INVITE request.

If the re-INVITE is caused by a retargeting service (transfer, park/pickup, conference join, etc.) the headers will identify the new participant, e.g. the transferred or transferred-to user in the case of call transfer.

D SIP—Media Hold

There are several procedures for call hold specified in the various versions of the SIP protocol. These procedures concentrate on the manipulation of media streams but do not provide explicit information that Call Hold was invoked. Therefore, the Held User Agent can only implicitly deduce that Call Hold was invoked due to the change of media stream properties.

RFC 2543 specified that placing a user on hold is accomplished by sending a re-INVITE request setting the connection address to 0.0.0.0 in the SDP connection data line. The usage of the 0.0.0.0 IP address for putting a call on hold is now deprecated, since it doesn't allow for RTCP to be used with held streams, doesn't work with IPv6, and breaks with connection oriented media.

However, since implementations use this method today, a UA should support receipt of the 0.0.0.0 IP address for reasons of backward compatibility. Nevertheless sending of the 0.0.0.0 IP address SHALL no longer be used for new implementations, unless there exists a requirement to suppress generating of RTCP data.

Note: Suppressing RTCP is outside of the scope of this specification.

Basic procedures for Hold are described in RFC 3264. Although RFC 3264 recommends the usage of SDP with the media direction attributes sendonly or inactive to put a media stream on hold, it indicates also the possibility to use port number 0 in order to suppress the sending of media.

Note: The following description assumes there is only a single active media stream for an active dialog. The operation of media hold with multiple active dialogs is outside the scope of this document.

D.1 Actions at the Holding UA

Hold may be invoked by the Holding User with optional provisioning of MOH by the Holding User's UA or by the OpenScape Voice.

D.1.1 Normal Procedures for Hold

On receipt of a user indication to put an active dialog on hold, the Holding UA should send a SIP re-INVITE request according to RFC 3261 to the Held UA with SDP offer information according to the template in Table J.4. The SDP version field should be incremented from the last SDP offer received or sent. This new

SDP offer instructs the Held UA to stop sending media. The inactive media direction attribute should be set in the SDP information, unless the UA is able to provide MOH, which is indicated by setting the sendonly media direction attribute.

The SIP re-INVITE request should contain the Held UA's URI in the Request URI and the following headers (optional usage of other headers is not prohibited):

- To: URI as suitable according to previous requests/responses, To-Tag;
- From: URI as suitable according to previous requests/responses, From-Tag,
- Contact: URI of the Holding UA,
- Content-Type: e.g. application/sdp;
- Authorization: as specified in RFC 2617 and RFC 3261,
- Call-ID, Via, Max-Forwards, CSeq, Content-Length according to RFC 3261:

Item	Value
Version	v=0
Origin	o=<username> <session id> <version> IN IP4 <address>
Session Name	s=-
Time	t=0 0
Connection data	c=IN IP4 <address>
Media	m=audio <Port number> RTP/AVP 0
Media attribute	a=rtpmap:0 PCMU/8000
Media attribute	a=sendonly/inactive

Table J.4 Example SDP data using sendonly/inactive media direction attribute for hold

On receipt of the corresponding 200 OK response from the Held UA indicating the recvonly or inactive media direction attribute in the SDP information, Call Hold is successfully invoked.

On receipt of a SIP re-INVITE request without a SDP offer information, the Holding UA should send a 200 OK response with SDP offer information and the media direction attribute as desired.

On receipt of an ACK acknowledgement message with SDP answer and a sendonly or inactive media direction attribute in the SDP information the Holding UA should stop sending media towards the Held UA. In case of inactive, the Holding UA should provide MOH locally for the held media stream. In case of sendonly, the Holding UA may check afterwards, if RTP is received for the media stream offered with a sendonly media direction attribute. If not, the Holding UA may provide MOH locally for the media stream.

D.1.2 Normal Procedures Retrieval

On receipt of a user indication to retrieve a dialog from hold, the Holding UA should send a SIP re-INVITE request to the Held UA with SDP offer information according to the template in Table J.5. The version field should be incremented from the last SDP offer received or sent. The sendrecv media direction attribute should be set in the SDP offer.

Note: In some circumstance another SDP attribute may be more suitable.

This instructs the Held UA to restart sending and, in case of a former inactive media direction attribute, receiving media.

Item	Value
Version	v=0
Origin	o=<username> <session id> <version> IN IP4 <address>
Session Name	s=-
Time	t=0 0
Connection data	c=IN IP4 <address>
Media	m=audio <Port number> RTP/AVP 0
Media attribute	a=rtpmap:0 PCMU/8000
Media attribute	a=sendrecv

Table J.5 Example SDP data using sendrecv media direction attribute for retrieval

D.1.3 Exceptional Procedures

On receipt of 488 (Not Acceptable Here) response from the Held UA the optional SDP answer indicates port number 0 for all m-lines the Holding UA may indicate to the Holding User that the request for Call Hold was rejected.

Rejection of Hold does not prevent the UA to carry out further actions, e.g. still providing MOH. Nevertheless, if no further actions are taken by the UA, the RTP stream, that was established prior to the Hold request, shall remain active as specified in RFC 3264 and RFC 3261.

D.2 Actions at the OpenScape Voice serving the Holding UA

D.2.1 Normal Procedures for Hold

On receipt of a SIP re-INVITE request from the Holding UA with SDP offer information indicating the sendonly media direction attribute, the OpenScape Voice may either

1. forward the SIP INVITE request and its response unchanged or

2. (via OpenScape Voice configuration) override the sendonly media direction attribute and introduce a media server.

If the SDP offer information indicates the inactive media direction attribute, the OpenScape Voice may either

1. forward the SIP INVITE request and its response unchanged or

2. store the received SIP INVITE request and establish a dialog with a media server on behalf of the Holding User.

When a media server is introduced , the Holding OpenScape Voice will send a SIP INVITE request to the media server without SDP information in the body.

Note: Usage of another protocol, e.g. MGCP, for the interface towards the media server is not prohibited, but out of the scope of this specification.

On receipt of a 200 OK response including a SDP offer from the media server, the OpenScape Voice will replace the SDP information in the stored SIP INVITE request by the received SDP and forward this SIP INVITE request towards the Held UA.

On receipt of a 200 OK response from the Held UA indicating the recvonly media direction attribute in the SDP information, the OpenScape Voice will forward the 200 OK response to the Holding UA, if not already sent, and will send an ACK acknowledgement message with an SDP answer to the media server. The SDP offer of the media server will be answered with the media direction attribute set to recvonly or inactive as received from the Held.

Note: The Held UA will expect RCTP packets being sent from the media server and will send its RTCP packets to the media server. The Holding UA should be aware of this situation.

Now, in the case of the Real Time Transport Protocol [RFC 3550], RTCP is still sent and received for sendonly, recvonly, and inactive streams. The Holding UA and Held UA do not have a clue that the SDP offer/answer was manipulated.

Thus the Holding UA still sends RTCP packets to the Held UA. On the other hand, the Held UA does not expect to receive RTCP packets from the Holding UA any longer due to the newly negotiated SDP parameters. In addition the Held UA is sending its RTCP packets to the media server instead of sending it to the Holding UA. Therefore the Holding UA will not receive the expected RTCP Packets.

In order to resolve this situation, the OpenScape Voice will immediately renegotiate the SDP parameters and send SIP re-INVITE request with a SDP offer to the Holding UA with the SDP parameter as in the previous SDP answer, except that the version field in the SDP o-line shall be incremented (as specified in (RFC 3264) and the port number(s) in the m-line(s) SHALL be set to zero. This terminates the RTP stream. In order to keep the SDP version numbers at the Holding UA and Held UA synchronized, the OpenScape Voice will, on receipt of the 200 OK response from the Holding UA, send a SIP re-INVITE request with a new SDP offer to the Held UA with the same SDP parameters as in the previous SDP offer, except that the version field in the SDP o-line shall be incremented. The port number(s) in the m-line(s) will be the same as in the previous SDP offer. The 200 OK response from the Held UA will not be forwarded to the Holding UA.

D.2.2 Normal Procedures for Retrieval

On receipt of a SIP re-INVITE request from the Holding UA with SDP offer information indicating the sendrecv media direction attribute, i.e. the Holding User wants to retrieve a held call, the OpenScape Voice will in case the OpenScape Voice invited a media server for MOH send a BYE request to the media server(s) that provide(s) MOH, and forward the SIP re-INVITE request unchanged to the Held UA. Further actions will be performed according to RFC 3261.

Simultaneous Hold will be handled as specified in Section D.4, “Simultaneous Hold” below.

D.2.3 Exceptional Procedures

If OpenScape Voice initiated dialog establishment to the media server fails, the OpenScape Voice will forward the unchanged SIP re-INVITE request from the Holding UA to the Held UA.

D.3 Actions at the Held UA

D.3.1 Normal Procedures

On receipt of a SIP re-INVITE request from the Holding UA with SDP offer information including a sendonly or inactive media direction attribute in the SDP information, the Held UA should check if the SDP offer is acceptable. If so, the Held UA should stop sending media towards the Holding UA and send a SDP answer according to (RFC 3264) indicating the recvonly or inactive media direction attribute in a 200 OK response to the Holding UA. In case of inactive, the Held UA should provide MOH locally for the held media stream. In case of sendonly, the Held UA may check afterwards, if RTP is received for the media stream offered with a sendonly media direction attribute. If not, the Held UA may provide MOH locally for the media streams.

Note: Checking for reception of MOH is advisable in case of Holding UAs that are not conformant to this specification. The UA providing local MOH must be prepared that RTP may arrive from the Holding User's side at any time.

The locally provided MOH depends on the type of media stream, e.g. music/ announcements for audio streams, a freeze frame for video streams or other implementation-specific indications.

If the SDP offer is not acceptable, the Held UA should send a 488 (Not Acceptable Here) response to the Holding UA in order to reject the SDP offer. According to section 14.1 of RFC 3261, this leaves the SDP session parameters unchanged as if no SIP re-INVITE request had been issued.

On receipt of a SIP re-INVITE request without a SDP offer information, the Held UA should send a 200 OK response with SDP offer information and the media direction attribute as desired.

On receipt of an ACK acknowledgement message with SDP answer and a sendonly or inactive media direction in the SDP information the Held UA should stop sending media towards the Holding UA. In case of inactive, the Held UA should provide MOH. In case of sendonly, the Held UA may check afterwards, if RTP is received. If not, the Held UA may provide MOH locally.

Once held, the receipt of a SIP re-INVITE request with SDP offer information including a sendrecv media direction attribute indicates retrieval. This request should be treated as specified in RFC 3261, RFC 3264 and RFC 3311.

Also once held, the receipt a of SIP re-INVITE request without a SDP offer information, the Held UA should send a 200 OK response with SDP offer information and the media direction attribute as desired. If afterwards an ACK

acknowledgement message with SDP answer and a sendrecv media direction attribute in the SDP information is received, the Held UA should interpret this as retrieval from hold.

D.3.2 Exceptional Procedures

No special procedures deviating from those specified in RFC 3261 and RFC 3264 are necessary, if Hold is handled as specified in this document. Nevertheless the Held UA should be prepared to receive the following SDP data.

According to the now deprecated RFC 2543 placing a user on hold was accomplished by setting the connection address to 0.0.0.0. This method is no longer recommended. For interworking reasons the Held UA should be capable of receiving SDP with connection address of 0.0.0.0, in which case it means that neither RTP nor RTCP should be sent to the peer.

Although RFC 3264 recommends the usage of the media direction attributes sendonly or inactive to put a media stream on hold, it indicates also the possibility to use port number 0 in order to suppress the sending of media. For interworking reasons the Held UA should be capable of receiving SDP with port number 0.

D.4 Simultaneous Hold

D.4.1 Normal Procedures for Simultaneous Hold

Note: Pre-condition for this scenario is that Hold was invoked successfully.

On receipt of SIP INVITE request from the Held UA indicating the inactive media direction attribute in the SDP information, the Holding UA may indicate to the User that Simultaneous Call Hold is invoked by the Held UA. The Holding UA should send a 200 OK response with a SDP answer indicating the inactive media direction attribute in the SDP offer.

In case that the OpenScape Voice invited one a media server according to Sect. D.2 , the OpenScape Voice will send the BYE request to the media server.

D.4.2 Normal Procedures for Retrieval from Simultaneous Hold

Note: This section uses the terms Holding User and Held User although both user maintain these roles simultaneously in this scenario.

On receipt of a user indication to retrieve a dialog from Simultaneous hold, the UA should send a SIP re-INVITE request to its peer UA with SDP offer information according to the template in Table J.5. The version field should be incremented from the last SDP offer received or sent. The sendrecv media direction attribute should be set in the SDP offer. This allows the peer UA to restart sending and receiving media.

If in case of Simultaneous Call Hold a SIP INVITE request is received from the Held UA indicating the recvonly or sendrecv media direction attribute in the SDP information, the Holding UA should perform one of the following options:

- Indicate that it is still holding the call by returning an SDP answer information containing the inactive media direction attribute towards the retrieving UA.
- Indicate that it is still holding the call and provide MOH by returning an SDP answer information containing the sendonly media direction attribute towards the retrieving UA.
- If it is willing to start to send media again and the SDP offer information contained the sendrecv media direction attribute then the Holding UA SHALL send an SDP answer information containing the sendrecv media direction attribute.

E SIP—Call Transfer

E.1 General Considerations

E.1.1 SIP Extensions

Call transfer requires the support of certain SIP extensions. The procedures below are based on the REFER method (RFC 3515) in conjunction with Referred-by (RFC 3892) and Replaces (RFC 3891) headers, and on 3pcc procedures (RFC 3725). The REFER method instructs a UA to perform an action, in this case establish a call to the indicated destination, or link two existing call legs to form a new call. For attended - and, in certain cases, semi-attended - transfer the inclusion of a Replaces header further instructs the destination to replace an existing dialog with the new one initiated by the INVITE just received. In all cases the Referred-by header, if present, identifies the initiator of call transfer. The progress of the transferred call is reported to the transferor using the NOTIFY method (RFC 3265).

User agents and servers capable of participating in call transfer should indicate support for these extensions during dialog establishment. In particular a Client or OpenScape Voice should indicate within requests (e.g. INVITE) and responses (e.g. 200 OK) exchanged during call establishment whether it supports

- Methods REFER and NOTIFY, indicated in the Allow header
- Extensions replaces and referred-by, indicated in the Supported header

---

Note: OpenScape Voice does not currently support implicit subscription to the 'refer' event when a REFER request is received, and does not send NOTIFY messages to report the progress of the REFER (transfer) request. OpenScape Voice support for these capabilities is planned for release V5.0 and this document will be updated at that time to describe the procedures.

E.2 Procedures for Blind (Unattended) Transfer

E.2.1 Actions at the Transferor Endpoint

E.2.1.1 Normal Procedures

In order to initiate a blind transfer on an active call between transferee and transferor the transferor endpoint should send a REFER request on the existing dialog towards the OpenScape Voice. Depending on local policy, the transferor endpoint may put the transferee on hold prior to sending the REFER request.

Content of REFER:

Request URI: Contact URI associated with the transferee (remote target URI of original dialog)

To: as for original dialog

From: as for original dialog

Refer-to: <URI of transfer target [;method=INVITE]> (most likely an AoR, but Contact also possible in specific use cases)

Contact: Contact URI of transferor (as for original dialog)

Referred-by: URI (AoR) of transferor (optional)

On receipt of a BYE request on dialog2 after REFER has been sent on dialog1 the transferor endpoint should proceed with clearing dialog2 and the associated session.

On receipt of a BYE request on dialog1 after receipt of a "202 Accepted" response to the sent REFER request the transferor endpoint should proceed with clearing dialog1 and the associated session.

E.2.1.2 Exceptional Procedures

If the REFER request fails (i.e. a final 4xx, 5xx or 6xx (REFER) response is received) the original call should remain in place. Depending on the failure reason, the transferor endpoint MAY attempt itself to call the transfer target and perform an attended transfer according to Sect. E.3. If no further transfer attempt is started the transferor endpoint should inform user A of the failure and should retrieve the original call from hold if it has been put on hold before.

If the remote side (transferor server or transferee) terminates the original dialog - by sending a BYE request for the original call and/or terminating the 'refer' subscription - before a final (INVITE) response for the transferred call has been reported the transferor endpoint should consider the transfer attempt finished.

If the original call was already released the transferor endpoint may receive a recall from the remote side (OpenScape Voice or transferee).

E.2.2 Actions at OpenScape Voice

Upon receipt of a REFER request the OpenScape Voice serving user A will behave as follows:

- Act upon the REFER request and perform call transfer as described below.

E.2.2.1 Normal Procedures

On receipt of a REFER request on an existing confirmed dialog the will check whether the request is acceptable, and if so respond with "202 Accepted". The OpenScape Voice will then try to establish a call to the transfer target by sending an INVITE message to the URI taken from the Refer-to header of the received REFER request.

The OpenScape Voice will send the INVITE request without SDP. On receipt of an SDP offer from the transfer target - in the 200 OK (INVITE) response - the OpenScape Voice will send a re-INVITE (or optionally UPDATE) request with this SDP offer to the transferee and then pass on the transferee's SDP answer to the transfer target (in the ACK request).

E.2.2.2 Exceptional Procedures

If a REFER request cannot be accepted the OpenScape Voice will return a failure (REFER) response (4xx, 5xx, 6xx) describing the reason, chosen according to RFC 3515.

The OpenScape Voice will send a BYE request to the transferor endpoint on the original dialog. This will cause the transferor endpoint to release the call. In this case the OpenScape Voice may initiate a recall later on if the transfer attempt fails.

If the transferee sends a BYE request before or while transfer is taking place the OpenScape Voice will

- send to the transferor endpoint a BYE request for the original call if it still exists;
- send to the transfer target a CANCEL request for the transferred call if already initiated;

These actions terminate the original dialog, and no recall will be initiated.

If the call to the transfer target cannot be established and the transferee is still available the OpenScape Voice will retain the original call if still available, and may initiate a recall to the transferor endpoint if supported and if the original call leg towards the transferor endpoint was already released (by the transferor endpoint or the transferor server).

E.2.3 Actions at the Transferee Endpoint

The transferee endpoint may receive re-INVITE or UPDATE requests according to 3pcc procedures.

E.2.3.1 Transfer using 3pcc Procedures

No specific actions beyond basic SIP are required.

E.2.4 Actions at the Transferee OpenScape Voice

In our current implementations no call transfer specific actions are required from the OpenScape Voice serving user B - the OpenScape Voice will pass all requests and responses to and from the transferee endpoint without acting on them (other than the usual SIP message handling of a B2BUA).

E.2.5 Actions at the Transfer Target Endpoint and Transfer Target Server

No special action required.

E.3 Procedures for Attended Transfer

An attended transfer requires that besides the (answered) call to be transferred (between transferee and transferor, in the following called dialog1) another answered call between transferor and transfer target exists (in the following called dialog2).

Note that dialog1 and dialog2 actually designate a chain of dialogs whenever one or more B2BUAs are present in the respective call signalling path.

E.3.1 Actions at the Transferor Endpoint

E.3.1.1 Normal Procedures

To initiate transfer the transferor endpoint should send a REFER request on dialog1 towards the transferee. Depending on local policy, the transferor endpoint may put the transferee on hold prior to sending the REFER request. The transferor endpoint may also put the transfer target on hold prior to sending the REFER request.

Content of REFER:

Request URI:Contact URI associated with the transferee (remote target URI of dialog1)

To: as for dialog1

From: as for dialog1

Refer-to: <Contact URI of transfer target [;method=INVITE] ?Replaces=dialog2 [&Require=replaces]> (remote target URI of dialog2)

Contact: Contact URI of transferor (as in dialog1)

Referred-by: URI (AoR) of transferor (optional)

Note: The replaces URI-parameter included in Refer-to contains the call-ID, From-tag and To-tag of dialog2. Example (all on one line): Refer-To:<sip:transfertarget@chicago.example.com?Replaces=09045924358817344 5%3Bto-tag%3D9m2n3wq%3Bfrom-tag%3D763231&Require=replaces>

On receipt of a BYE request on dialog2 after REFER has been sent on dialog1 the transferor endpoint should proceed with clearing dialog2 and the associated session.

On receipt of a BYE request on dialog1 after receipt of a "202 Accepted" response to the sent REFER request the transferor endpoint should proceed with clearing dialog1 and the associated session.

E.3.1.2 Exceptional Procedures

If the REFER request fails (i.e. a final 4xx, 5xx or 6xx (REFER) response is received) dialog1 should remain in place. Depending on the failure reason and local policy the transferor endpoint may attempt another attended transfer, e.g. with reversed roles of transferee and transfer target or with user C's AoR instead of the contact URI in the Refer-to header. If no further transfer attempt is started and dialog1 is on hold, the transferor endpoint should retrieve it from hold.

If the remote side (transferor server or transferee) terminates dialog1 - by sending a BYE request for the original call- before a final (INVITE) response for the transferred call has been reported the transferor endpoint should consider the transfer attempt finished.

Note: If either side released the call associated with dialog1 after acceptance of REFER and prior to completion of call transfer the transferor endpoint may receive a recall from the remote side (OpenScape Voice or transferee) if the transfer attempt failed.

E.3.2 Actions at OpenScape Voice

Upon receipt of a REFER request the OpenScape Voice serving user A will behave as follows:

- Act upon the REFER request and perform call transfer as described below.

E.3.2.1 Normal Procedures

On receipt of a REFER request from the transferor endpoint on dialog1 the OpenScape Voice serving user A will check whether the request is acceptable (e.g. check that it "owns" the URI in the Refer-to header field and that it recognizes the dialog identified in the embedded Replaces header field), and if so respond with "202 Accepted".

Then the OpenScape Voice will send on dialog1 a re-INVITE request without SDP towards the transferee. On receipt of an SDP offer from the transferee - in the 200 OK (INVITE) response - the OpenScape Voice will send on dialog2 (with the matching dialog2 being selected by means of the content of the Refer-to header of the received REFER request)

- a BYE request to the transferor endpoint, and
- a re-INVITE (or optionally an UPDATE) request with the transferee's SDP offer towards the transfer target,

and then pass on the transfer target's SDP answer to the transferee (in the ACK request).

The OpenScape Voice will also send a BYE request to the transferor endpoint on dialog1.

E.3.2.2 Exceptional procedures

If a REFER request cannot be accepted the transferor server (OpenScape Voice serving user A) will return a failure (REFER) response (4xx, 5xx, 6xx) describing the reason, chosen according to RFC 3515.

The OpenScape Voice will send a BYE request to the transferor endpoint on dialog1. This will cause the transferor endpoint to release the call. In this case the OpenScape Voice may initiate a recall later on if the transfer attempt fails.

If the transferee releases the call (by sending a BYE request on dialog1) before or while transfer is taking place the OpenScape Voice will

- send to the transferor endpoint a BYE request on dialog1 if the original call still exists;
- send to the transfer target a BYE request on dialog2;

These actions terminate dialog1, and no recall will be initiated.

If the transfer target releases the call (by sending a BYE request on dialog2) before or while transfer is taking place and the transferee is still available the OpenScape Voice will:

- retain the original call (dialog1) if still available, or
- initiate a recall to the transferor endpoint according to Sect. E.5.2, if supported and if the original call leg towards the transferor endpoint was already released (by the transferor endpoint or the transferor server).

E.3.2.3 Actions at the Transferee Endpoint

The transferee endpoint will be involved in attended call transfer as follows:

- The transferee endpoint may receive re-INVITE or UPDATE requests according to 3pcc procedures.

E.3.2.4 Transfer Using 3pcc Procedures

No specific actions beyond basic SIP are required.

E.3.3 Actions at the Transferee OpenScape Voice

E.3.3.1 Transparent Server

In our current implementations no call transfer specific actions are required from the OpenScape Voice serving user B - the OpenScape Voice will pass all requests and responses to and from the transferee without acting on them (other than the usual SIP message handling of a B2BUA).

E.3.4 Actions at the Transfer Target Endpoint

The transfer target endpoint will be involved in attended call transfer as follows:

- The transfer target endpoint may receive re-INVITE or UPDATE requests according to 3pcc procedures.

E.3.4.1 Transfer Using 3pcc Procedures

No special actions are required.

E.3.5 Actions at the Transfer Target OpenScape Voice

E.3.5.1 Transparent Server

In our current implementations no call transfer specific actions are required from the OpenScape Voice serving user C - the OpenScape Voice will pass all requests and responses to and from the transfer target endpoint without acting on them (other than the usual SIP message handling of a B2BUA).

E.4 Procedures for Semi-attended Transfer

Unlike an attended transfer, the second call involved in semi-attended call transfer is not yet fully established. The 'replaces' mechanism cannot be used in this situation in its usual form - in fact RFC 3891 does not permit replacing an early dialog at its UAS (destination) end. Therefore OpenScape Voice currently uses "partial replacement" where OpenScape Voice handles early dialogs towards the transfer target, intercepts the transferred call, and takes care of the 'replaces' request.

A semi-attended transfer requires that besides the (answered) call to be transferred (between transferee and transferor, in the following called dialog1) another early dialog between transferor and transfer target exists (in the following called dialog2) that has at least reached ringing or queued state, in other words has produced a 180 Ringing or 182 Queued response.

Note that dialog1 and dialog2 actually designate a chain of dialogs whenever one or more B2BUAs are present in the respective call signalling path.

E.4.1 Actions at the Transferor Endpoint

E.4.1.1 Normal Procedures

The transferor endpoint can assume that its (early) dialog2 towards the transfer target will remain a single instance between itself and the OpenScape Voice and therefore it may initiate call transfer as if it were an attended transfer, sending a REFER request on dialog1 towards the transferee. The Refer-to header should contain the remote target URI of dialog2 and a 'replaces' parameter for dialog2.

Since it may take a while until user C answers it makes more sense than in the attended transfer case for the transferor endpoint to release the call associated with dialog1 earlier than on completion of call transfer, e.g. on receipt of the "202 Accepted" (REFER) response.

E.4.1.2 Exceptional Procedures

If the REFER request fails (i.e. causes a non-2xx (REFER) final response) the transferor endpoint may start another call transfer attempt instead, e.g. with waiting for an answer. If no further transfer attempt follows the transferor endpoint should retain the alerting call and indicate to user A that transfer has failed.

E.4.2 Actions at the Transferor OpenScape Voice

The procedures below SHALL apply.

E.4.2.1 Normal Procedures

If the OpenScape Voice serving user A receives on dialog1 a REFER request from the transferor endpoint that contains in the Refer-to header a 'Replaces=dialog2' parameter with all dialog2 instances being in the early state

and the request is acceptable (meaning that the OpenScape Voice "owns" the URI in Refer-to and recognizes the dialog identified in the embedded Replaces) the OpenScape Voice will respond with "202 Accepted".

Then the OpenScape Voice

- Will send a "487 Request Terminated" (INVITE) response to the transferor endpoint on ('upstream') dialog2;
- Will take care of any SDP offer/answer exchange required to connect the transferee to a media server providing ringback tone.
- and will send a BYE request to the transferor endpoint on dialog1. This will cause the transferor endpoint to release the call and also its 'refer' subscription. OpenScape Voice may initiate a recall later on if the transfer attempt fails.

The transferor OpenScape Voice may apply a no-answer timer for a pending semi-attended transfer request. The value for this timer may depend on whether transfer is into ringing or into camp-on.

As soon as a dialog2 instance reaches the confirmed state the OpenScape Voice will cancel all other dialog2 instances according to RFC 3261 and continue with the procedures of Sect. E.3 above.

Note: This will comprise sending re-INVITE without SDP or with the OpenScape Voice's own SDP offer towards the transferee, sending re-INVITE with the triggered SDP offer (user B) or the OpenScape Voices own SDP offer towards the transfer target, and, in the former case, relaying the SDP answer (user C) to the transferee. This exchange will also update the remote user identities (B and C, respectively).

E.4.2.2 Exceptional Procedures

If a REFER request cannot be accepted the OpenScape Voice serving user A will return a failure (REFER) response (4xx, 5xx, 6xx) describing the reason, chosen according to RFC 3515.

If the transferee releases the call (associated with dialog1) before or while transfer is taking place the OpenScape Voice shall

- send to the transferor endpoint a BYE request on dialog1 if the original call still exists;
- send to the candidate transfer targets a CANCEL request on the dialog2 instances; and

These actions terminate dialog1, and no recall will be initiated.

If all dialog2 instances fail (i.e. generate non-2xx final (INVITE) responses) or no candidate transfer target answers the call (dialog2 instance) before the no-answer timer expires and the transferee is still available the OpenScape Voice shall

- cancel any dialog2 instances still pending (using standard RFC 3261 procedures);
- retain the original call (dialog1) if still available, or
- initiate a recall to the transferor endpoint according to Sect. E.5.2 below, if supported and if the original call leg towards the transferor endpoint was already released (by the transferor endpoint or the transferor server).

E.4.3 Actions at the Transferee Endpoint

No specific actions are required beyond Sect. E.3.2.3 above.

E.4.4 Actions at the Transferee OpenScape Voice

No specific actions are required beyond Sect. E.3.2 above.

E.4.5 Actions at the Transfer Target Endpoint

Section E.3, “Procedures for Attended Transfer”.

With our current implementations the transfer target endpoint should not receive an INVITE (replaces) request for a dialog2 in the early state; if it did it should reject the request as specified in RFC 3891.

E.4.6 Actions at the Transfer Target OpenScape Voice

Refer to Section E.3, “Procedures for Attended Transfer”.

E.5 Recall

E.5.1 Actions at the Transferor Endpoint

Having initiated a call transfer, the transferor endpoint may check for a certain period after the original call was released any incoming INVITE request for being a recall. If the INVITE contains header X-Siemens-Call-Type:recall-transfer the call should be treated as recall, otherwise the request should be accepted as a new call.

A header X-Siemens-Call-Type:recall-transfer received in an INVITE request should be ignored and the request treated as an ordinary call attempt if no recall was expected.

Any specific actions for a recall are an implementation matter outside the scope of this document.

Note: If the original call (dialog1) has not been released the transferor endpoint may apply local recall actions if a condition for a recall occurs. This is also outside the scope of this document.

E.5.2 Actions at the Transferor OpenScape Voice

If after accepting a REFER request the original (dialog1) call leg towards the transferor endpoint was released but the call leg towards the transferee is still present, the transferor OpenScape Voice may initiate a recall to the transferor endpoint in either of two situations:

- If transfer fails, e.g. (4xx, 5xx, 6xx) final (INVITE) responses are received from all candidate transfer targets; or
- If the transfer attempt times out locally, i.e. the OpenScape Voice sends a CANCEL request to the candidate transfer targets after having waited for a (2xx) final (INVITE) response for an implementation specific period.

Both situations - transfer failure or timeout - may occur in case of blind or semi-attended transfer, the first one possibly also for an attended transfer, however in the attended transfer case the original call is probably still available for recalling user A locally.

The OpenScape Voice will initiate a recall by sending a new INVITE request to the transferor endpoint's contact URI, including user B's identity in header P-Asserted-Identity as well as header X-Siemens-Call-Type:recall-transfer, if this header value is supported. The OpenScape Voice will also send a re-INVITE request on dialog1 to the transferee and connect transferor endpoint and

transferee according to 3pcc procedures. Whether transferor endpoint or transferee is called first and how SDP offers and answers are included depends on exactly when in the course of call transfer execution recall is initiated.

E.5.3 Actions at the Transferee Endpoint or OpenScape Voice—Acting on REFER

After accepting a REFER request, if the original call (dialog1) has already been released but user B is still available, the transferee endpoint or server may initiate a recall to the transferor in either of two situations:

- if transfer fails, i.e. a (4xx, 5xx, 6xx) final (INVITE) response is received from the transfer target in reply to the 'referred' INVITE request; or
- if the transfer attempt times out locally, i.e. the transferee endpoint or server sends a CANCEL request to the transfer target after having waited for a (2xx) final (INVITE) response for an implementation specific period. This situation will not occur in the attended call transfer case.

The transferee endpoint or server should initiate a recall by sending a new INVITE request to the transferor's contact URI (which should still be known from the 'refer' subscription), including user B's identity in header P-Asserted-Identity, an SDP offer (for user B), and header X-Siemens-Call-Type:recall-transfer, if this header value is supported.

A recall may be initiated for all variants of call transfer - blind, attended and semi-attended - although least likely for attended transfer since in this case the original call is probably still available.

F SIP—Group Pick-Up

Note: The Pick-Up Group URI is the feature access code configured for the Group Pick-Up feature. In the following example *01 is the Group Pick-Up feature access code:
sip:*01@10.232.27.102:5060;transport=udp SIP/2.0

F.1 Subscription of Pick-Up Group Members to Pick-Up Service

F.1.1 Actions at the Pick-Up Service

F.1.1.1 Normal Procedures

On receipt of a SIP SUBSCRIBE request from a Pick-Up Group member identifying a Pick-UP Group in the Request-URI, the Pick-UP Service in OpenScape Voice will check if the request is acceptable. If so, it will return a 200 OK response. The 200 OK response will contain the following headers:

- To: URI as received, To-Tag
- From: as received from UA
- Contact: URI of Pick-UP Group
- Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length according to RFC 3261

The From-Tag sent in the 200 OK response SHALL be re-used in the subsequent SIP NOTIFY requests as To-Tag. Afterwards, the Pick-UP Service will send a SIP NOTIFY request to the sender of the SIP SUBSCRIBE request as specified in RFC 3265.

The body of the SIP NOTIFY request will either be empty or may contain dialog state data according to Sect. F.1.2 if a call is currently alerting at a Pick-Up Group member.

Re-subscription/Keep alive will be treated as specified in RFC 3265.

F.1.1.2 Exceptional Procedures

If a SIP SUBSCRIBE request to a Pick-Up Group is not acceptable, the Pick-Up Service will return a 4xx/5xx/6xx response indicating the reason for rejection.

F.1.2 Actions at the Pick-Up Group Member

Note: It is out of scope of this specification how a UA determines to be a Pick-Up Group member for a specific Pick-UP Group—e.g., a SIP UA may be configured or administrated as a member of Pick-UP Group or the UA may learn this due to user input.

F.1.2.1 Normal Procedures

In order to subscribe to the dialog state of the Pick-UP Group, the SIP Client should send a SIP SUBSCRIBE request to the Pick-UP Service in OpenScape Voice. The Request URI of the SIP SUBSCRIBE request should be set to the URI of the Pick-UP Group. The SIP SUBSCRIBE request should contain the following headers:

To: URI of Pick-Up Group

From: URI of Pick-Up Group member, From-Tag

Contact: URI of Pick-Up Group member UA

Event: dialog

Expires: set to an appropriate value (default 120 s)

Accept: application/dialog-info+xml

Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length according to RFC 3261.

On receipt of a 200 OK response to the SIP SUBSCRIBE request and a SIP NOTIFY request from the Pick-UP Service, the subscription is successful. The SIP NOTIFY request should be treated as specified in RFC 3265.

Re-subscription/Keep alive should be treated as specified in RFC 3265].

F.1.2.2 Exceptional Procedures

On receipt of a 401 (Unauthorized) or 407 (Proxy Authentication Required), the Picking-Up UA should act as specified in RFC 3261 and RFC 3265. On receipt of a 4xx/5xx/6xx response to the SIP SUBSCRIBE request from the Pick-UP Service, the subscription is unsuccessful. Further actions are out of scope of this specification.

F.2 Notification About Incoming Call

F.2.1 Actions at the Pick-UP Service

F.2.1.1 Normal Procedures

When the Pick-UP Service gets knowledge about an alerting call at a Pick-UP Group member, all other members will be notified about an incoming call to the Pick-UP Group. Therefore, the Group Pick-UP Service will send a SIP NOTIFY request to all UAs subscribed to the Pick-UP Group.

The Request URI of these SIP NOTIFY requests will be set to the Contact URI received in the previous SIP SUBSCRIBE requests from Sect. F.2.2.1 above and will contain the following headers:

To: URI of Pick-UP Group member, To-Tag (received as From-Tag in the SIP SUBSCRIBE request)

From: URI of the Pick-UP Group, From-Tag (as sent in the To-Tag in 200 OK

Contact: URI of the Pick-UP Group UA

Event: dialog

Content-Type: application/dialog-info+xml

Subscription-State: active; expires= set to appropriate actual value

Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length according to RFC 3265.

The body of the SIP NOTIFY will contain the XML encoded dialog state information of the incoming call according to RFC 3903 and as indicated in the SIP NOTIFY request. The dialog state will be indicated as received. Below is a template for the XML-body:

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
      version="0" state="full" entity="URI of Pick-UP Server">
<dialog id="id value"
      call-id="call-id of the dialog"
      local-tag="to-tag data" < -- if available --;>
      remote-tag="from-tag data" < -- if available --;>
      direction="recipient"
     <state>early</state> < -- or confirmed or terminated --;>
      <local>
```

```
    <identity display="AoR of Target user" </identity>
   </local>
  <remote>
    <identity display="AoR of Wanted user" </identity>
    </target>
   </remote>
 </dialog>
</dialog-info>
```

All Pick-UP Group member UAs, including the Target/Alerting UA, get the SIP NOTIFY request message about an alerting incoming call to a member of the Pick-UP Group.

In addition the Group Pick-UP Service will notify all Pick-UP Group members when the call is not alerting any longer (due to sending state 'confirmed' in the XML body).

F.2.1.2 Exceptional Procedures

If a received SIP NOTIFY request is not acceptable, the Group Pick-UP service will send back an appropriate 4xx response.

F.2.2 Actions at Pick-UP Group Members

F.2.2.1 Normal Procedures

On receipt of a SIP NOTIFY request from the Pick-UP Service, the Pick-UP Group UA should indicate the received information to the User and should send a 200 OK response to the Pick-UP Service.

Note: If there is more than one call ringing at the Pick-UP Group, the NOTIFY may contain all the information on all the dialogs.

The user should be notified that a call is waiting to be picked up.

If dialog state 'confirmed' or 'terminated' is indicated, the user should be notified that the call is no longer waiting to be picked-up. Any stored information related to the indicated dialog should be deleted.

F.2.2.2 Exceptional Procedures

Not applicable.

F.3 Pick-UP by Group Member

F.3.1 Actions at the Pick-UP Service

F.3.1.1 Normal Procedures

On receipt of a SIP INVITE request with the Group Pick-UP access code as the user part of the Request URI , the Pick-UP Service will check if the request is valid, e.g. if there exists at least one early dialog for this Pick-UP Group.

If so, the Pick-UP Service chooses one early dialog. Then the Pick-UP Service will co-ordinate the early dialog from the Picking-Up UA with the early dialog from the Wanted UA in the following way.

The Pick-UP Service will send a 200 OK for the initial SIP INVITE request to the Wanted UA and another 200 OK to the Picking-UP UA. The 200 OK response sent to the Wanted UA will include a Pick-UP Service generated answer information, whereas the 200 OK response sent to the Picking-UP UA will just include a Pick-UP Service generated answer information. On receipt of the ACK acknowledge message from both UAs, the Pick-UP Service will send a re-INVITE request to the Wanted UA with no SDP offer information. On receipt of the 200 OK response with the SDP offer information from the Wanted UA, the Pick-UP Service will send this SDP offer information within a re-INVITE request to the Picking-UP UA. Upon receipt of the 200 OK response with the SDP answer information from the Picking-Up UA the Pick-UP Service will send this SDP answer information in the ACK acknowledgment to the Wanted UA.

If the Group Pick-UP was successful (i.e. Picking-Up UA and Wanted UA are linked together), the Pick-UP Service will send a SIP CANCEL request to the Target UA.

If an alerting call is either picked-up successfully or answered, the Pick-UP Service will send a SIP NOTIFY request to all other UAs of the Pick-UP Group indicating dialog state 'confirmed' in the body.

If an alerting call is terminated before answered, the Pick-UP Service will send a SIP NOTIFY request to all other UAs of the Pick-UP Group indicating dialog state 'terminated' in the body.

F.3.1.2 Alternate Procedures

The procedures presented in (F.3.1.1) above are followed except that the INVITE from the Picking-Up UA does not include an SDP offer. In this case the SDP offer from the Wanted UA is passed to the Picking-Up UA in the 200 OK and the SDP answer from the Picking-UA received in the ACK is passed to the Wanted UA in the 200 OK for the INVITE, completing the Call Pick-UP.

F.3.1.3 Exceptional Procedures

On receipt of a SIP INVITE request with the Group Pick-UP access code as the user part of the Request URI and no dialog is waiting to be picked-up, the Pick-UP Service will send back an appropriate 4xx response..

F.3.2 Actions at the Picking-Up UA

F.3.2.1 Normal Procedures

On invocation of Pick-UP due to receipt of a corresponding indication from the user, the Picking-Up UA should send a SIP INVITE request according to RFC 3261 towards the stored remote-target URI (see Sect. F.2).

The SIP INVITE will indicate in the user part of the Request URI the access code for the Group Pick-UP Service:

- To: remote URI,
- From: URI of Pick-UP Group member, From-Tag
- Contact: URI of Picking-Up UA
- Content-Type: application/sdp
- Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length, Allow, Supported according to RFC 3261]

SDP offer Information SHALL be sent in the body as specified in RFC 3264. On receipt 200 OK response and SDP answer information the Pick-UP was successful.

The Picking-UP UA may receive a re-INVITE request with a SDP offer information (from the Wanted UA) immediately afterwards. In this case the Picking-Up UA should send the SDP answer information in the 200 OK response and act according to RFC 3261 or RFC 3311 respectively

In addition the Picking-Up UA receives - as all other UA of the Pick-UP Group - a SIP NOTIFY request from the Pick-UP Service indicating dialog state 'confirmed' in the body.

F.3.2.2 Alternate Procedures

The procedures presented in (i) above are followed except that the INVITE from the Picking-UP UA does not include an SDP offer.

Since the SDP offer/answer exchange is complete between the Picking-UP UA and the Wanted UA, i.e., no interim SDP has been used, no SDP renegotiation using a re-INVITE with the Picking-UP UA is required.

In addition the Picking-Up UA receives - as all other UA of the Pick-Up Group - a SIP NOTIFY request from the Pick-Up Service indicating dialog state 'confirmed' in the body.

F.3.2.3 Exceptional Procedures

Not applicable.

F.3.3 Actions at the Wanted UA

F.3.3.1 Normal Procedures

On receipt of a 200 OK response to an initial SIP INVITE request including SDP answer information, the Wanted UA should send an ACK acknowledgment. In case that the Pick-UP Service just provided answer information, the Wanted UA may receive a re-INVITE request with no SDP offer information immediately afterwards. In this case the Wanted UA should send the SDP offer information in the 200 OK response and act according to RFC 3261

F.3.3.2 Alternate Procedures

The procedures presented in (i) above are followed except that the 200 OK response includes the SDP offer from the Picking-Up UA.

Since the SDP offer/answer exchange is complete between the Picking-UP UA and the Wanted UA, i.e., no interim SDP has been used, no SDP renegotiation using the re-INVITE with the Wanted UA is required.

In addition the Picking-Up UA receives - as all other UA of the Pick-Up Group - a SIP NOTIFY request from the Pick-Up Service indicating dialog state 'confirmed' in the body.

F.3.3.3 Exceptional Procedures

The exceptional procedures should be as specified in RFC 3891.

F.3.4 Actions at the Pick-UP Group UAs other than the Target/Picking-Up UA

F.3.4.1 Normal Procedures

If one alerting call is either picked-up successfully or answered, the Pick-UP Service will send a SIP NOTIFY in ‘early’ state request to all other UAs of the Pick-UP Group indicating more calls in the queue for the group. This is so the next call in the queue can be displayed properly at the alerting clients.

If an alerting call is terminated before answered the Pick-UP Service will send a SIP NOTIFY request to all other UAs of the Pick-UP Group indicating dialog state 'terminated' in the body.

F.3.4.2 Exceptional Procedures

Not applicable.

F.3.5 Actions at the Target UA

F.3.5.1 Normal Procedures

No special procedures, the Target UA will receive a SIP CANCEL request when the call is picked-up elsewhere.

F.3.6 Exceptional Procedures

Not applicable.

F.3.7 Message Flows for Group Call Pick-UP

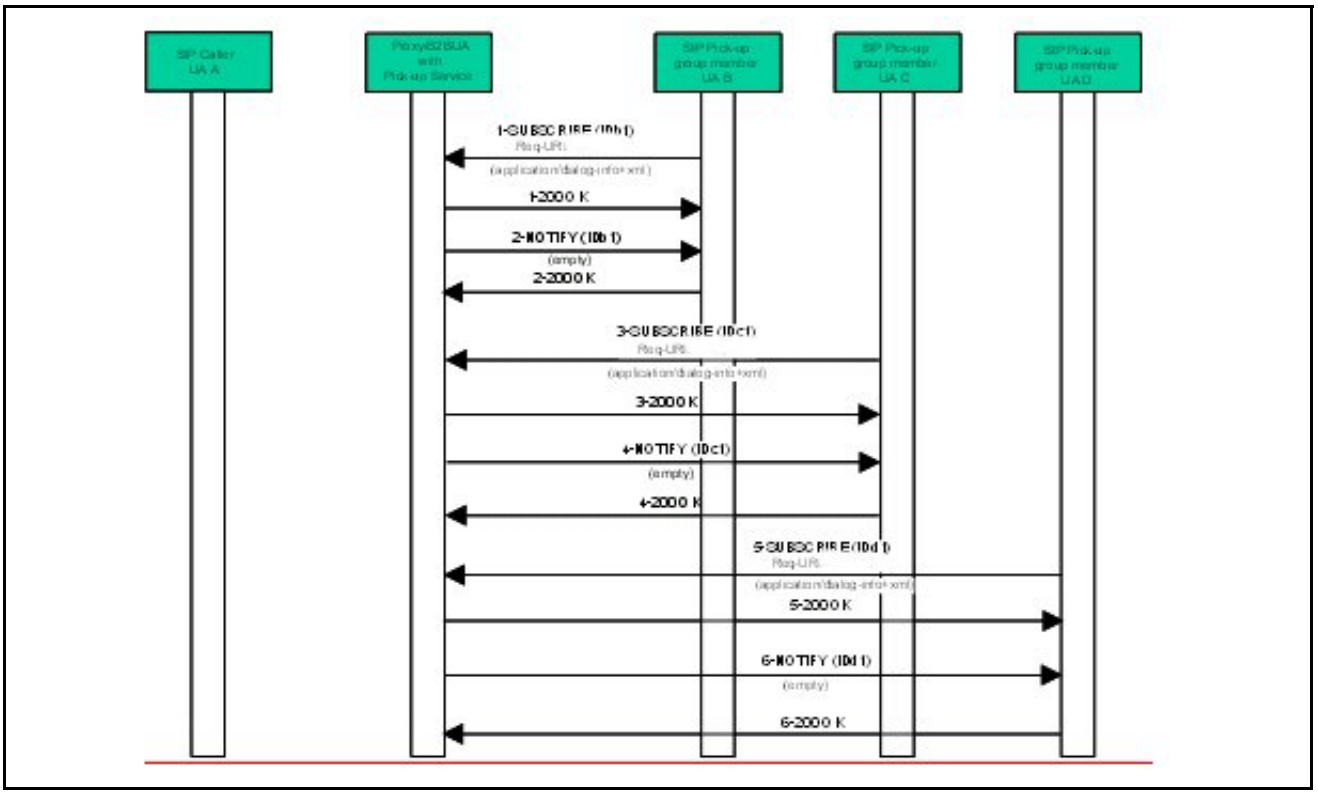


Figure F.1 Flow for Subscription to a Pick-UP Service

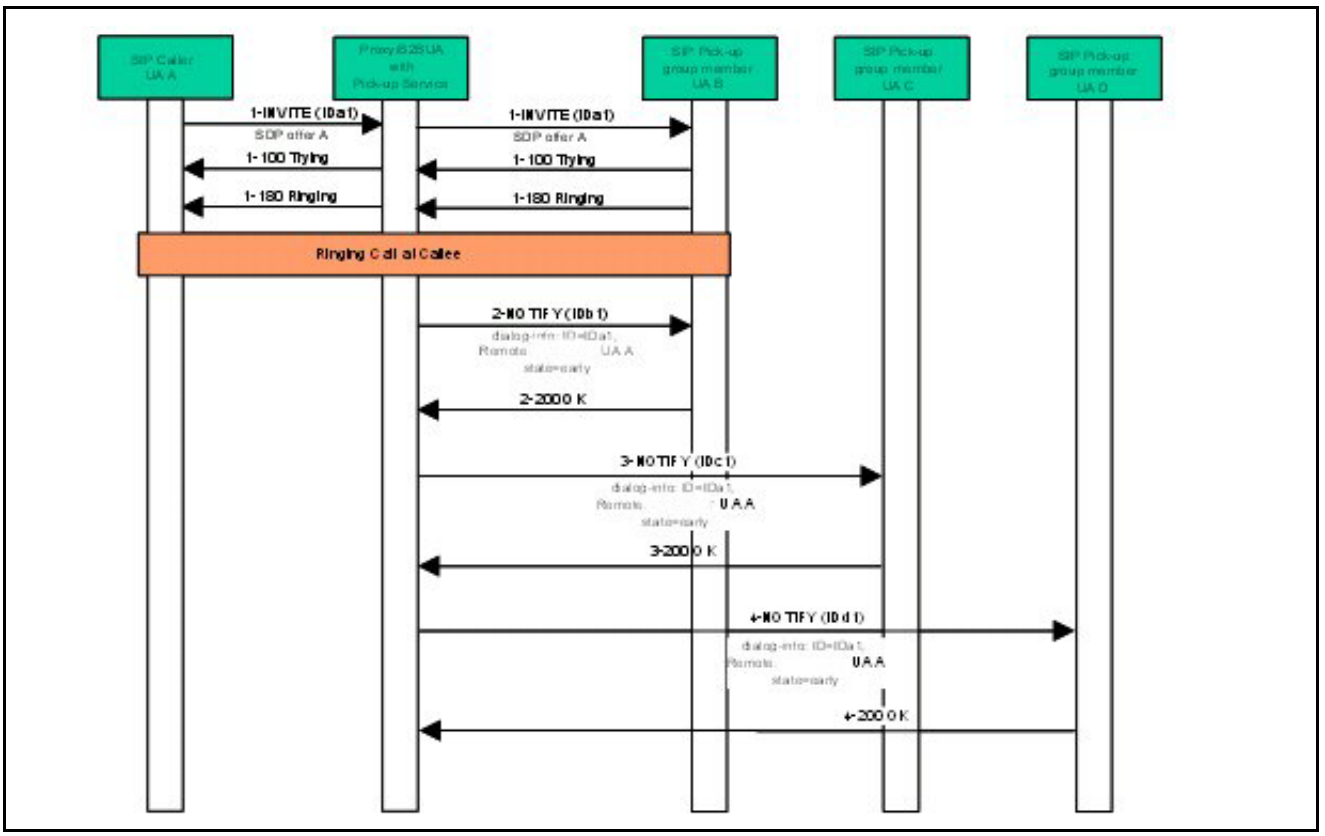


Figure F.2 Flow for Call to Pick-UP Group Member and Notification of Other Members

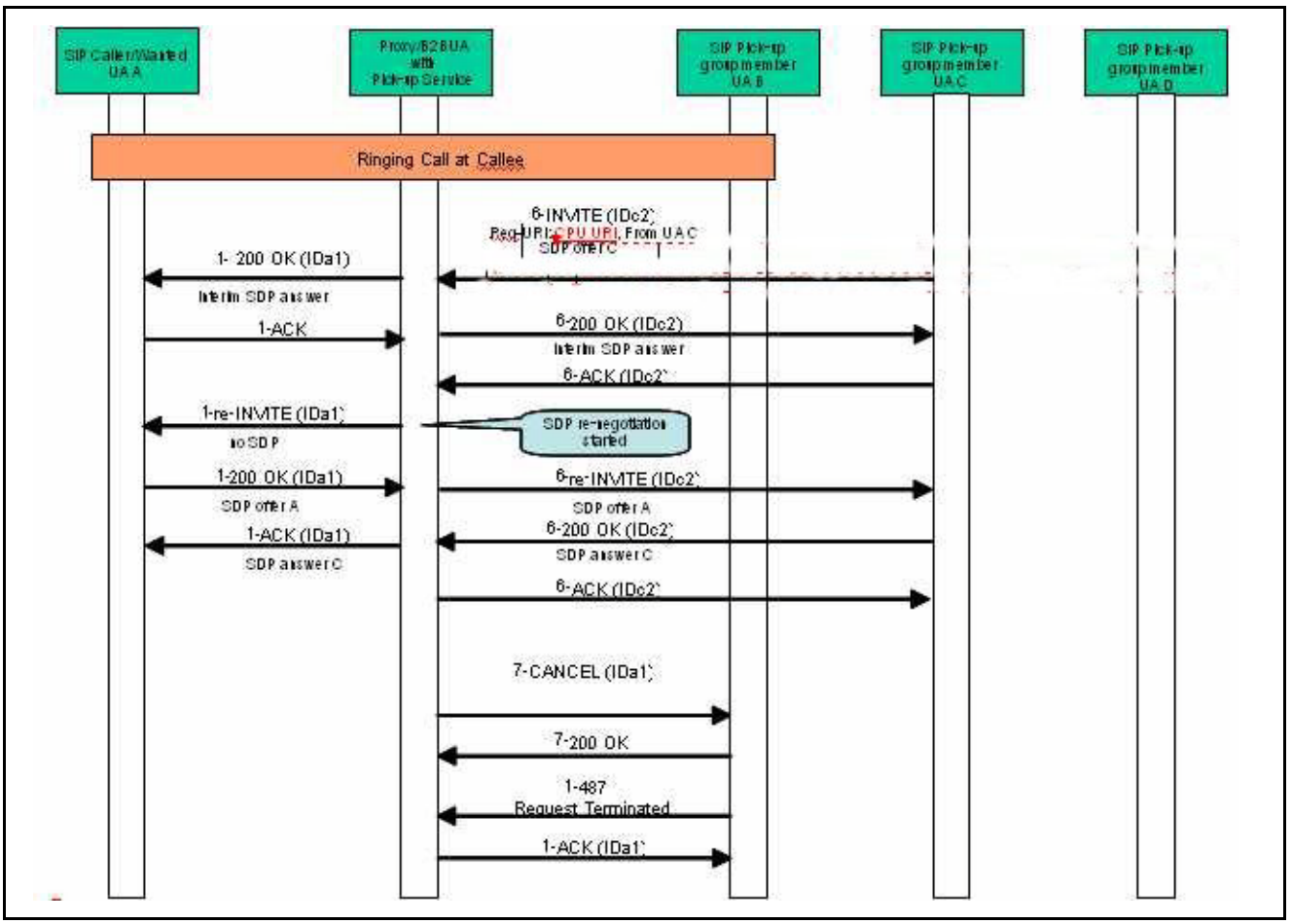


Figure F.3 Flow for Pick-UP by a Pick-UP Group Member

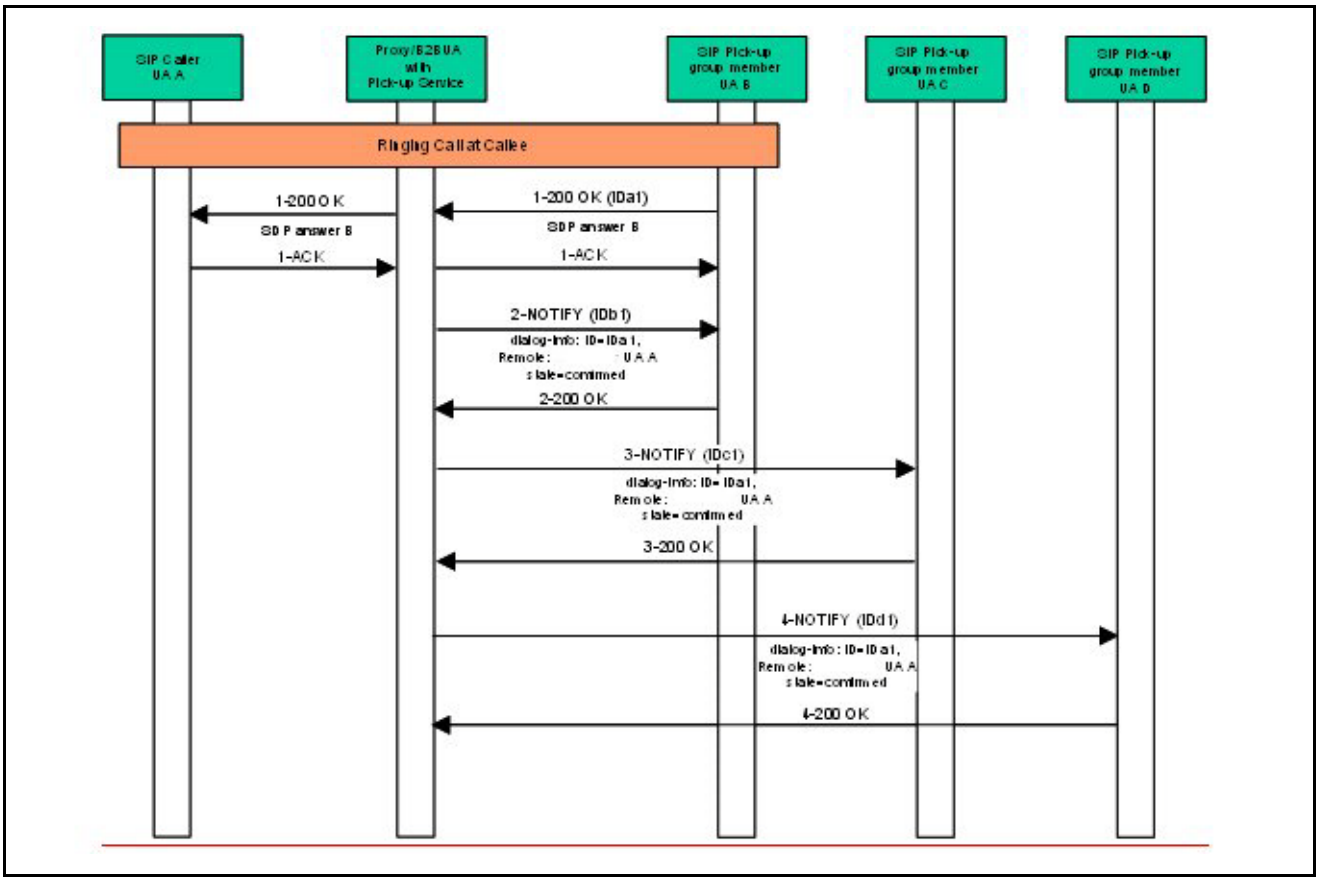


Figure F.4 Flow for Notification of Other Members About Pick-UP (Continuation of Figure F.3)

G SIP—Diversion

G.1 Actions at the Diverting OpenScape Voice

G.1.1 Normal Procedures

On receipt of a SIP INVITE request the Diverting OpenScape Voice will query the location service about Request Targets for the AoR in Request URI according to section 16.5 of RFC 3261. If the SIP INVITE request is to be retargeted based on the information received from the location service, the Diverting OpenScape Voice will construct a Diversion: header field according to SIP DIVERSION and will forward it together with the SIP INVITE request. The Diversion header field will contain the Request-URI of the request prior to the diversion, usually the AoR of the served user.

Based on the registration information at the location service a query might deliver either contact addresses or another Diverted-to AoR. If the Request Target appears to be such an AoR and if this AoR resides in the same domain that the OpenScape Voice is responsible for, the determination of Request Targets will be repeated recursively until a list of contact URIs is available. If the OpenScape Voice is not responsible for the received AoR, the request will be forwarded as specified in section 16.6 of RFC 3261. Upon retargeting the OpenScape Voice will generate a Diversion: header field or in case of multiple Diversion generate and append an additional Diversion: header field for each instance of diversion. The element counter will be set to 1 for each newly included Diversion: header field. The Diverting OpenScape Voice may change the content of the Diversion: header field as long as the content resolves to the same user, e.g. 1234@192.168.1.1 to thomas@siemens-enterprise.com.

This specification defines only the usage of the diversion-reason values "deflection", "user-busy", "unconditional" and "no-answer". Other values should not be sent by equipment conformant to this specification.

On receipt of a 302 "Moved Temporarily" response to a SIP INVITE request, a OpenScape Voice will either relay this response towards the Calling UA or intercept this response. In case of interception the OpenScape Voice will determine the new Request Targets for Call Diversion from the Contact: header field in the 302 "Moved Temporarily" response and proceed as specified above. If one or more Diversion: header fields are included in the 302 "Moved Temporarily" response the OpenScape Voice will forward it unchanged together with the SIP INVITE request, unless e.g. configuration data prohibit sending of the Diversion: header fields.

If a Diversion: header field is not included in the 302 "Moved Temporarily" response the OpenScape Voice may generate and include it in the SIP INVITE request. Using reason=unconditional seems to be most adequate in this case, unless a 180 "Ringing" provisional response was received. In the latter case reason=user-busy is appropriate.

In case of multiple diversions, the Diverting OpenScape Voice may as well re-introduce additional Diversion: header fields that were present in a SIP INVITE request but were not received in a subsequent 302 "Moved Temporarily" response. The number of re-introduced Diversion: header fields may be constrained by implementation-specific limits.

The OpenScape Voice may send a 181 "Call Is Being Forwarded" provisional response towards the Calling UAC in case that Call Diversion is performed at the Diverting Proxy/B2BUA. The 181 "Call Is Being Forwarded" provisional response may contain a P-Asserted-Identity: header field with the identity of the forwarding party.

G.1.2 Exceptional Procedures

In addition to the procedures as specified in RFC 3261 the following applies.

Receipt of a 302 "Moved Temporarily" with an undefined or unknown Diversion: header field element or value e.g. diversion-reason value "time-of-day" should not cause a protocol error. The Diversion: header field should be copied unchanged into a resulting new SIP INVITE request.

G.2 Actions at the Served UA

This specifies procedures for UAs that are capable of invoking call diversion locally.

G.2.1 Normal Procedures

A Served UA shall follow procedures as specified in RFC 3261 and send a 302 "Moved Temporarily" response to the SIP INVITE request. The Served UA should generate a Diversion: header field to be sent with the 302 "Moved Temporarily" response. In case of a gateway, the Diversion: header field may be generated based on information received from the other network.

This specification defines only the usage of the diversion-reason values "deflection", "user-busy", "unconditional" and "no-answer". Other values should not be sent by equipment conformant to this specification.

If a Diversion: header field was already received in the SIP INVITE request, it should be copied into the 302 "Moved Temporarily" response and the newly created Diversion: header field should be appended.

G.2.2 Exceptional Procedures

Receipt of a SIP INVITE request with an undefined or unknown Diversion: header field element or value e.g. diversion-reason value "time-of-day" should not cause a protocol error. An implementation may choose to ignore the unknown content or provide e.g. some default display to a user. The received Diversion: header field should be copied unchanged into an eventual 302 "Moved Temporarily" response.

G.3 Actions at the Diverted-To UA

G.3.1 Normal Procedures

On receipt of one or more Diversion: header fields, the Diverted-To UA should present the included information to the User or in case of a gateway it should signal into the other network that diversion takes place. The Diversion: header field contains an URI provided by the diverting UA and may contain a display name that is associated with the diverting user.

In case of multiple Diversion: header fields, the last one refers to the latest diversion.

Note: In case of an application, e.g. voice mail, this information may trigger application specific behaviour, e.g. selection of the correct voice mail box. However such details are out of the scope of this specification.

G.3.2 Exceptional Procedures

Receipt of an undefined or unknown Diversion: header field element or value e.g. diversion-reason value "time-of-day" should NOT cause a protocol error. An implementation may choose to ignore the unknown content or provide e.g. some default display to a user.

G.4 Actions at the Calling UA

G.4.1 Normal Procedures

On receipt of one or more a Diversion: header fields in a 302 "Moved Temporarily" response, the Calling UA should generate a new INVITE request unless it is a gateway that can signal the intent of the 302 response into the other network. If the Calling UA generates a new INVITE request, it should send the SIP INVITE request towards the URI in the Contact header field received in the 302 "Moved Temporarily" response and should present the received information to the User. All received Diversion: header fields should be included in the SIP INVITE request.

If the calling UA is co-located with a gateway, the sending of the SIP INVITE request on receipt of 302 "Moved Temporarily" may be omitted based on the capabilities of the protocol for which the gateway provides interworking. Further details of the interworking function are out of the scope of this document.

If the calling UA is co-located with a gateway, the initial SIP INVITE request may already contain a Diversion: header field based on the capabilities of the protocol for which the gateway provides interworking. The element counter may be set to a value of 1 or greater based on the number of diversions that have already occurred. Further details of the interworking function are out of the scope of this document.

On receipt of a 181 "Call Is Being Forwarded" provisional response, the Calling UA shall act as specified in RFC 3261 and should present this information to the User or in case of a gateway, the gateway SHOULD send forwarding information into the other network. The 181 "Call Is Being Forwarded" provisional response contains information from the Served UA and may contain a P-Asserted-Identity indicating the identity of the Served User.

Any P-Asserted-Identity header field in a subsequent response to the INVITE request identifies the diverted-to user.

G.4.2 Exceptional Procedures

Receipt of an undefined or unknown Diversion: header field element or value e.g. diversion-reason value "time-of-day" should NOT cause a protocol error. An implementation MAY choose to ignore the unknown content or provide e.g. some default display to a user.

H Conferencing

H.1 Procedures for a Centralized Conference

Figure H.1 shows an example in which a single user device (e.g., phone) contains the media client, call control client, notification client and floor control client. The media mixer is at a media server, the conference focus and notification server are in the OpenScape Voice, and the floor control server is located in an application. MGCP and SIP is used for the vertical interface between the conference focus and the media mixer and CSTA is a candidate for use between the floor control server and the conference focus / notification server. SIP is used for call control and notification but the floor control protocol is outside the scope of this document.

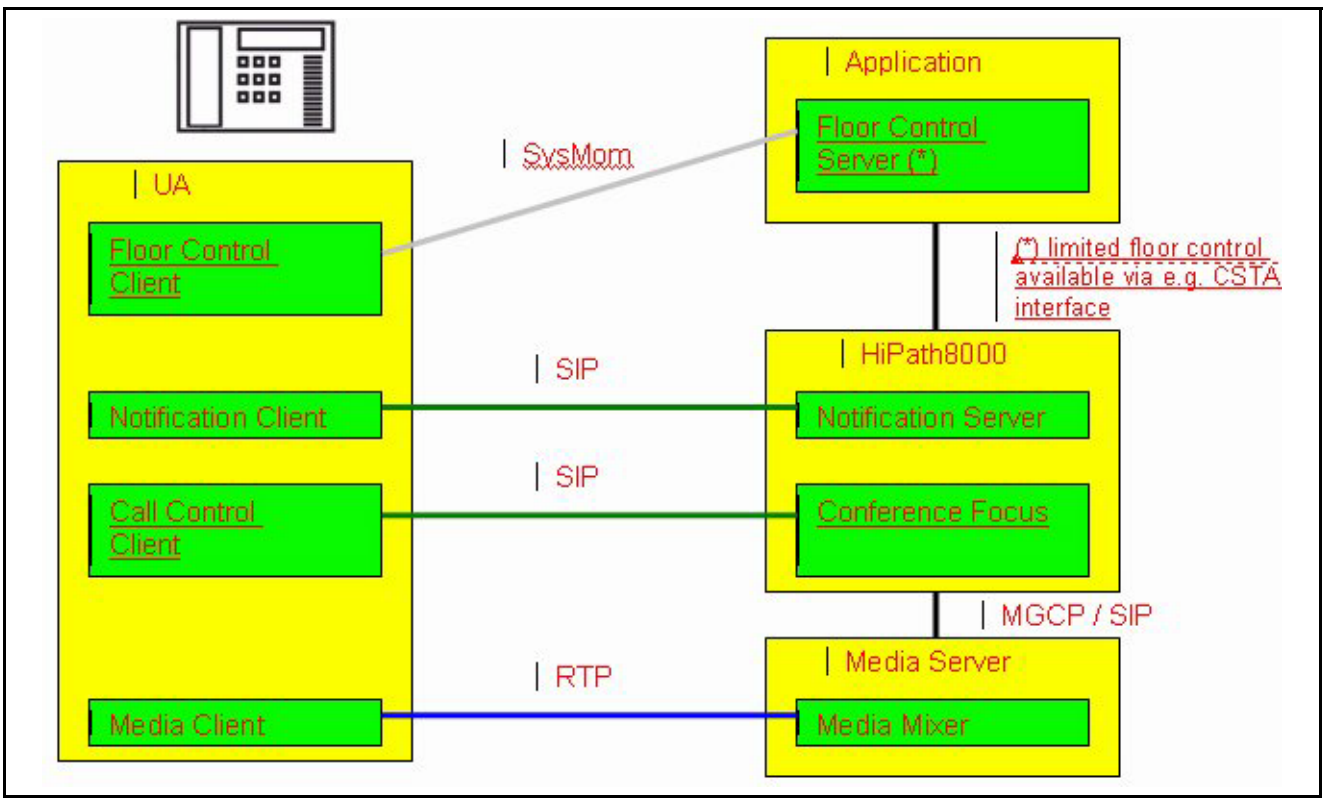


Figure H.1 Centralized Conference

In the following, only the procedure for the SIP entities involved in a centralized conference are specified. The procedures for the Conference Server are independent from the Conference Server Location. The Conference Server may e.g. be co-located within a OpenScape Voice or within an Application Server (or even within a Media Server).

A centralized conference is created by a Conference Controller UA by using the Conference Factory URI.

Note: Generally in SIP networks other methods to create a centralized conference are conceivable. However, in a first step for OpenScape Voice SIP networks this is the approach for creating a centralized conference.

Therefore it is assumed that a Conference Server has allocated and published the Conference Factory DN. In order to create a centralized conference the Conference Controller dials a Conference Factory URI (Sect. 8.8.1), which is provisioned on his UA (e.g. a 'create conference' button is pre-configured on his terminal device or the Conference Factory URI can be discovered using other means).

H.2 Actions at a Conference Controller / Conference Participant UA

For a central conference the SIP Conference Controller / Conference Participant UAs SHALL support

- SIP according to RFC 3261
- SDP according to RFC 3264 and RFC 4566
- the UA capabilities in SIP according toRFC 3840, i.e. receiving the 'isfocus' parameter,
- REFER Method according to RFC 3515.

In addition the SIP UAs compliant to this specification MAY optionally support

- SIP-Specific Event Notification according to RFC 3265 as subscriber,
- SIP conferencing events package according to RFC 4575 as subscriber,
- the Allow-Event: header indicating 'conference' according to RFC 3265 and RFC 4575,
- the Accept: header indicating support of the body application/conference-info+xml RFC 4575 in SIP SUBSCRIBE requests,
- sending the SIP Referred-by: header according to RFC 3892.

In the following the SIP procedures for the Call Control Client and the Notification Client within the Conference Controller UA are specified.

H.2.1 Creation of the Centralized Conference

H.2.1.1 Normal Procedures

In order to establish a centralized conference the Conference Controller UA SHALL send a SIP INVITE request according RFC 3261 to the Conference Factory URI. The SDP handling SHALL be according to the offer-answer model in RFC 3264.

The SIP INVITE request SHALL contain the following headers (optional usage of other headers is not prohibited):

To: Conference Factory URI,

From: URI of the Conference Controller, From-Tag,

Contact: URI of the Conference Controller UA,

Content-Type: application/sdp,

Authorization: as specified in RFC 2617 and RFC 3261,

Call-ID, Via, Max-Forwards, CSeq, Allow, Content-Length, Supported according to RFC 3261.

Then in successful case the SIP session SHALL be established as for a two party call (i.e. outgoing call establishment) and a SIP 200 OK response from the Conference Focus as detailed in Sect. H.3.1.1 below is received.

Owing to receipt of 'isfocus' the Call Control Client SHOULD indicate to the user that he is in conference. In addition, if Allow-Event: header indicating 'conference' is also received and the Conference Controller supports the conference event package (RFC 4575), the Call Control Client SHALL advise its Notification Client to subscribe to the conference event package. The appropriate procedures are specified below in Sect. H.3.2.

H.2.1.2 Exceptional Procedures

If the SIP INVITE request is rejected, the failure handling SHALL be as described in RFC 3261 section 14.1.

H.2.2 A Conference Participant Adds Another Participant via REFER

H.2.2.1 Normal Procedures

A SIP user which is participant in a running conference may add another user to the conference using a REFER request. In the following it is assumed that a confirmed dialog1 between the Conference Participant UA A and the Conference Focus and another confirmed dialog2 between the Conference Participant and the new participant UA B (i.e. user B to be added to the conference) already exists.

In this architecture there is a B2BUA (OpenScape Voice) between UA A and UA B.

1. Procedures at UA A

   Depending on local policy, UA A MAY put on hold either UA B or the Conference Focus or both (see Sect. H.6) below for Interactions with SIP Media Hold) previously. Then UA A SHALL send a SIP REFER request with a Refer-To: header RFC 3515 on dialog2 towards UA B

   Therefore UA A SHALL copy the Conference URI (including any URI parameters) from the Contact: header field as received from the Conference Focus into the Refer-to: header field, but NOT any received Contact header parameters (such as 'isfocus').

   The SIP REFER request SHALL contain the following headers:

   Request URI: Contact URI of user B

   To: URI of user B, To-tag (as in dialog2)

   From: URI of user A, From-Tag (as in dialog 2),

   Refer-to: Conference URI (URI of the Focus) [method=INVITE]

   Contact: Contact URI of user A (as in dialog2)

   Referred-by: URI of user A, RFC 3892 (optional)

   Call-ID, Via, Max-Forwards, CSeq, Content-Length according to RFC 3515

Note: The To: and the From: header field URIs are not necessarily those of user A and user B respectively, but they reflect the URIs used during dialog establishment.

On receipt of a 202-Accepted response to the SIP REFER request on dialog2 or SIP NOTIFY requests indicating "Trying" and/or "Ringing", UA A SHOULD take no action. On receipt of a SIP NOTIFY request indicating "200 OK" on

dialog2, UA A SHALL send SIP BYE request towards UA B and both the REFER-initiated implicit subscription and dialog2 SHALL be terminated. However, this is standard REFER RFC 3265 behaviour and is repeated here just for completeness. After terminating dialog2 UA A SHALL send a SIP reINVITE on dialog1 to re-establish the previously held connection to the Media Server.

2. Procedures at UA B

   On receipt of a SIP re-INVITE request with the Contact: header field containing the capability parameter 'isfocus' (and possibly including a Referred-By: header indicating the URI of user A) and a SDP offer from the B2BUA, UA B

   - SHALL return a SIP 200 OK response with the SDP answer,
   - SHOULD indicate to the user that the dialog to user A is replaced and it is now participating in a conference,

   and if in addition the Allow-Event: header indicating 'conference' is received in the SIP re INVITE request and UA B supports the conference event package RFC 4575, then UA B's Call Control Client SHALL instruct (e.g. via an internal message which is out of scope of this specification) its Notification Client to subscribe to the conference event package. The appropriate procedures are as described below in Sect. H.3.

3. Procedures at a OpenScape Voice

   On receipt of a SIP REFER request indicating method=INVITE in the context of a dialog for an existing call, OpenScape Voice SHALL act in accordance with RFC 3515 and RFC 3892.

   The Refer-To: header contains the Conference URI which is recognized by the OpenScape Voice performing the role of the Conference Focus. OpenScape Voice then establishes a connection at the Conference Server (Media Server) for UA B.

   Once a connection has been created at the Conference Server (Media Server), the OpenScape Voice SHALL send a SIP re-INVITE request to UA B. In the Contact: header the Conference URI and the capability parameter 'isfocus' RFC 3840 SHALL be indicated. In addition the SDP offer indication of the Conference Focus (i.e. the Media server) SHALL be included.

   On receipt of a SIP 200 OK response from UA B and a SDP answer indication, the OpenScape Voice SHALL send this SDP answer indication to the Media Server and return a SIP ACK response to UA B.

   Towards UA A the OpenScape Voice SHALL send a 202-Accepted response to the SIP REFER request and appropriate SIP NOTIFY requests (indicating "Trying", "Ringing", "200 OK, 4xx responses) according to RFC 3515. On receipt of the SIP NOTIFY indicating "200 OK" UA A SHALL send a SIP BYE

request to release its call to UA B and shall then send a SIP reINVITE to OpenScape Voice to re-establish the previously held connection with the Media Server. OpenScape Voice SHALL return a SIP 200 OK to UA A.

H.2.2.2 Exceptional Procedures

1. Procedures at UA A

   If either

   - the REFER request on dialog2 is rejected, or
   - a SIP NOTIFY request with a final response 4xx, 5xx or 6xx is received

   then UA A SHOULD indicate the failure to user A. Dialog2 still exists and it will be up to UA A how to proceed (e.g. reconnect user A with user B).

   Note: In the case where a conference is full, and UA A nevertheless sends a REFER request to add a new participant, then a NOTIFY request with a 486 Busy Here response will be received by UA A. In the case where addition to the conference is not authorized, a NOTIFY request with a 403 Forbidden response will be received by UA A.

   If UA B clears dialog2 before the REFER procedure is completed then UA A SHOULD indicate the failure to user A.

2. Procedures at UA B

   If the SIP REFER request cannot be accepted, UA B SHALL act in accordance with RFC 3515.

   If the SIP re-INVITE request cannot be accepted, the failure handling SHALL be as described in RFC 3261 section 14.2.

3. Procedures at OpenScape Voice

   If the SIP REFER request cannot be accepted, the OpenScape Voice SHALL act in accordance with RFC 3515.

H.2.3 Subscription to the Conference Package

H.2.3.1 Normal Procedures

On receipt of an indication to subscribe to the conference event package RFC 4575, the conference participant's Notification Client (i.e. the subscriber) SHALL send a SIP SUBSCRIBE request to the Conference Focus URI.

The SIP SUBSCRIBE request SHALL contain the following headers (optional usage of other headers is not prohibited):

To: Conference URI

From: URI of the Subscriber (e.g. Party A); From-Tag

Contact: URI of the Subscriber's UA (e.g. UA A)

Content-Length: 0

Event: conference

Subscription-State: active; expires=set to a default value

Accept: application/sdp, application/conference-info+xml, message/sipfrag

Authorization, Call-ID, Via, Max-Forwards, CSeq, Allow, Content-Length

If the subscription is successful the subscriber SHALL receive a SIP 200 OK and an initial SIP NOTIFY request according to Sect. H.3 below.

Further SIP NOTIFY requests MAY be received as long as the subscription persists and when there is a change in the conference information state.

The subscriber SHALL support the receipt of SIP NOTIFY requests containing both full as well as just deltas of the XML encoded conference information document (i.e., partial Notifies).

The conference information received within the SIP NOTIFY body MAY be used to update the UA's local display.

The subscribers SHOULD use the element <maximum-user-count> to determine whether to indicate to users the ability to add further members.

The subscription is terminated if the dialog terminates or if a Contact: header field without an 'isfocus' parameter is received. However, to terminate a subscription extraordinary, the subscriber SHALL send a SIP SUBSCRIBE request wherein the value of the Expires: header is set to NULL. Then a SIP NOTIFY request containing a Subscription-State: header with value 'terminated' is expected from Conference Notification Server.

H.2.3.2 Exceptional Procedures

On receipt of a 401 (unauthorized) or a 407 (Proxy Authentication Required), the Conference Participants UA SHALL act as specified in RFC 3265 or RFC 3261, respectively.

H.3 Actions at a Conference Server (OpenScape Voice)

The SIP Conference Server UAs compliant to this specification SHALL support

- SIP according to RFC 3261
- SDP according to RFC 3264 and RFC 4566
- SIP-Specific Event Notification according to RFC 3265 and the event package for conference state RFC 4575 and be able to act as notifier,
- the UA capabilities in SIP according to RFC 3840 as far as necessary for this specification, i.e. sending of the 'isfocus' parameter,
- the Allow-Event: header indicating 'conference' according to RFC 3265 and RFC 4575.

In addition the SIP Conference Server UAs compliant to this specification MAY optionally support

- receiving the SIP Referred-by: header according to RFC 3892.

In the following the SIP call control procedures for the Conference Focus and the Conference Notification Server are specified.

H.3.1 Creation of the Centralized Conference

H.3.1.1 Normal Procedures on the Conference Focus

On receipt of a SIP INVITE request according to Sect. H.2.1.1 the Conference Factory application SHALL route the SIP INVITE request to the Conference Focus. Details of this interface are out of scope of this specification. However, the Request URI received by the Conference Focus MAY already contain the Conference URI. Then the Conference Focus SHALL invoke authorisation procedures and MAY invoke authentication procedures for the INVITE initiated UA/user.

After successful authorisation and possibly authentication the Conference Focus SHALL act in accordance with RFC 3261 and RFC 3264 and return a SIP 200 OK response and a SDP answer indication to the Conference Controller UA. The 200 OK response SHALL contain the following headers:

To: URI as received, To-Tag

From: as received from Conference Controller UA; From-Tag

Contact: Conference URI; isfocus

Allow-Event: conference,

Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length

The Conference Focus SHALL inform via an internal message the Conference Notification Server about the established conference. However, this indication is out of scope of this specification.

H.3.1.2 Exceptional Procedures on the Conference Focus

If the SIP INVITE request is not acceptable, the Conference Focus SHALL send a 4xx response, as specified in RFC 3261, to the conference controller UA.

H.3.2 Subscription to the Conference Package

H.3.2.1 Normal Procedures on the Conference Notification Server (OpenScape Voice)

Upon receipt of a SIP SUBSCRIBE request for the conference event package RFC 4575, the Conference Notification Server SHALL verify that the Contact URI in the SUBSCRIBE request equates to the UA in one of the conference INVITE-initiated dialogs. If the subscription checks are satisfactory according the SIP authentication procedures the Conference Notification Server SHALL return a SIP 200 OK and the initial SIP NOTIFY request to the appropriate UAs Notification Client.

The 200 OK response SHALL contain the following headers:

To: URI as received, To-Tag

From: as received from Conference Controller UA; From-Tag

Contact: Conference URI

Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length.

The Request URI of the SIP NOTIFY request SHALL be set to the Contact URI of the conference participant (as received in the SIP SUBSCRIBE request) and the following headers SHALL be used (optional usage of other headers is not prohibited):

To: URI of the Subscriber, To-Tag (as From-Tag of the SUBSCRIBE)

From: Conference URI; From-Tag (as sent in the To-Tag in 200 OK above)

Contact: Conference URI

Event: conference

Subscription-State: active; expires (default value)

Content-Type: application/conference-info+xml

Authorization, Call-ID, Via, Max-Forwards, CSeq, Allow, Accept, Content-Length

The body of the SIP NOTIFY request SHALL contain the XML encoded conference information document. This conference information document describes the state of the conference.

A conference notification server in conformance with this specification SHALL support as a minimum the <conference-description> element, the <users> element and <maximum-user-count> as specified in RFC 4575.

The <conference-description> element SHALL contain as a minimum element <display-text>, indicating the endpoint type and the line identity.

The <users> element SHALL contain a list of <user> elements. Each <user> element SHALL indicate the user identity or an indication of anonymous.

The <maximum-user-count> element SHALL provide a value representing the overall number of users allowed to join the conference. This value is set by an administrator and can reflect the local conference policy. For a local conference the conference notification server SHOULD set the value of this element to 3, if it does not allow other members to add further members (even if it allows the conference controller to create conferences with more than 3 members).

Whenever there is a change in the state in any conference information (e.g. other participants were added to the conference) further SIP NOTIFY requests SHALL be sent to the Notification Client of the subscribed participant(s).

The initial SIP NOTIFY requests sent to the participants following subscription or refresh SHALL contain the full XML encoded conference information document, whereas all other SIP NOTIFY requests sent to the conference participants MAY contain just deltas of the XML encoded conference information document.

H.3.2.2 Exceptional Procedures on the Conference Notification Server

If the SIP SUBSCRIBE request is not acceptable, the Conference Notification server SHALL send a 4xx response.

H.4 Procedures for termination of a Centralized Conference

A central conference is terminated when all conference participants left the conference due to sending SIP BYE requests to the Conference Focus.

H.5 Procedures for Cascaded Centralized and Local Conferences

A centralized conference MAY be enhanced due to establishing of a local conference on a conference participant UA. This cascading of conferences is uncontrollable by the Conference Focus, and can be performed even when Conference Focus limits the conference participants to a maximum number.

H.6 Feature Interactions

H.6.1 Call Hold

The procedures for SIP Media Hold on the Holding UA as well as on the Held UA SHALL be as defined in Sect. 8.2.

However, the Media Client on the Conference Focus within an OpenScape Voice SIP network SHOULD not be provided with Music on Hold. Therefore OpenScape Voice and a conference aware UA must NOT introduce a media server that provides music on hold into a conference. A UA must NOT knowingly send music on hold to a conference.

I Directed Call Pick-UP

I.1 Actions at the Picking-Up UA

I.1.1 Normal Procedures

On receipt of a user indication to Pick-UP a call, the picking-up UA SHALL subscribe to the dialog state of the Target UA using the dialog-event package in accordance with RFC 4235 at the URI of the Target user. The URI of the Target user is supplied by the picking-up user.

Note: This should be the AoR of the target user.

The SIP SUBSCRIBE request SHALL be in accordance with RFC 3265 and it SHALL contain an Expires: header field with value "0". In addition the SIP SUBSCRIBE request SHALL contain the X-Siemens-Call-Type: header field indicating "DIR_PICK" .

The SIP SUBSCRIBE request SHALL contain the following headers:

- To: URI of the Target user
- From: URI of picking-up user, From-Tag
- Contact: URI of picking-up UA
- Event: dialog
- Expires: "0"
- Accept: application/dialog-info+xml
- X-Siemens-Call-Type : "DIR_PICK"
- Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length

On receipt of a 200 OK response to the SIP SUBSCRIBE request and SIP NOTIFY request from the Target UA, the picking-up UA SHALL return a SIP 200 OK to the Target UA and the subscription is completed successfully. However, this is mentioned just for completeness and is standard behaviour in accordance with RFC 3265 and RFC 4235

Note: Subscribe requests may fork and thus generate multiple dialogs and multiple notifications.

The picking-up UA SHALL use information as received in the body of the SIP NOTIFY request to select a suitable dialog at the Wanted UA. This means that the dialog needs to be in a suitable state (i.e., either ringing or directed parked or manual hold) and for which sufficient information has been provided.

Note: SIP NOTIFY request should only contain dialogs that are in a suitable state to be picked up, but because of working with third party devices that do not conform to this specification, there may be dialogs in other states.

Then the picking-up UA SHALL establish a call to the Wanted UA in accordance with RFC 3261. The SIP INVITE SHALL indicate in the Request URI the contact URI as received in the target uri of the remote element of the SIP NOTIFY request and in the Replaces: header field the Call-ID as received in the call id element of the SIP NOTIFY request. If the local-tag and remote-tag elements were received in the dialog-info package of the SIP NOTIFY request, then the to-tag and from-tag SHALL be provided in the Replaces: header field. The direction element of the dialog-info package SHALL be taken into consideration.

Note: From the three elements local-tag, remote-tag and direction it is apparent which one is the to-tag and which one is the from-tag. If direction="recipient" is received, the element local-tag contains the to-tag and the element remote-tag contains the from-tag; else, if direction="initiator" is received, the element local-tag contains the from-tag and the element remote-tag contains the to-tag.

In addition the SIP INVITE request SHALL contain the X-Siemens-Call-Type: header field indicating "DIR_PICK" if received in the SIP NOTIFY request and if supported by the picking-up UA.

The SIP INVITE request SHALL contain the following headers:

- To: Contact URI (as received in the NOTIFY)
- From: URI of picking-up user, From-Tag
- Contact: URI of picking-up UA
- Require: replaces
- Replaces: Call-ID; to-tag; from-tag (as received in the dialog package of the NOTIFY request)
- Content-Type: application/sdp
- X-Siemens-Call-Type : "DIR_PICK"
- Authorization, Call-ID, Via, Max-Forwards, CSeq, Content-Length, Allow, Supported according to RFC 3261.

The SDP offer information RFC 4566 from the picking-up UA SHALL be sent in accordance with RFC 3264.

In case that a ringing call is to be picked up, the picking-up UA SHALL include the early-only-Flag in the Replaces: header field according to RFC 3891. This avoids picking up the confirmed dialog, if the Target User has answered in the meantime.

Note: In case of picking up a confirmed dialog (directed parked or manual hold dialog) there might be collision cases, when multiple UA try to pickup a call. However, countermeasures are currently out of scope of this specification.

On receipt of a 200 OK response to its INVITE request, the picking-up UA SHALL act as specified in RFC 3261.

In case that the B2BUA just provided answer information, the picking-up UA may receive a re-INVITE request with SDP offer information (from the Wanted UA) immediately afterwards. In this case the picking-up UA SHALL send the SDP answer information in the 200 OK response and act according to RFC 3261 or RFC 3311 respectively.

In both cases the Directed Call Pick-UP was successful.

I.1.2 Alternate Procedures

The procedures presented in (a) above are followed except that the INVITE from the Picking-Up UA does not include an SDP offer. The SDP offer from the Wanted UA is passed to the Picking-Up UA in the 200 OK and the SDP answer from the Picking-UP UA received in the ACK is passed to the Wanted UA in the 200 OK for the INVITE, completing the Call Pick-UP.

I.1.3 Exceptional Procedures

On receipt of a 4xx / 5xx/6xx response to the SIP INVITE or SIP SUBSCRIBE request, the picking-up UA SHALL act as specified in RFC 3265 and RFC 3261 and SHOULD indicate failure of Directed Call Pick-UP to the picking-up user unless means of recovery are available.

I.2 Actions at the OpenScape Voice Server (Pick-UP B2BUA)

I.2.1 Normal Procedures for the OpenScape Voice Server

On receipt of a SUBSCRIBE request (see Section I.1.1, “Normal Procedures”) with the X-Siemens-Call-Type: header field indicating "DIR_PICK", the Pick-UP B2BUA acts as notifier in accordance with RFC 3265.

Acting as notifier, the Pick-UP B2BUA SHALL send a 200 OK response to the received SIP SUBSCRIBE request and send a SIP NOTIFY request to the subscriber in accordance with RFC 3265. The SIP SUBSCRIBE request SHALL include the X-Siemens-Call-Type: header field with "DIR_PICK" and indicate value "0" in the Expires: header field.

The body of the SIP NOTIFY SHALL contain the XML encoded dialog state information of the incoming (early dialog), the directed parked or the manual hold call according to RFC 4235.

The following is a template example for the XML-body for a dialog in confirmed call state:

The following is a template example for the XML-body for a dialog in confirmed call state:

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
             version="Version number"
             state="full"
             entity="URI of Target User">
  <dialog id="id value"
          call-id="call-id of the dialog to be picked-up"
          local-tag= "local-tag value for dialog1" < -- optional, if stored in B2BUA --;>
          remote-tag= "remote-tag value for dialog1" < -- optional, if stored in B2BUA --;>
          direction="recipient"  < -- could be "initiator", depending who initiated the call
                                                    in case of manual hold call --;>
    <state>confirmed</state>
    <local>
      <identity display="Display Name">"AoR of Target user" </identity>
      <target uri="Contact URI Target UA" </target>
    </local>
    <remote>
```

```
      <identity display="Display Name">"AoR of Wanted/Calling user" </identity>
      <target uri="Contact URI of B2BUA" </target>
    </remote>
  </dialog>
</dialog-info>
```

The element "target uri" of the remote element SHALL indicate Contact URI of the Wanted user from the viewpoint of the Target UA.

On receipt of a SIP INVITE request including the X-Siemens-Call-Type: header field with "DIR_PICK" and a Replaces: header field from the picking-up UA, the Pick-UP B2BUA SHALL terminate the SIP INVITE request.

When the Pick-UP B2BUA terminates the SIP INVITE request with the Replaces: header field, the Pick-UP B2BUA SHALL send to the picking-up UA a 200 OK response with a B2BUA generated answer information.

Then the Pick-UP B2BUA SHALL co-ordinate both call legs further onwards and SHALL act as follows:

- In case that the dialog to be picked up is in early dialog state, the Pick-UP B2BUA SHALL send to the Wanted UA a 200 OK response including a X-Siemens-Call-Type: header field indicating "DIR_PICK" and a B2BUA generated answer information. On receipt of the ACK acknowledgment the Pick-UP B2BUA SHALL send to the Wanted UA a re-INVITE request with no SDP information and on receipt of the 200 OK response with the SDP offer information from the Wanted UA, the Pick-UP B2BUA SHALL send this SDP offer information in a re-INVITE request to the picking-up UA. Upon receipt of the 200 OK response with the SDP answer information from the picking-up UA the Pick-UP B2BUA SHALL send this SDP answer information in the ACK acknowledgment to the Wanted UA.

  If the Directed Call Pick-UP was successful (i.e. picking-up UA and Wanted UA are linked together), the Pick-UP B2BUA SHALL send a SIP CANCEL request to the Target UA. The CANCEL request SHALL include the X-Siemens-Call-Type: header field indicating "DIR_PICK" and a Reason: header field indicating SIP; cause=200; text="Call completed elsewhere" according to RFC 3326.

- In case that the dialog to be picked up is in a confirmed dialog state (i.e., directed parked or manual hold dialog), the Pick-UP B2BUA SHALL send a re-INVITE request to the Wanted UA. The SIP re-INVITE request SHALL include the X-Siemens-Call-Type: header field indicating "DIR_PICK" and the no SDP information. On receipt of the 200 OK response with the SDP offer information from the Wanted UA, the Pick-UP B2BUA SHALL send this SDP offer information in a re-INVITE request to the picking-up UA. Upon receipt of the 200 OK response with the SDP answer information from the picking-up UA the Pick-UP B2BUA SHALL send this SDP answer information in the ACK acknowledgment to the Wanted UA.

If the directed call Pick-UP was successful (i.e. picking-up UA and Wanted UA are linked together), the Pick-UP B2BUA SHALL send a SIP BYE request to the Target UA. The SIP BYE request SHALL include the X-Siemens-Call-Type: header field indicating "DIR_PICK" and a Reason: header field indicating SIP; cause=200; text="Call completed elsewhere" according to RFC 3326.

I.2.2 Alternate Procedures for the OpenScape Voice server

The procedures presented in (a) above are followed except that the INVITE from the Picking-Up UA does not include an SDP offer. The SDP offer from the Wanted UA is passed to the Picking-Up UA in the 200 OK and the SDP answer from the Picking-UP UA received in the ACK is passed to the WantedUA in the 200 OK for the INVITE, completing the Call Pick-UP.

I.2.3 Exceptional Procedures for the Pick-UP B2BUA

If on receipt of a SIP SUBSCRIBE request with the X-Siemens-Call-Type: header field indicating value "DIR_PICK" and there is no dialog that can be picked up from the Target UA, the Pick-UP B2BUA SHALL return a SIP NOTIFY request indicating 'terminated' in the Subscription-State: header field and SHALL NOT list any dialog in the body.

On receipt of a 4xx / 5xx/6xx response to the SIP INVITE request or the SIP re-INVITE request respectively, the Pick-UP B2BUA SHALL act as specified in RFC 3261 and forward this response to the picking-up UA.

I.3 Actions at the Wanted UA

I.3.1 Normal Procedures

On receipt, in the early dialog state, of a 200 OK response to the initial SIP INVITE request including an X-Siemens-Call-Type: header field indicating "DIR_PICK" and a SDP answer information, the Wanted UA SHALL send the ACK acknowledgment. In case that the B2BUA just provided answer information, the Wanted UA may receive a re-INVITE request with no SDP information immediately afterwards. In this case the Wanted UA SHALL send the SDP offer information in the 200 OK response and act according to RFC 3261.

If the Wanted UA receives in the confirmed dialog state a re-INVITE request with no SDP information and with an X-Siemens-Call-Type: header field indicating "DIR_PICK", the Wanted UA SHALL send the SDP offer information in the 200 OK response and act according to RFC 3261.

Based on the received X-Siemens-Call-Type: header field indicating "DIR_PICK" the Wanted UA MAY indicate to the user that the dialog was picked-up.

If the Wanted UA supports the Replaces: header field according to RFC 3891 then on receipt of a SIP INVITE request from the picking-up UA including a Replaces: header field the Wanted UA MAY indicate to the user that the dialog was replaced and SHALL act as specified in RFC 3891.

If in addition the X-Siemens-Call-Type: header field indicating "DIR_PICK" was received then the Wanted UA MAY indicate to the user that the dialog was picked-up.

The SIP CANCEL request sent to the Target user, in case that an early dialog was picked up, SHALL contain the X-Siemens-Call-Type: header field indicating "DIR_PICK".

The SIP BYE request sent to the Target user, in case that a confirmed dialog was picked up, SHALL contain the X-Siemens-Call-Type: header field indicating "DIR_PICK".

I.3.2 Exceptional Procedures

On receipt of a SIP INVITE request from the picking-up UA including Replaces: header field either indicating a dialog that cannot be matched to a dialog at the Wanted UA or indicating dialog state "early" and there is no early dialog available the Wanted UA, the Wanted UA SHALL act according to RFC 3891.

I.4 Message Flow for Directed Call Pick-UP

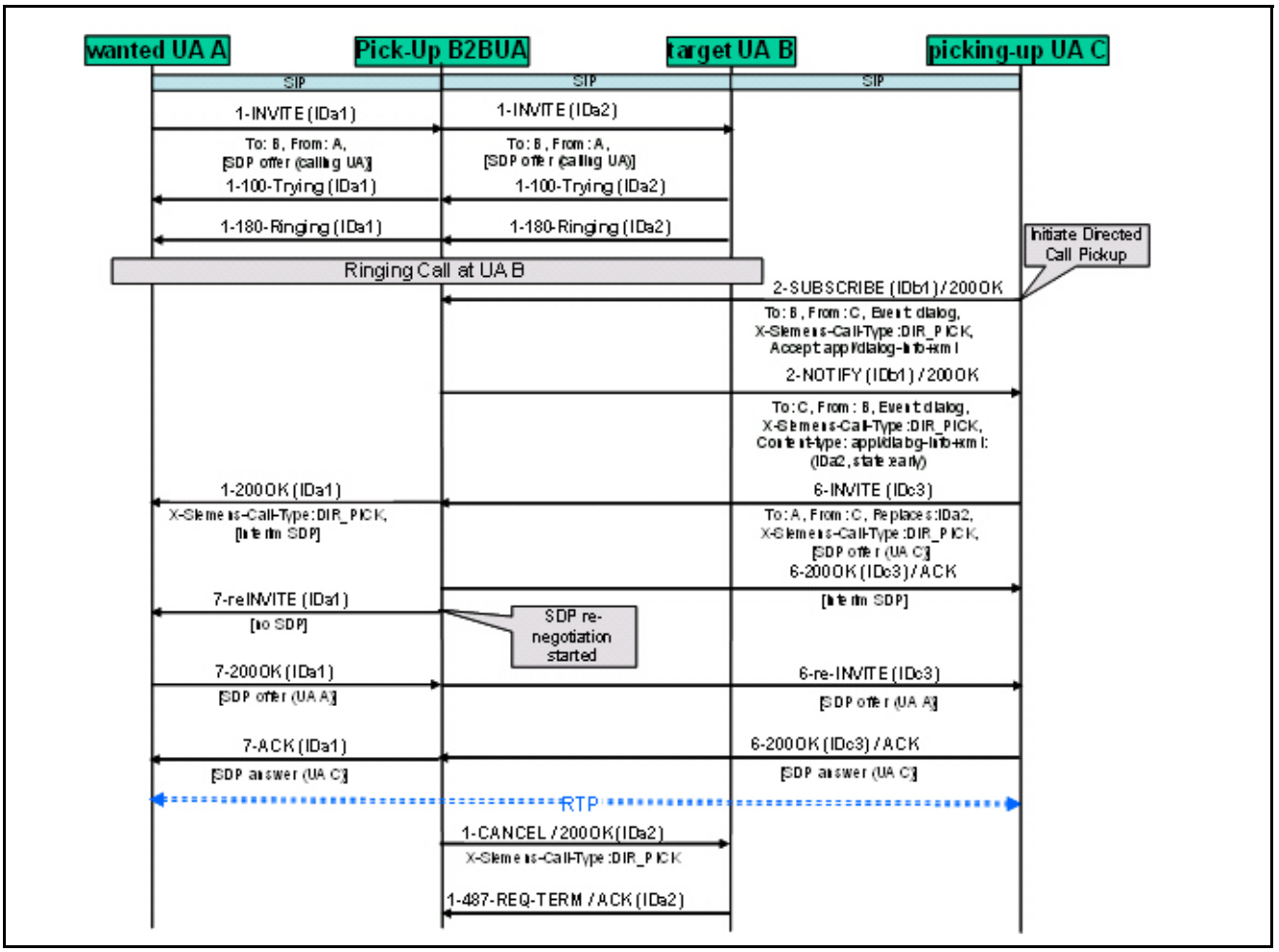


Figure I.1 Message Flow for Directed Call Pick-UP

J Quality of Service (QoS) Monitoring

This specification provides sufficient information for a Client to report QoS information to OpenScape Voice. Full details of OpenScape Voice QoS Monitoring procedures are available in other Siemens specifications that are available upon request.

Table J.1 lists the selected values which may be transported in the SIP BYE message.

Value	Connection Parameter
Media Type	MT
Date and time of the beginning of the report period	TB
Date and time of the end of the report period	TE
IP address, local	IPL
Port #, local	PTL
IP address, remote	IPR
Port #, remote	PTR
Identifier of RTP stream sent	SRCS
Identifier of RTP stream received	SRCR
Codec - Encoding/Decoding	EN.DE
Number of Silence Suppression Activations	SS
Count of good packets	PR
Maximum jitter	JI
Average round trip delay	LA
Count of lost packets	PL
Count of discarded packets	PD
Consecutive packet loss	CPL
Consecutive good packets	CGP
Packets sent (PS= Pr + PD)	PS
Octets sent	OS
Octets received	OR
Impairments from low-bit codecs and packet loss	IE
Telephone Echo Signal Attenuation in dB	TCLW

Table J.1 QoS Reports Carried in SIP BYE Requests

J.1 Procedures for an Entity Sending a BYE Request/Response

SIP clients may send BYE requests or 200 OK responses to BYE requests including an X-Siemens-RTP-stats header field [Sect. 5.11.48].

Note: The X-Siemens-RTP-stats header field currently only supports IPv4 addresses and can therefore not currently be used by IPv6 devices. The X-Siemens-RTP-stats header will be enhanced in a future release to support IPv4 and IPv6 addresses.

When sending X-Siemens-RTP-Stats in a BYE request or response, the sender must construct this header field with one or more parameters as specified below. Any combination of these parameters is possible subject to no more than one instance of each parameter type.

The X-Siemens-RTP-stats header field is defined as:

```
x-siemens-rtp-stats = "X-Siemens-RTP-stats" COLON x-
siemens-rtp-stats-params

x-siemens-rtp-stats-params = x-siemens-rtp-stats-param
*("," x-siemens-rtp-stats-param)

X-siemens-rtp-stats-param = tb / te / ipl / ... / os / or
/ generic-param
```

The generic-param RFC 3261 parameter allows for the specification of additional parameters in the future. One particular future enhancement might be a parameter to identify a particular medium in the context of a call with multiple RTP-based media (e.g., audio, main video, presentation video), based on the principle of one X-Siemens-RTP-stats header field per medium. In the absence of any parameter identifying a particular medium, in the absence of any parameter identifying a particular medium the header field applies to the first or sole audio stream identified in SDP.

An example of a complete BYE request is:

```
: BYE sip:49897221000@10.26.100.123:5060;transport=udp SIP/2.0
Via: SIP/2.0/UDP 10.234.3.157;branch=z9hG4bK+5e7f1+10.234.3.157+1
Max-Forwards: 70
Call-ID: kd98e8ekdk9d9
From:   <sip:495251820625@10.26.100.17>;tag=a6fc85ba
To: 49897221000
<sip:49897221000@10.26.100.123:5060>;tag=3950514679
CSeq: 1234567 BYE
Content-Length: 0
X-Siemens-RTP-stats: TB=xx;TE=xx;IPL=xx;PTL=xx;IPR=xx;PTR=xx;
            EN=xx;DE=xx;PR=xx;JI=xx;LA=xx;
            PL=xx;PD=xx;CPL=xx;CGP=xx;
            PS=xx;OS=xx;OR=xx
```

Table J.2 shows the connection parameter specifications for QoS statistics carried in the SIP BYE message.

Connection Parameter	Description / Specification	Details	Example
MT	Media Type	The MT parameter MUST consist of 1 digit which described the media type of the following RT/RTCP parameters 0 = audio 1 = video 2 = text 3 = application 4 = message	MT=0 The following QoS parameters are related to an audio media.

TB	Time Begin
	
	Date and Time of the beginning of report period
	
	TB parameter specification:
	
	`tb = "TB" EQUAL tb-param`
	
	`tb-param = 2DIGIT 2DIGIT 2DIGIT "." 1*DIGIT`
		The TB parameter indicates the beginning of the QoS reporting period which is typically the beginning of the call. It MUST consist of 6 digits and followed by at least one additional digit separated by a decimal point, where the first 4 digits represent the time in 24 hour format indicating when the QoS reporting period started for the call. The 5th and 6th digits are seconds and the value after the decimal point are milliseconds of the reporting period.	B=152359.34
			
			The reporting period started at 15:23 and 59 seconds, 34 milliseconds.

TE	Time End
	
	Date and Time of the end of report period
	
	TE parameter specification:
	
	`te = "TE" EQUAL te-param`
	
	`te-param = DIGIT 2DIGIT 2DIGIT "." 1*DIGIT`
		The TE parameter indicates the end of the QoS reporting period which is typically the end of the call. It MUST consist of 6 digits and MUST consist additional of minimum 1 digit separated by a decimal point, where the first 4 digits represent the time in 24 hour format indicating when the QoS reporting period ended, which is the end of the call. The 5th and 6th digits are seconds and the value after the decimal point are milliseconds of the reporting period.	TE=160312.77
			
			The reporting period ended at 16:03 and 12 seconds and 77 milliseconds

IPL	IP Address, Local
	
	IPL parameter specification:
	
	`ipl = "IPL" EQUAL IPv4address`
	
	`IPv4address= 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT`
		The IPL parameter MUST be the IP address of the entity sending the BYE request or BYE response.	IPL=::10.26.100.123
			
			The SIP client with IPv4 address 10.26.100.123 sends this BYE request or BYE response.

Table J.2 Connection Parameter Specifications for QoS statistics in the X-Siemens-RTP-stats header

Connection Parameter	Description / Specification	Details	Example

PTL	Port #, Local
	
	The PTL parameter specification is:
	
	`ptl = "PTL" EQUAL port`
	
	`port = 1*5DIGIT`
		The PTL parameter MUST be the port at which the RTP stream was received. 1 - 5 digits	PTL=40432
			
			The SIP client received at port 40432

IPR	IP address, Remote
	
	IPR parameter specification:
	`ipr = "IPR" EQUAL IPv4address`
	
	`IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT`
		The IPR parameter MUST be the IP address of the entity sending the RTP stream.	IPR=::10.21.92.59
			
			The SIP client with IPv4 address 10.21.92.59 sends the RTP stream.

PTR	Port #, Remote
	
	PTR parameter specification:
	
	`ptr = "PTR" EQUAL port`
	
	`port = 1*5DIGIT`
		The PTR parameter MUST be the port at which the RTP stream was sent. 1 - 5 digits	PTR=41904
			
			The SIP client sent at port 41904.

SRCS	Source Synchronization Sending Identifier	he SRCS parameter (Source Synchronization Identifier) of the RTP stream that is ""sent"" by the reporting system. This parameter MUST be provided only when the SIP BYE consists of more than one mediastream of the same mediaType (e.g. MT on both streams were 0 / audio). 1 - 10 digits.	RCS=932629361 Identifier of RTP stream that was sent
SRCR	Source Synchronization Receiving Identifier	The SRCR parameter of the RTP stream that is ""received"" by the reporting system. This is the SSRC that is assigned to the stream from which the report was sent. This parameter MUST be provided only when the SIP BYE consists of more than one mediastream of the same mediaType (e.g. MT on both streams were 0 / audio). 1 - 10 digits.	SRCR=819472634 Identifier of RTP stream received

Table J.2 Connection Parameter Specifications for QoS statistics in the X-Siemens-RTP-stats header

Connection Parameter	Description / Specification	Details	Example

EN	Encoding codec used in the RTP stream
	
	EN parameter specification:
	
	`en = "EN" EQUAL en-param`
	
	`en-param = 1*3DIGIT`
		The EN parameter MUST consist of 1 - 3 digits, according to RFC3551 Payload Type Definitions, and represent the encoding codec.
		
		0 = PCMU
		3 = GSM
		4 = G723
		8 = PCMA
		9 = G722
		15 = G728
		18 = G729
			EN=18
			
			The G729 codec was used for encoding the RTP stream.

DE	Decoding codec used in the RTP stream
	
	DE parameter specification:
	
	`de = "DE" EQUAL de-param`
	
	`de-param = 1*3DIGIT`
		The DE parameter MUST consist of 1 - 3 digits according to RFC3551 and represent the decoding codec.
		
		0 = PCMU
		3 = GSM
		4 = G723
		8 = PCMA
		9 = G722
		15 = G728
		18 = G729
			DE=18
			
			The G729 codec was used for decoding the RTP stream.

SS	Silence Suppression activations
	
	SS parameter specification:
	
	`ss = "SS" EQUAL ss-param`
	
	`ss-param = 1*3DIGIT`
		The total number of Silence Suppression activations seen in the received RTP stream between TB and TE	SS=17
			
			17 silent suppression activations have been seen in the received RTP stream.

PR	Packets Received
	
	PR parameter specification:
	
	`pr = "PR" EQUAL pr-param`
	
	`pr-param = 1*DIGIT`
		The PR parameter MUST consist of a minimum of 1 digit and represents the total number of count good packets seen in the received RTP stream.	PR=20317
			
			20317 good packets have been seen in the received RTP stream.

JI	Jitter
	
	JI parameter specification:
	
	`ji = "JI" EQUAL ji-param`
	
	`ji-param = 1*DIGIT`
		The JI parameter MUST consist of minimum 1 digit and represents the maximum jitter encountered over the reporting period in milliseconds.	JI=3
			
			The maximum Jitter over the reporting period was 3 ms.

Table J.2 Connection Parameter Specifications for QoS statistics in the X-Siemens-RTP-stats header

Connection Parameter	Description / Specification	Details	Example

LA	Latency
	
	LA parameter specification:
	
	`la = "LA" EQUAL la-param`
	
	`la-param = 1*DIGIT`
		The LA parameter MUST consist of minimum 1 digit and represents the average round trip delay value over the reporting period. For each RTCP packet (not for each RTP packet) the delay is calculated. The average value is taken and truncated to milliseconds. Average round trip delay between TB and TE.	LA=150
			
			150 ms have been calculated as the average round trip delay for the RTP packets.

PL	Packets Lost
	
	PL parameter specification:
	
	`pl = "PL" EQUAL pl-param`
	
	`pl-param = 1*DIGIT`
		The PL parameter MUST consist of minimum 1 digit and represent the total number of lost packets during the last report period before end of call in the received RTP stream.	PL=37
			
			37 packets have been lost in the received RTP stream.

PD	Packets Discarded
	
	PD parameter specification:
	
	`pd = "PD" EQUAL pd-param`
	
	`pd-param = 1*DIGIT`
		The PD parameter MUST consist of minimum 1 digit and represent the total number of discarded packets in the received RTP stream.	"PD=0
			
			0 packets have been discarded in the received RTP stream.

CPL	Consecutive Packets Lost
	
	CPL parameter specification:
	
	`cpl = "CPL" EQUAL cpl-param`
	
	`cpl-param = 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT`
		he CPL parameter MUST consist of N times minimum 1 digit and represent a statistic of consecutive lost packets in the received RTP stream.	CPL=30.2.1
			
			0 consecutive lost packets have been recorded in the received RTP stream.

Table J.2 Connection Parameter Specifications for QoS statistics in the X-Siemens-RTP-stats header

Connection Parameter	Description / Specification	Details	Example

CGP	Consecutive good packets between TB and TE
	
	CGP parameter specification:
	
	`cgp = "CGP" EQUAL cgp-param`
	
	`cgp-param = 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT "." 1*DIGIT`
		The CGP parameter MUST consist of N times minimum 1 digit and represent a statistic of consecutive good packets in the received RTP stream	CGP=0.0.2.1.6.15329.23

PS	Packets Sent
	
	PS parameter specification:
	
	`ps = "PS" EQUAL ps-param`
	
	`ps-param = 1*DIGIT`
		The PS parameter MUST consist of minimum 1 digit and represent the total number of packets that were sent in the received RTP stream. (PS= Pr + PD)	PS=20354
			
			20354 packets sent in the RTP stream.

OS	Octets Sent
	
	OS parameter specification:
	
	`os = "OS" EQUAL os-param`
	
	`os-param = 1*DIGIT`
		The OS parameter MUST consist of minimum 1 digit and represent the statistcs of octets that were sent in the received RTP stream.	OS=3218
			
			3218 octets sent in the RTP stream.

OR	Octets Received
	
	OR parameter specification:
	
	`or = "OR" EQUAL or-param`
	
	`or-param = 1*DIGIT`
		The OR parameter MUST consist of minimum 1 digit and represent the statistics of octets Number of all good octets received on the connection in the report period.
		
		Good octets are all octets that arrived in time to be inserted in the voice stream and are not discarded for any reason (e.g., buffer overrun).
			OR=43218
			
			43218 octets received in the RTP stream.

IE	Impairments	The IE parameter MUST consist of minimum 1 digit and represent the impairments from low-bit codecs and packet loss.	IE=10 10 impairments from low-bit codecs and packet loss.
TCLW	Telephony echo signal attenuation	The TCLW parameter MUST consist of minimum 1 digit and is a single number that indicates how well the telephone attenuates it's echo signal. TCL is expressed in dB.	TCLW=40 40 dB

Table J.2 Connection Parameter Specifications for QoS statistics in the X-Siemens-RTP-stats header

K Media Recording

The following describes the architecture of the media recording solutions to be deployed with SIP based systems and covers both endpoint and server based recording solutions. The specification is not specific to any particular type of media (voice, video, etc.) and does not specify any special procedures relating to the type of media.

Currently the procedures only cover the establishment of the session with a Recording Server and the associated SIP extensions that are required.

Possibly future extensions to this specification include enhancing the interface to the the Recording Server to provide functionality such as:

- Side-bar or whisper functionality, where specific comments may be injected into the recording device's media stream as booktags
- Providing the Recording Server with details of the participants being recorded.
- Playback
- Discard of the recording

This specification does not impose requirements in the Recording Server itself which is assumed to be either a third party SIP client or a media server. This however may change in the future as more complex recording features are developed and standard based interfaces are defined by the IETF.

K.1 Endpoint Controlled Recording

In this model of session recording the Recording Client initiates the establishment of a SIP session to the Recording Server and controls the session and the associated media. The establishment of the session with the Recording Server may be established in response to a user request or may be established automatically by the endpoint due to configuration however this is not relevant to this specification. The Recording Client includes an indication that the session is being established for the purpose of recording in the SIP INVITE so that other entities (E.g. SIP Server) can handle the session appropriately.

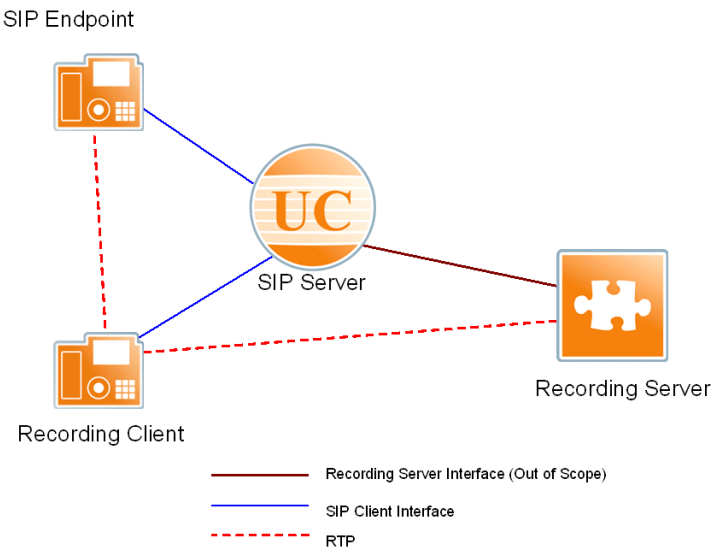


Figure 10 Endpoint Controlled Recording

K.2 Server Controlled Recording

In this model of session recording an application initiates the recording of a session by making a request to the SIP Server using an application interface (E.g. CSTA). The recording is initiated by the SIP Server which redirects the media streams to a Recording Server which acts like a conference server by mixing the media streams as well as recording.

The SIP Server uses SIP 3PCC techniques to control to start and stop the recording by redirecting the media streams and the SIP endpoints may be unaware that any recording is taking place although enhancements to the SIP Client Interface are specified in this document which allow the SIP Server to notify the SIP Endpoint that the session is being recorded.

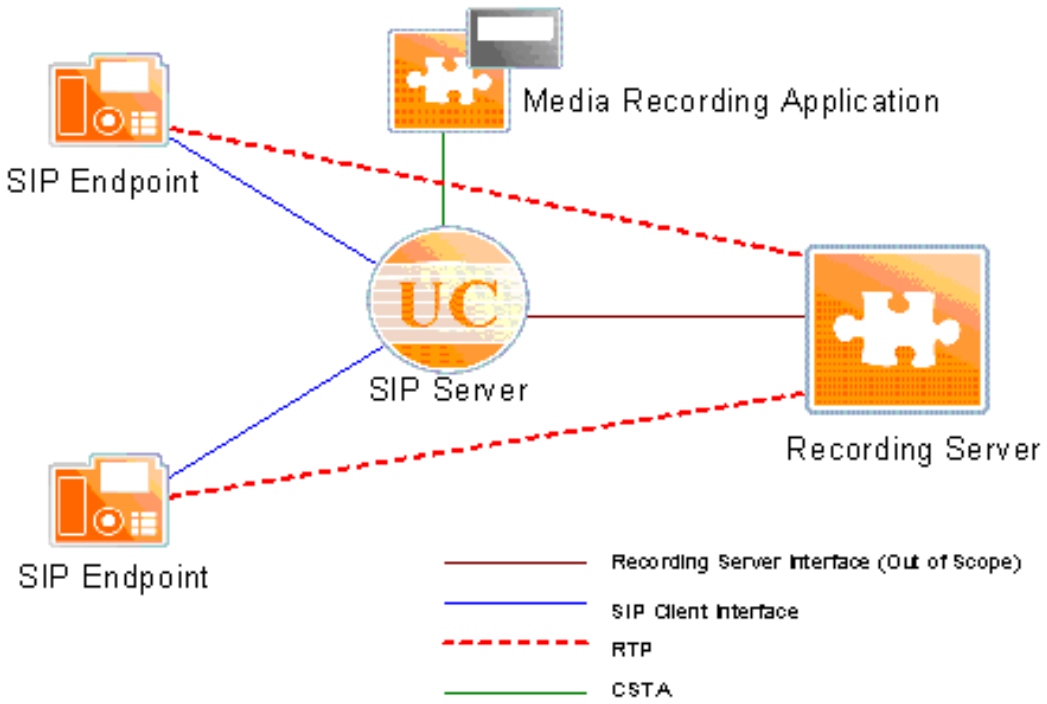


Figure 11 SIP Server Controlled Recording

K.3 Procedures for Endpoint Controlled Recording

K.3.1 Actions at the Recording Client

The Recording Client makes the decision as to which sessions are recorded using local policy in the endpoint. Some example policies include:

1. Record all calls with no user control.
2. Record all calls with user override.
3. User controlled recording.

The Recording Server URI is configured in the SIP endpoint.

To initiate the recording of a session the Recording Client SHALL send a SIP INVITE request to the Recording Server URI using normal SIP mechanisms.

The SIP INVITE request SHALL contain an X-Siemens-Call-Type header with the value "recording" the purpose of which is to inform the SIP Server/B2BUA that the session is being established for the purpose of recording another session.

After receipt of a 200OK response from the Media Recorder then the Recording Client mixes the recorded media streams locally in the endpoint and routes the mixed streams to the Media Recorder.

The Recording Client MAY also include an X-Siemens-Call-Type header with the value "recorded" in a SIP INVITE, re-INVITE, UPDATE or 200OK on the dialog(s) that is being recorded.

K.3.2 Actions at the SIP Server (B2BUA)

The SIP Server must support receipt of an INVITE requests from the Recording Client and take note of the X-Siemens-Call-Type value of "recording".

How the B2BUA makes use of the "recording" indication is outside the scope of this specification but example uses include:

a) Hide the recording session from CSTA Applications.

b) Hide the recording session from keyset devices monitoring the Recording Client.

c) Prevent bridging on to the recording session.

The SIP Server MAY include an X-Siemens-Call-Type header with the value "recorded" in a SIP INVITE, re-INVITE, UPDATE or 200OK sent to any SIP Endpoint that is providing media into the session being recorded.

The SIP Server may add a Diversion header field with the identity of the party with which the Recording Client is connected (call-partner), in the SIP INVITE that it forwards to the Recording Server. The Diversion header field shall have the following format:

Diversion: Call-partner’s Name/Number, reason=unknown, counter=1

K.3.3 Actions at the SIP Endpoint

A SIP Endpoint must be able to receive an X-Siemens-Call-Type header with the value "recorded" in a SIP INVITE, re-INVITE, UPDATE or 200OK and MAY render this information to the user.

K.3.4 Actions at the Recording Server

No special procedures are currently defined for the Recording Server which is assumed to interface to the SIP Server using a standard SIP Client or Networking interface.

K.3.5 Security Considerations

The Recording Client must support the procedures for signalling and payload encryption when establishing the session with the Recording Server.

If the session or sessions that are to be recorded are secure but the session which is established with the Recording Server is determined to be insecure, then the Recording Client must send a re-INVITE on each dialog which is being recorded and include the X-Siemens-Call-Type header with the value "ST-insecure".

K.4 Procedures for Server Controlled Recording

K.4.1 Actions at the SIP Server

The SIP Server may initiate the recording of a session during the initial establishment of the session or when the session is already active.

The SIP Server uses standard SIP 3PCC techniques to route the media streams to a Recording Server.

The SIP Server must inform the SIP endpoint that the session is being recorded by including the "X-Siemens-Call-Type" header with the value "recorded" in an INVITE, re-INVITE, UPDATE or 200OK as appropriate.

K.4.2 Actions at the SIP Endpoint

See Sect. K.3.3.

K.4.3 Actions at the Recording Server

No special procedures are currently defined for the Recording Server which is assumed to interface to the SIP Server using a standard SIP Client or Networking interface.

The Recording Server in this model is however required to mix the media streams.

K.4.4 Security Considerations

The SIP Server must support the procedures for signalling and payload encryption when establishing the session with the SIP Endpoint.

If the session with the Recording Server is determined to be insecure then the SIP Server must inform the SIP Endpoint using the X-Siemens-Call-Type header with the value "ST-insecure".

In the case of server controlled recording the procedures for enabling and disabling recording will result in the redirection of media streams and therefore when using SRTP for payload encryption the endpoint must support the procedures for re-keying.

K.5 Section Non-standard data

This specification uses the X-Siemens-Call-Type header field with the specific values "recording" and "recorded".

Index

T

U