

SIP Handbook: Services, Technologies, and Security

January 13, 2008

Contents

1	Narrowcasting in SIP: Articulated Privacy Control	1
1.1	Introduction	2
1.2	SIP Conferencing: Model and Control	5
1.2.1	Conference Model	5
1.2.2	Conference Control	6
1.3	Media Privacy: Narrowcasting Concept	8
1.3.1	Mute	11
1.3.2	Deafen	11
1.3.3	Select (Solo)	12
1.3.4	Attend	12
1.4	System Design and Implementation	12
1.4.1	Policy Configuration	13
1.4.2	Policy Evaluation	13
1.4.3	Media Mixing and Distribution	14

1.4.4	Sample Mixing Configuration	15
1.4.5	Narrowcasting in Virtual Environments	18
1.4.6	System Performance	18
1.5	Conclusion and Future Research	20
1.5.1	Practical Conferencing	20
1.5.2	Presence	22
1.5.3	Architectural and Interface Refinement	22
1.5.4	Convergence	24

Chapter 1

Narrowcasting in SIP: Articulated Privacy Control

Sabbir Alam, Michael Cohen, & Julián Villegas

Spatial Media Group, University of Aizu

Aizu-Wakamatsu 965-8580; Japan

E-mail: {d8062102, mcohen}@u-aizu.ac.jp, julovi@yahoo.com

Phone: [+81](242)37-2537 Fax: [+81](242)37-2772

Ashir Ahmed

Dept. of CSCE, Kyushu University

744 Moto'oka, Fukuoka 819-0395; Japan

E-mail: ashir@c.csce.kyushu-u.ac.jp

Phone: [+81](928)02-3667 Fax: [+81](928)02-3850

1.1 Introduction

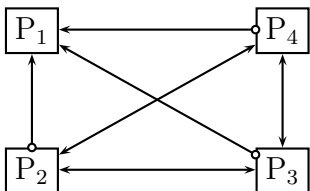
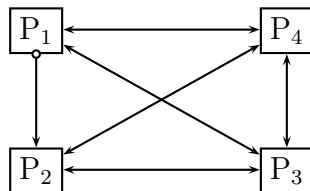
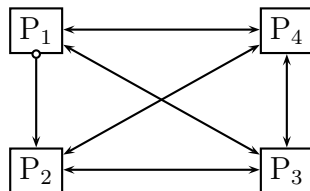
In multimedia conferencing, media streams are exchanged between participants once a session is established by setting up communication channels within a group. By default, each participant receives a combined stream obtained by mixing media transmitted by the other participants. Situations arise when a participant wants to select a subset of the conference participants to whom her media are sent or from whom streams are received. Media filters are necessary to allow privacy of the participants in the conference. In analogy to broad-, multi-, any-, and swarm-casting, narrowcasting is a technique for limiting and focusing information streams. Narrowcasting systems extend broad- and multicasting systems by allowing media streams to be filtered— for relevancy control, privacy, security, and user interface optimization. In this chapter, we describe four narrowcasting commands— `mute`, `deafen`, `select`, and `attend`— to provide distributed privacy.

Extensive research has been carried out in the area of conference and floor control [17] [13]. Conventional features regarding media privacy in conferences are typically limited to scheduling and selecting the speaker. Advanced conferencing features such as adding/deleting participants, changing user agents or modes (like switching from a desktop to a mobile phone), changing media, authenticating or authorizing participants, granting privileges, controlling presentation of media, sidebars, passive participants, whisper/private messages, audio-only, and lecture mode are described in RFC 4597 [15]. Media privacy features allow participants to control their own information and to distribute their attention, based on secrecy, anonymity, and solitude [24].

Mute is a popular feature for media privacy. It has three different varieties: self-mute, PBX-mute, and narrowcasting mute, juxtaposed in Table 1.1. Self-mute allows a user to withhold his media from other participants. In PBX-mute, a controller disables a participant's outgoing media to other participants. Narrowcasting mute refers to P2P control with which a participant (controller) can select another participant (controllee) to disallow the controllee's media towards the controller.

A Call Whisper [4] feature allows a participant to whisper to one or more participants

Table 1.1: Three different Mute operations

	Self-Mute	PBX Mute	Narrowcasting mute
Media Vector			
Media Distribution	$P_1 \leftarrow (P_2 + P_3 + P_4)$ $P_2 \leftarrow (P_3 + P_4)$ $P_3 \leftarrow (P_2 + P_4)$ $P_4 \leftarrow (P_2 + P_3)$	$P_1 \leftarrow (P_3 + P_4)$ $P_2 \leftarrow (P_1 + P_3 + P_4)$ $P_3 \leftarrow (P_1 + P_2 + P_4)$ $P_4 \leftarrow (P_1 + P_2 + P_3)$	$P_1 \leftarrow (P_3 + P_4)$ $P_2 \leftarrow (P_1 + P_3 + P_4)$ $P_3 \leftarrow (P_1 + P_2 + P_4)$ $P_4 \leftarrow (P_1 + P_2 + P_3)$
Semantics	Self Control. P_1 mutes himself by turning off his mic so that no media goes to the media server, or P_1 can send a “self-mute” signal to the application server so that the media server simulates self-censorship.	Control by Admin. P_1 is muted by moderator. P_1 ’s media is not mixed in the media server. P_1 is in a listen-only, or “lurker” (stealth) mode.	P2P Control. P_1 mutes P_2 . P_2 may speak to everyone, but P_1 won’t hear his voice.

in a group. This walkie-talkie-like feature creates a one-way voice or video communication for a limited period of time. The session terminates when the controller releases the “PTT” (push to talk) button, so such a system is not practical when a longer session or two-way communication session is necessary. Voice Chat [27] allows participants to create one or more private audio conferences. Although the communication channel in the private voice chat group provides two-way communication, participants can hear the main conference at low volume. Private conversation [26] offers a private video, voice, and text conversation session inside a main conference. It is similar to a call whisper feature, but adds two-way communication capability and text messaging. In a WebEx (www.webex.com) audio conference, a conference host can (un)mute the microphones to allow only certain attendees to speak. An ‘audio-only’ option host can grant and restore speaking privileges to attendees, so that designated attendees can listen but not speak. WebEx participants can have a private chat with someone during a meeting. Whisper Coaching (www.audiocodes.com) allows a

supervisor to listen to a main conference conversation while talking to a selected set of participants at the conference. The privacy control allowed by these applications is rather blunt. In order to better control media privacy, we are exploring the concept and practical applications of narrowcasting [16] [2] [3].

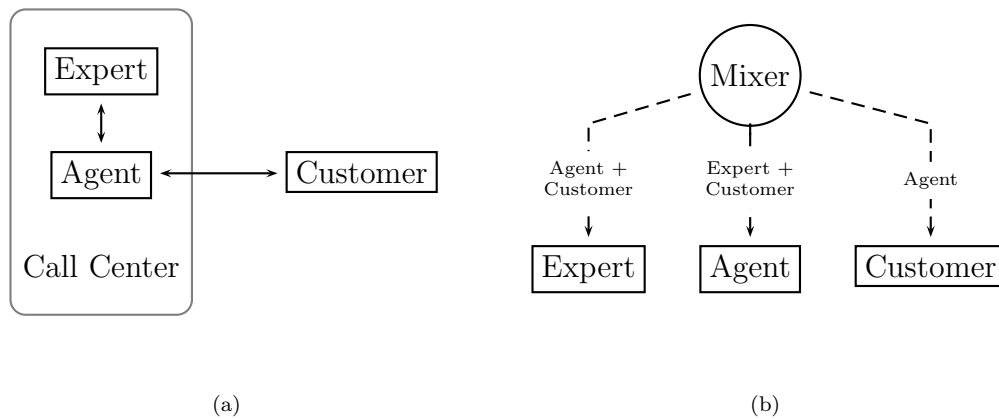


Figure 1.1: Media Privacy: A Call Center Application Scenario

A call center scenario provides an example of media privacy: in instances when a first-tier agent cannot answer a customer’s questions, the agent could have a private side-channel communication with supervisor as back-up for realtime customer support, as shown in Fig. 1.1(a). Privacy control is invoked so that the expert’s media goes only to the agent, not to the customer, as shown in Fig. 1.1(b). As a result, the agent can improve customer satisfaction. Traditional conferencing systems do not generally provide such features. In this chapter, we describe a mechanism and instance of “Media Server Component Model” architecture for policy-based media mixing with a centralized media mixer within the standard SIP [21] framework for multimedia conferencing systems. We have defined media privacy commands and developed a policy evaluation algorithm, a media mixing and delivery mechanism considering policy configured by conference participants. For instance, **attend**, one of the narrowcasting commands defined and implemented, can accommodate such call center privacy control requirements.

1.2 SIP Conferencing: Model and Control

A traditional conferencing system using the PSTN (public switched telephone network) has limited features, implemented in a centrally controlled conference server. A more modern infrastructure, SIP, uses internet signaling and media streams. Due to the simplicity and flexibility of its control and management of multimedia conference services, we concentrate on SIP-based conferencing models.

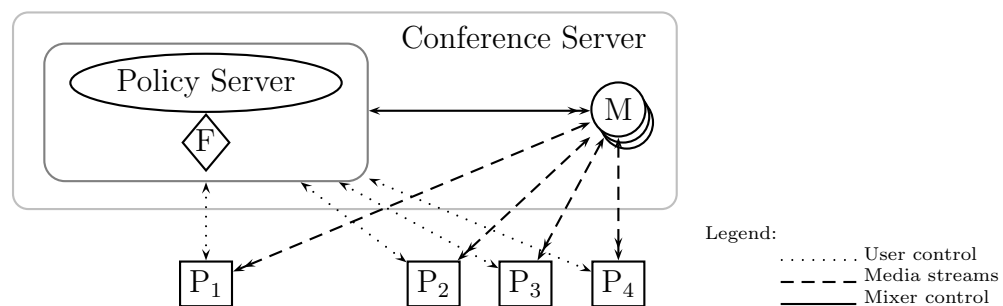


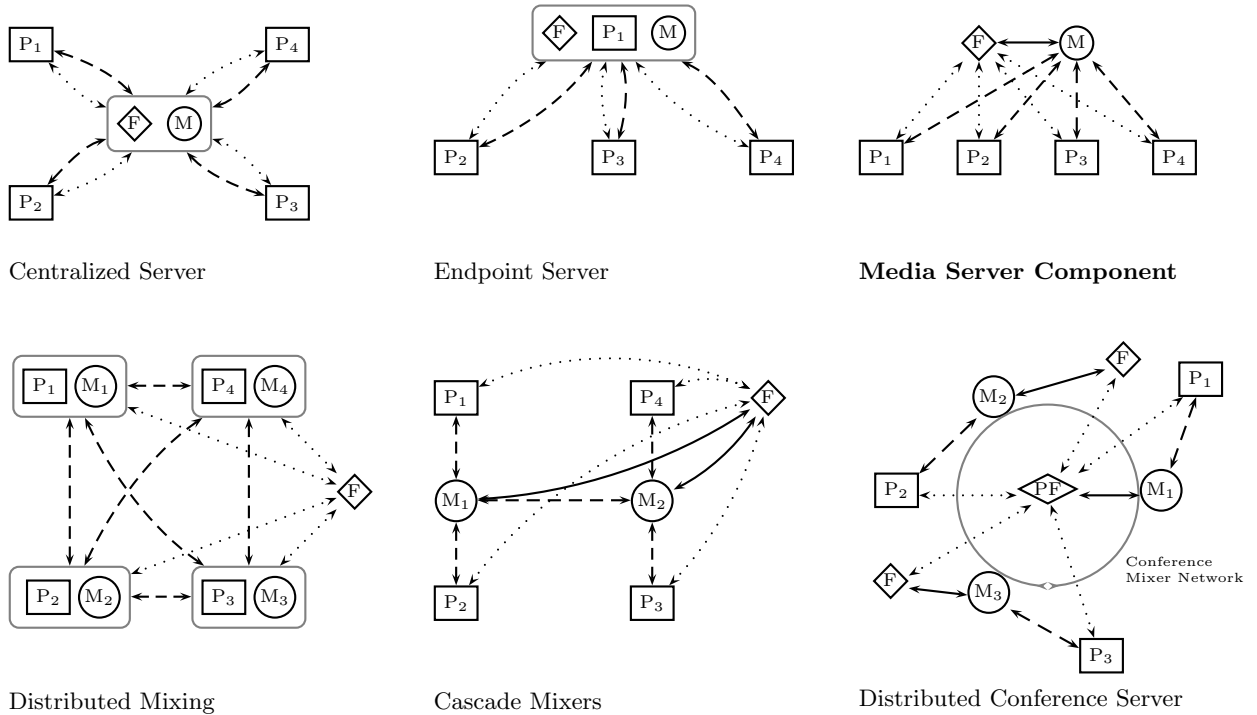
Figure 1.2: Typical Conference Architecture: F is the focus, M are media mixers, and P_i are the participants.

A conference server and the participants are two major components of a centralized conference system (Fig. 1.2). A SIP conference server comprises a focus, policy server, and media mixer. The focus handles the conference control—creating, modifying, and terminating conferences. Conference policy is managed by the policy server, which configures the media server. Mixing and distribution of media streams are the main functions of a media mixer, which returns some composite display to the respective terminals, as suggested by the multiple arrowheads on the return vectors. Value-added services such as monitoring conference status, participant status, and billing can be implemented inside or outside of this framework.

1.2.1 Conference Model

There are two generic conference models: loosely and tightly coupled. In a loosely coupled model, there is neither a central point of control nor a conference server, whereas in a tightly coupled model, a centralized conference control server manages the conferences. A tightly

Table 1.2: Conferencing Models: ‘P’ indicates participant, ‘F’ indicates focus, ‘M’ indicates media mixer, and (in the last model) ‘PF’ indicates Primary Focus. Dotted lines indicate signaling, dashed lines indicate media transmission, and solid lines indicate mixer control.



coupled conferencing model can be further classified into six different types depending on the location of the focus and the mixer, as illustrated in Table.1.2, including the Media Server Component Model used for our proof-of-concept. These models are detailed by J. Rosenberg [19] and Y. Cho et. al. [5].

1.2.2 Conference Control

Conference control refers to the ability of a participant to manipulate the state of a session. A conference is represented by a unique URI (uniform resource identifier), usually a SIP URI, that identifies the focus of a conference. (A conference URI can be emailed, sent in an instant message, linked on a web page, or obtained from some non-SIP mechanism.) Conference control includes three primary functions:

- Creation: A participant joins a conference by sending an INVITE request to its focus (“dial-in”) or by the focus sending an INVITE request to the participant (“dial-out”) citing the conference URI.
- Modification: A participant or focus can modify a session in a conference using a re-INVITE. For instance, when an audio conference extends to video, the focus re-INVITES each participant adding a video media stream. A participant or focus may also put media streams on hold, or take them off hold. Narrowcasting commands are applied to a session by selectively enabling the media streams.
- Termination: A privileged participant (typically a moderator or conference creator) terminates a session by sending a BYE request to the focus. The focus then distributes a BYE request to all other participants in the conference, terminating the session.



Figure 1.3: Privacy: Freedom from disturbance. (©2008 The New Yorker Collection from cartoonbank.com. All rights reserved.)

Privacy has two interpretations. The first association, with sources, is that of avoiding “leaks,” protecting secrets. But a second interpretation, with sinks, means freedom from

disturbance, in the sense of solitude, not being bothered by irrelevance or interruption, as suggested by Fig. 1.3. Our distributed interface features narrowcasting operations that manage privacy in both senses, by filtering duplex media flow through an articulated conferencing model that limits and focuses information streams.

1.3 Media Privacy: Narrowcasting Concept

In traditional conferencing systems, participants have little or no privacy, as their voices are by default shared with all others in a session. Such systems cannot offer participants the options of muting and deafening other members. The concept of narrowcasting can be applied to make these kinds of filters available in multimedia conferencing systems. Our system treats media sinks (in the simplest case, listeners) as full citizens, peers of the media sources (conversants' voices), and we defined therefore duals of `mute & select`: `deafen & attend`, which respectively block a sink or focus on it to the exclusion of others. Fig. 1.4 shows a famous Japanese carving which illustrates the features of narrowcasting. Three monkeys—Mizaru (with covered eyes), Iwazaru (covered mouth), and Kikazaru (blocked ears)—manifest the notion of limiting media vectors. Mizaru can not see but can hear and speak; Iwazaru can not speak but can see and hear; Kikazaru can not hear but can speak and see.



Figure 1.4: Media Privacy (Narrowcasting Features)

For modern groupware situations like teleconferences, in which everyone can have presence across the global network, users want to shift and distribute attention (apathy) and

accessibility/availability/exposure (privacy), and narrowcasting provides a formalization of such filters. The narrowcasting predicate calculus [8] shown in Fig. 1.5 is an appropriate basis for such a permission scheme.

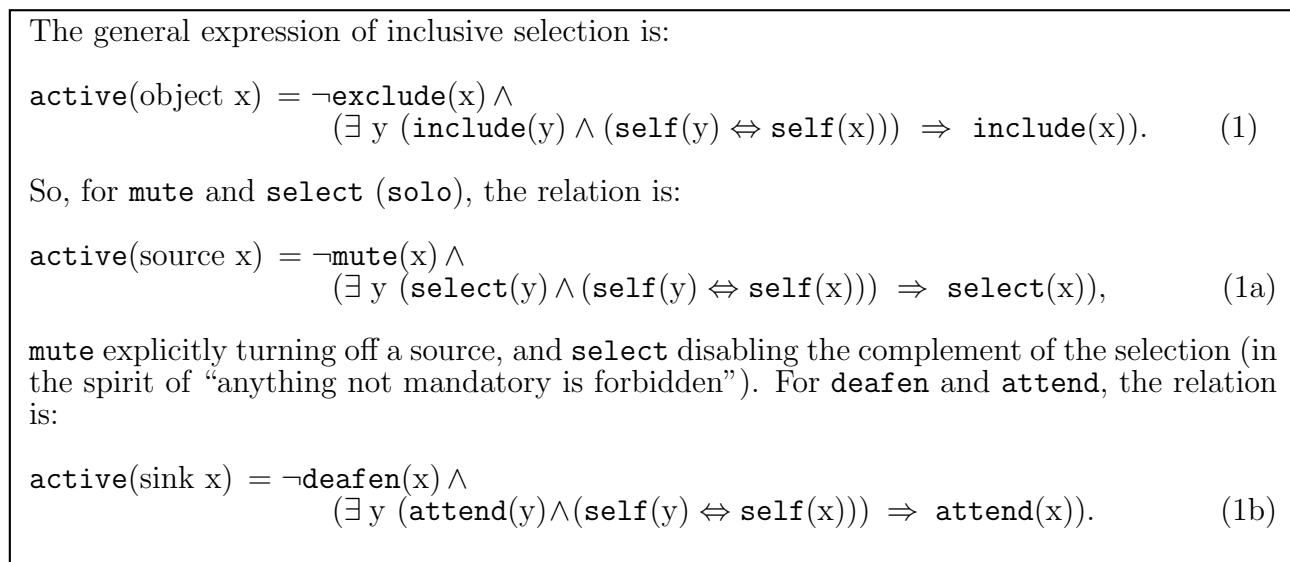


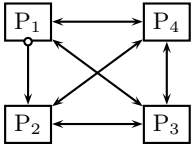
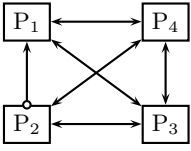
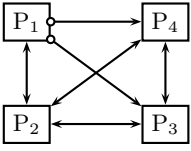
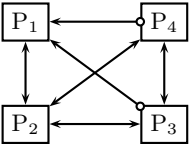
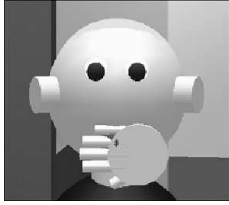



Figure 1.5: Formalization of narrowcasting and selection functions in predicate calculus notation, where ‘ \neg ’ means “not,” ‘ \wedge ’ means conjunction (logical “and”), ‘ \exists ’ means “there exists,” ‘ \Rightarrow ’ means “implies,” and ‘ \Leftrightarrow ’ means mutual implication (equivalence).

The duality between source and sink operations is tight, and the semantics are identical: an object is inclusively enabled by default unless, it explicitly excluded (with ^{source}**mute** or ^{sinks}**deafen**), or, peers of the same **self**/non-**self** class are explicitly included (with ^{source}**select [solo]** or ^{sinks}**attend**) when the respective object is not. Narrowcasting attributes are not mutually exclusive, and the dimensions are orthogonal. Because a source or a sink is active by default, invoking **exclude** and **include** operations simultaneously on an object results in its being disabled. For instance, a sink might be first **attended**, perhaps as a member of some non-singleton subset of a space’s sinks, then later **deafened**, so that both attributes are simultaneously applied. (As audibility is assumed to be a revocable privilege, such a seemingly conflicted attribute state disables the sink, whose attention would be restored upon resetting its **deafen** flag.) Symmetrically, a source might be **selected** and then **muted**, akin to making a “short list” but relegated to back-up.

Narrowcasting audio commands are listed and their characteristics arrayed in Table 1.3.

Our design allows each user to send or receive data streams to/from a specific list of recipients in a session. For easier understanding, we consider only audio streams in this chapter. However, this design applies equally well to other media types.

Table 1.3: Narrowcasting Commands

	P_1 mutes P_2	P_1 deafens P_2	P_1 selects P_2	P_1 attends P_2
Media Vectors				
Semantics	Block the media stream coming from a source.	Block media streams going to a sink.	Limit the projected sound to particular sources.	Limit the received sound to particular sinks.
Situation	A participant wants to block media from specific participants.	A participant wants to block media to specific participants.	A participant wants to receive media only from particular participants.	A participant wants to send media to specific participants.
Figurative Avatars				
Mobile Icons	$\bar{\Delta}$	$-\Delta-$	$+\Delta$	$+\Delta+$
Media Distribution	$\begin{bmatrix} \times & 1 & 1 & 1 \\ \mathbf{0} & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}$	$\begin{bmatrix} \times & \mathbf{0} & 1 & 1 \\ 1 & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}$	$\begin{bmatrix} \times & 1 & 1 & 1 \\ \mathbf{1} & \times & 1 & 1 \\ \mathbf{0} & 1 & \times & 1 \\ \mathbf{0} & 1 & 1 & \times \end{bmatrix}$	$\begin{bmatrix} \times & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 1 & \times & 1 & 1 \\ 1 & 1 & \times & 1 \\ 1 & 1 & 1 & \times \end{bmatrix}$

In this section, we formally define four narrowcasting commands. In the following expressions, P_a denotes the actor (controller), P_o the object (controllee), P_i a sender of the media (source), P_j a receiver of the media (sink), and $a, i, j, o \in \{1..n\}$, where n is the total number of participants.

1.3.1 Mute

The narrowcasting command `mute` blocks media coming from a source. The mute in traditional systems is a self-mute function which allows a user to withhold his/her media from other participants, but the modern `mute` is a control function that can select another participant (or a group of participants) to disallow media towards the controller, still allowing other participants to hear the controllee. The \sum operator composites media from the respective participants.

$$P_j \leftarrow \begin{cases} \sum_{i=1}^n P_i - P_j - P_o & \text{when } P_j = P_a, \\ \sum_{i=1}^n P_i - P_j & \text{otherwise.} \end{cases} \quad (1.1)$$

The example modeled by the matrix in the first column of Table 1.3 illustrates when P_1 mutes another participant P_2 ($a = 1$ and $o = 2$). In this example, $n=4$, $P_a=P_1$ (the controller), and $P_o=P_2$ (the controllee). Due to this operation, P_1 will not receive any media from P_2 .

1.3.2 Deafen

`Deafen` is a sink-related media privacy command that blocks media streams to a selected participant. For example, if Bob (P_1) wants to share his media with everyone in a conference except Alice (P_2), then Alice will not receive any streams from Bob if Bob `deafens` Alice. Transposing the participants one can realize the equivalent operation, P_2 mutes P_1 . The second column of Table 1.3 shows the media relationship among four participants.

$$P_j \leftarrow \begin{cases} \sum_{i=1}^n P_i - P_j - P_a & \text{when } P_j = P_o, \\ \sum_{i=1}^n P_i - P_j & \text{otherwise.} \end{cases} \quad (1.2)$$

Again in this example, $n=4$, $a = 1$, and $o = 2$.

1.3.3 Select (Solo)

The privacy command `select` limits received media to particular sources. Students might `select` a teacher to avoid distractions. P_1 will receive media only from P_2 if P_1 `selects` P_2 , implicitly `muting` the complement of the selection. The third column of Table 1.3 shows the media relationships among four participants; two vectors are disabled in this case.

$$P_j \leftarrow \begin{cases} P_o & \text{when } P_j = P_a, \\ \sum_{i=1}^n P_i - P_j & \text{otherwise.} \end{cases} \quad (1.3)$$

1.3.4 Attend

`Attend` is the other including command for media privacy, limiting received sound to a particular recipient. If Alice `attends` Bob, only Bob will hear Alice, since other participants are implicitly `deafened`. The rightmost column of Table 1.3 shows the media relationship among four participants; again two media vectors are suppressed.

$$P_j \leftarrow \begin{cases} \sum_{i=1}^n P_i - P_j & \text{when } P_j = P_o, \\ \sum_{i=1}^n P_i - P_j - P_a & \text{otherwise.} \end{cases} \quad (1.4)$$

1.4 System Design and Implementation

The main required functionalities for media policy configuration and control are policy configuration, policy evaluation, and media mixing and distribution. The Media Server Component Model (top right of Table 1.2) selected for our implementation comprises a centralized focus (collocated with the policy server), a centralized mixer, and participants. The architecture, elaborated in Fig. 1.6, embeds policy configuration, media mixing, and a collaborative virtual environment (CVE) interface within a SIP framework. All the components in this architecture are standard SIP UAs extended with additional user interfaces needed for media policy configuration and control. The communication protocols XCAP (Extensible Markup

Language Configuration Access Protocol) [20] and MSCML (Media Server Control Markup Language) [14] are IETF standards.

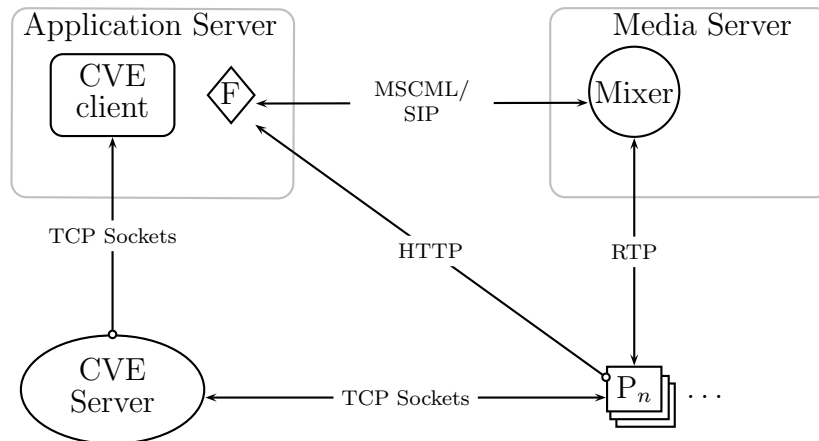


Figure 1.6: Media Server Component Model with Collaborative Virtual Environment Integration

1.4.1 Policy Configuration

In an extended SIP framework, conference participants could configure privacy by sending requests to the policy server using XCAP, a standardized way to use HTTP to store, retrieve, and manipulate configuration and application data in XML format. In our proof-of-concept, participants set policies using GUIs to invoke narrowcasting commands specifying controllees (to whom the narrowcasting commands apply) and control is via TCP sockets or HTTP directly (without XCAP).

1.4.2 Policy Evaluation

An application server performs three major functions to evaluate policy:

Evaluating policies configured by each participant: The policy from each participant can be logically compiled into a matrix, as shown in Table 1.4, where entry c_{ij} of the matrix represents connectivity of source i to sink j , and the main diagonal is populated

by “don’t care”s. Each participant (P_1, P_2, \dots, P_n) , where n is the total number of participants, logically sets permissions in authorized cells. Since a media relationship ultimately factors at least two participants, each cell contains policies from both. For example, $P_1 \rightarrow P_2$, i.e. media sourced at P_1 and sunk at P_2 , has policy involvement of both P_1 and P_2 : P_1 sets permissions about whether or not to send media to P_2 , and at the same time, P_2 sets permissions about whether or not to receive such media. The policy then is evaluated depending on the combined relationship between P_1 and P_2 .

Table 1.4: Policy Matrix $P = [p_{ij}]$

	P_1	P_2	\dots	P_n
P_1		$P_1(P_1 \rightarrow P_2)$ $P_2(P_1 \rightarrow P_2)$	\dots	$P_1(P_1 \rightarrow P_n)$ $P_n(P_1 \rightarrow P_n)$
P_2	$P_2(P_2 \rightarrow P_1)$ $P_1(P_2 \rightarrow P_1)$		\dots	$P_2(P_2 \rightarrow P_n)$ $P_n(P_2 \rightarrow P_n)$
\vdots	\vdots	\vdots	\ddots	\vdots
P_n	$P_n(P_n \rightarrow P_1)$ $P_1(P_n \rightarrow P_1)$	$P_n(P_n \rightarrow P_2)$ $P_2(P_n \rightarrow P_2)$	\dots	

Responding to participants regarding changes made in the policy: A policy evaluation report (confirming success or alerting failure of a configuration request) can be sent to participants via standard XCAP response codes.

Sending requests to a media mixer for necessary media mixing: After compiling the media policies, the system determines which media streams need to be mixed and delivered to whom. The policy server instructs the media mixer to perform the necessary mixing using standard MSCML.

1.4.3 Media Mixing and Distribution

The media server receives MSCML requests from a policy configuration server. According to the accumulated state, it performs the necessary mixing and delivers these streams to

subscribed participants. The maximum number of mixes, the power set of the participants excluding the empty and universal sets, is:

$$\sum_{i=1}^{n-1} {}_n C_i = 2^n - 2. \quad (1.5)$$

Therefore, for $n = 3, 4, 5$, the maximum number of mixes would be 6, 14, and 30, respectively. However, depending on participants' media privacy requests, the actual number of mixes might be fewer.

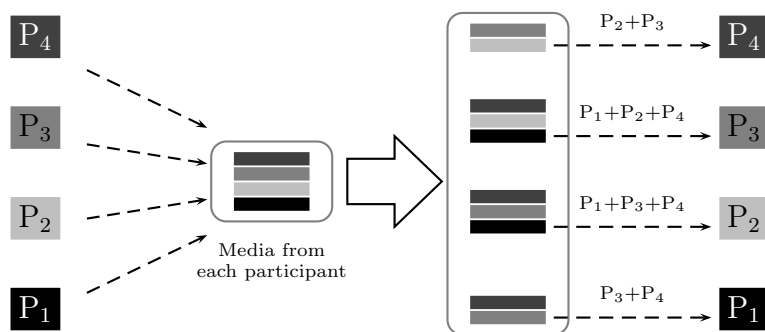


Figure 1.7: Media Mixing and Delivery (P₁ mutes P₂ and deafens P₄)

Fig. 1.7 illustrates narrowcasting media distribution between four participants when P₁ mutes P₂ and deafens P₄. All participants send their media to the media mixer. The media mixer mixes only the necessary streams and delivers them back to the appropriate recipients.

1.4.4 Sample Mixing Configuration

Our prototype environment comprises a SIP server (BEA WebLogic SIP Server), an application server (BEA WebLogic Workshop), a media server (Dialogic/Cantata Snowshore IP Media Server), and four SIP clients (X-lite). We implemented narrowcasting commands `mute`, `deafen`, `attend`, and (partially) `select`, integrating these filter functions into the application server. Fig. 1.8 shows the control and media streams among a participant, application server, and media mixer when applying a narrowcasting command. The following

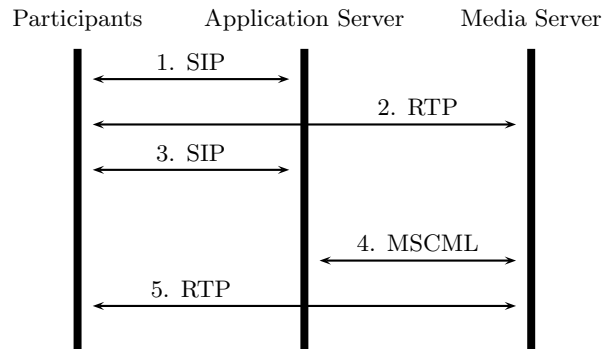


Figure 1.8: Communication Flow Between SIP Entities: A default configuration (1.) establishes a normal session (2.), but it can be adjusted (3.) to reconfigure (4.) the mixes returned to the participants (5.).

trace shows the MSCML code sent from the policy configuration (application) server to the media mixer when P_1 deafens P_2 . In each block, the first chunk is the SIP headers and the second chunk, the body, is the MSCML payload. P_1 makes a private group with P_3 and P_4 , so P_1 , P_3 , and P_4 can hear each other, but P_2 cannot hear P_1 . The policy server evaluates the policy and configures the media server. MSCML configuration and audibility are shown in Table 1.5.

Table 1.5: MSCML Configuration: P_1 deafens P_2

Participant	ID	Team Members	Mixmode	Listeners
P_1	P_1	P_3, P_4	Private	$P_2 + P_3 + P_4$
P_2	P_2	None	Full	$P_3 + P_4$
P_3	P_3	P_1, P_4	Full	$P_1 + P_2 + P_4$
P_4	P_4	P_1, P_3	Full	$P_1 + P_2 + P_3$

Note: irrelevant headers are elided and nested blocks indented for readability

INFO sip:192.168.1.12:5060 SIP/2.0

To: <sip:conf=conference_00@192.168.1.12>;tag=1168487679

Content-Length: 350

From: <sip:P1@192.168.1.11>;tag=ee2d88d1

Content-Type: application/mediaservercontrol+xml

Max-Forwards: 66

```

<?xml version='1.0'?>
<MediaServerControl version="1.0">
  <request>
    <configure_leg mixmode="private" id="sip:P1@192.168.1.11">
      <configure_team action="add">
        <teammate id="sip:P4@192.168.1.11"/>
        <teammate id="sip:P3@192.168.1.11"/>
      </configure_team>
    </configure_leg>
  </request>
</MediaServerControl>

```

The media server confirms the configured media mixing and delivery using MSCML.

```

INFO sip:app-1w4n5gq0kbcgv@192.168.1.11:5060;transport=udp;wlsscids=-7ba6e82e1ed19297;lr SIP/2.0
Via: SIP/2.0/UDP 192.168.1.12:5060
To: <sip:P1@192.168.1.11>;tag=ee2d88d1
From: sip:conf=conference_0@192.168.1.12;tag=1168487679
Content-Type: application/mediaservercontrol+xml
Content-Length: 281

```

```

<?xml version="1.0"?>
<MediaServerControl version="1.0">
  <response request="configure_leg" code="200" text="OK">
    <team id="sip:P1@192.168.1.11" numteam="2">
      <teammate id="sip:P3@192.168.1.11"/>
      <teammate id="sip:P4@192.168.1.11"/>
    </team>
  </response>
</MediaServerControl>

```

1.4.5 Narrowcasting in Virtual Environments

Our group developed “Multiplicity” [16] to manipulate narrowcasting attributes in virtual spaces via a Java3D interface [25] [23]. This “virtual reality”-style interface features perspective displays of virtual rooms with figurative avatars, each of which can be associated with an audio source, the voice of a corresponding user. A participant can rearrange the locations of avatars in virtual spaces and designate a sink, through whose ears the resulting spatialized soundscape can be heard. Also, a participant can apply narrowcasting attributes to the avatars, altering the sound mix. An action taken by a participant is communicated using our CVE client/server architecture. This framework allows multimodal clients to exchange status data through the network in a standardized manner. Clients currently include sound spatializers, telepresence applications, panoramic browsers, music visualizers, motion platforms, and mobile interfaces.

We created a bridge between our SIP narrowcasting controls and Multiplicity [1]. The results of narrowcasting operations are expressed aurally by the SIP-based mixer and visually by Multiplicity. Recalling the monkeys in Fig. 1.4, Fig. 1.9 illustrates the visual cues used for narrowcasting, including a hand covering the mouth of the `muted` avatar and hands clapped over the ears of a `deafened` avatar.

The bridge between the Java3D interface and the SIP-based backend is a ‘read-only’ CVE client embedded in the SIP application server. When the policy server is launched, the client connects to a CVE server and opens a channel for each member in the conference. Every time a user enables or disables one of the narrowcasting attributes in Multiplicity, the action is relayed to the embedded CVE client. As each message is received, the client invokes the necessary methods to reflect the changed status in the SIP conference.

1.4.6 System Performance

The narrowcasting control is basically light-weight: the commands are typically infrequent, and each of them is easily processed by an application server. For excluding narrowcasting

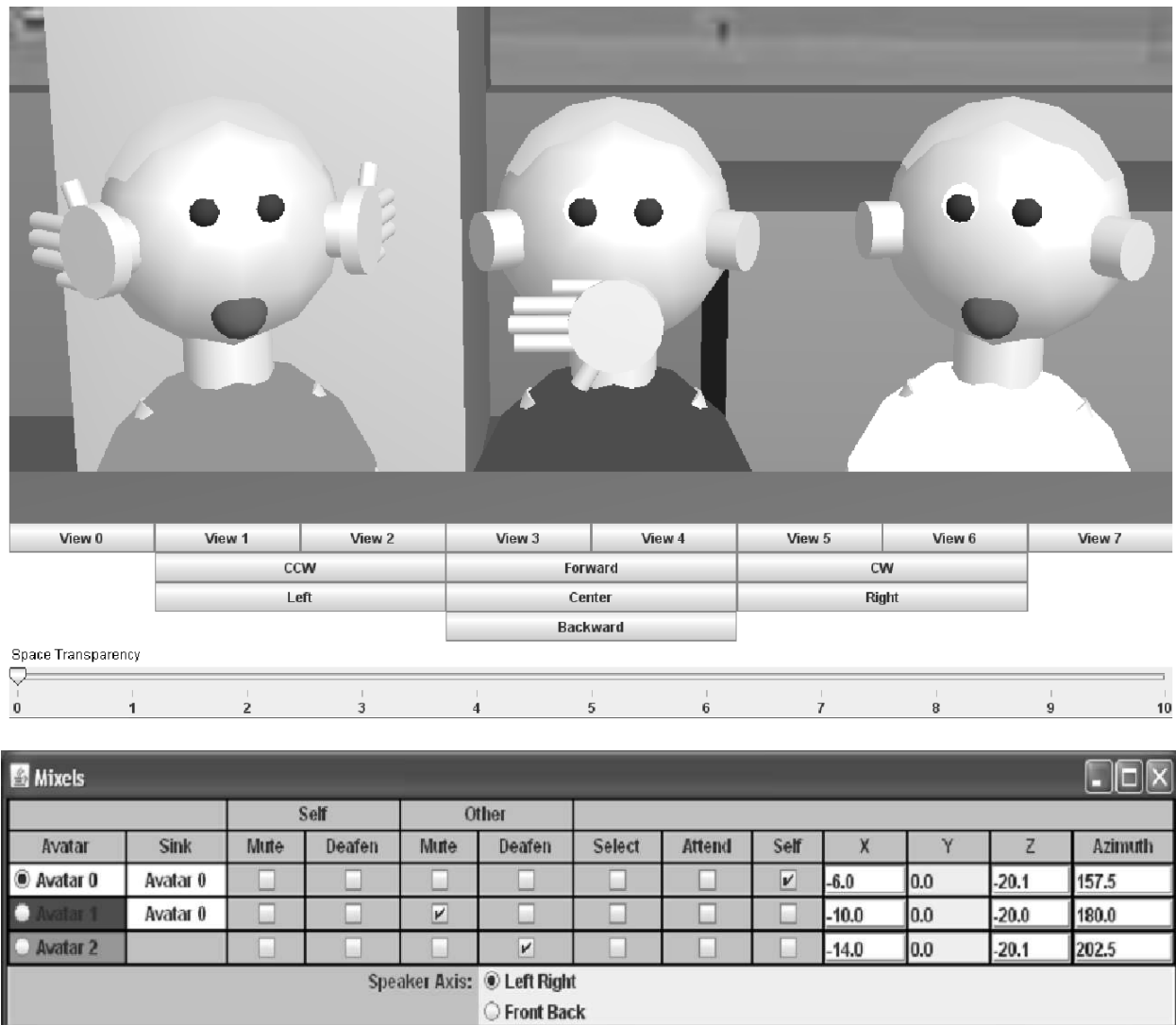


Figure 1.9: Narrowcasting Control in Virtual Environment: P_1 (avatar 0, right) mutes P_2 (avatar 1, middle) and deafens P_3 (avatar 2, left).

commands (`deafen` and `mute`), the time complexity is constant ($O(1)$), independent of the number of participants in a session. For including narrowcasting commands (`select` and `attend`), in which the connectivity state of the complement of the selection needs to be adjusted, the time complexity is $O(n)$, linear in the number of participants. The configuration for the IP Media Server used in our experiments supports up to 100 clients. Even though our laboratory testbed uses a much smaller user pool, typically about four, there is no reason not to assume that the signaling protocol can keep up with practical realtime demands and support the same number of session participants.

1.5 Conclusion and Future Research

We have described an instance of Media Server Component Model architecture for policy-based media-mixing and narrowcasting within the standard SIP framework for multimedia conferencing systems. Narrowcasting privacy commands were prototyped, including a policy evaluation algorithm, a media mixing and delivery mechanism that considers fine-grained policy configured by participants. The policy can be displayed and controlled via a 3D interface in which hands and other attributes (megaphones and ears trumpets) clapped over figurative avatars' mouths and ears represent audio stream filters. The popularity of applications like 'Second Life' extends the ways in which people interact. Such three-dimensional environments represent a fertile platform for virtual conferences, meetings, and concerts.

1.5.1 Practical Conferencing

In ordinary conversation, participants generally observe turn-taking, as in a CSMA/CD (carrier sense multiple access / collision detection) protocol with discretionary backup. That is, an utterance that collides with another will cause one or both of the simultaneous speakers to stop and wait until a break before repeating.

One might wonder what happens to such conversational turn-taking in the presence of asymmetric media filters and the absence of a moderator. Narrowcasting features— like blocklists, side channels, and call-within-a-call— complicate teleconferences, since a deafened conversant might not be aware that another is talking and multiple sources might speak at once. If some participants in a conference are muted or deafened to some other participants, without formal floor control there is a likelihood of some “talking on top of” others. In the absence of common floor control, won't private chats and decentralized control lead to anarchy? Without “traffic signals,” how can collisions be avoided?

In fact, such parallel conversation streams are not a problem. For example, if two participants set up a private side-conference using narrowcasting commands, even though their



Figure 1.10: Theme-based Discussion in Articulated Chatspace. (©2008 The New Yorker Collection from cartoonbank.com. All rights reserved.)

utterances might collide with others’, they wouldn’t expect or want others to stop conversing. Rather they “listen with one ear” to ongoing conversations while enjoying their own caucus. Listeners can still untangle conversational threads, by context, voice quality, etc., as suggested by Fig. 1.10. Just as in real social contexts, including informal gatherings like parties, multiple simultaneous speakers are analyzable. Even “linear” conversations like formal meetings might have some subsets of conversants whispering among themselves while a main speaker is talking. Narrowcasting audio interfaces are even more useful when extended by spatial sound and attenuation based on mutual virtual position (source projection, sink bearing, and distance) [11] [12], distributing the respective voices across a soundscape.

1.5.2 Presence

Presence, also known as presence information, conveys the ability and willingness of a user to communicate across a set of devices. The SIP events framework [18] defines general mechanisms for subscribing to, and receiving notifications of, events within SIP networks. It introduces the notion of a package, which is a specific instantiation of the events framework for a well-defined set of events. RFC 4575 [22] defines a conference event package which can be used by a conference notification service, as outlined in the SIP conferencing framework. As described there, subscriptions to a conference URI are routed to a focus that handles the conference. It acts as the notifier and provides clients with updates on conference state. The conference event package is not adequate enough to represent the status of the narrowcasting conference participants. Fig. 1.11 shows the conference-info document format defined in RFC 4575.

1.5.3 Architectural and Interface Refinement

Future research includes allowing selection of multiple sources and sinks for narrowcasting commands. Currently, the MSCML privacy model is too primitive and the language not expressive enough; we are exploring ways that it might be extended to support arbitrary multiuser narrowcasting configurations. We are also considering other conference models with multiple policy servers or media mixers. For instance, as multimedia processing becomes less of a specialized service and more of a commodity, a grid computing paradigm (such as that promoted by mediagrid.org) could be used instead of a centralized server architecture to mix and deliver media streams to distributed narrowcasting-enabled terminals. `Muffle` (partial `deafen`) and `muzzle` (partial `mute`) will enrich the narrowcasting state space [9] [6]. We will also generalize policy determination in metasessions with multiple simultaneous chatspaces, in which one has presence across multiple virtual spaces, each with several or many conversants, including “multipresence,” allowing designation of multiple instances of “self” [7].

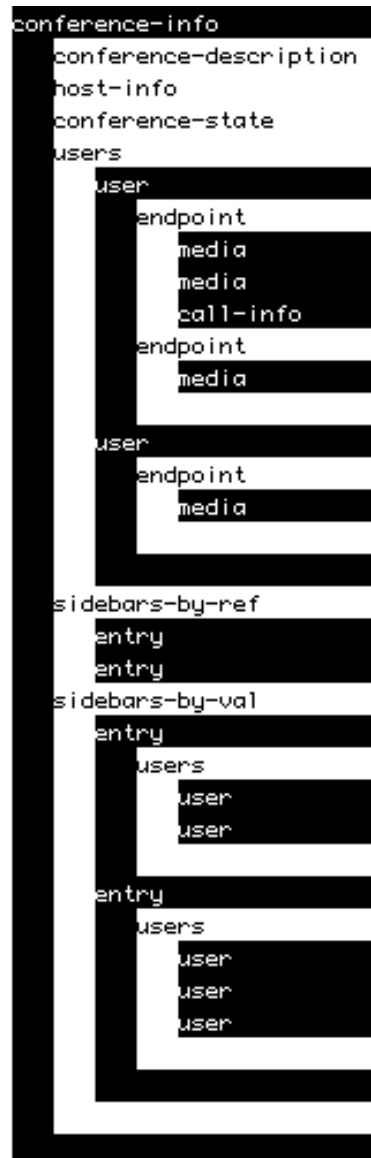


Figure 1.11: Sidebar sidebar sidebar: This separated figure (“sidebar”) uses a special outline display that graphically represents nested hierarchical elements (“sidebars”) to show the position within a database schema for conferencing where a side-channel (“sidebar”) would be specified. The `conference-info` document format shown can be extended by a newly defined event template-package. A template-package has all the properties of a regular SIP event package. It is always associated with some other event package, and can always be applied to any event package. In this case, the template-package inherits the status of the conference event package. In many cases, the information can be dynamically learned from the call signaling and can also be manually populated by an administrator— all subject to local policies. Some portions of the information are intended for processing by automata; others are for human consumption only. For example, the `<display-text>` sub-elements of elements `<conf-uris>`, `<service-uris>`, `<available-media>`, `<host-info>`, `<endpoint>`, and `<media>` (some of which are not shown above) are intended for display to human subscribers only. The template-package (perhaps named something like `conference.narrowcasting`) would define a matrix of narrowcasting status (`mute`, `deafen`, `select`, `attend`) along with the controller information.

1.5.4 Convergence

Besides wireline-connected workstation-based interfaces, narrowcasting might find an even more fertile platform in mobile devices. The ‘4-play’ convergence of telephony, television/video, internet, and wireless is driving a remarkable proliferation of new devices and services. Mobile terminals, almost as intimate as clothing, are a kind of wearable computer. A diversity of next-generation form factors for evolving smartphones is emerging, including mobile stereotelephony, spawned from cyberspatial audio [10] and augmented audio models. Meanwhile, location-based services, along with seamless handoff, FMC (fixed-mobile convergence), and heterogeneous roaming via MIMO (multiple input/multiple output) smart antennas leading to software-defined radio (SDR) and cognitive radio, leverage geolocation and portable GPS/GIS. Such advanced sensing enables ubicomp and ambient intelligence, including an awareness of user status and availability, and articulated models of privacy, like narrowcasting, that allow users to distribute their attention, availability, and virtual presence. Multipresence and persistent channels, encouraged by ABC (always best connected) networks, will extend the way people communicate.

Acknowledgments

This research has been sponsored in part by a grant from the Japan Science and Technology Foundation. We also thank Dialogic/Cantata and BEA for providing media and application servers for our experiments.

References

- [1] M. S. Alam, M. Cohen, and A. Ahmed. Articulated Narrowcasting for Privacy and Awareness in Multimedia Conferencing Systems and Design for Implementation Within a SIP Framework. In *Proc. ICME: Int. Conf. on Multimedia & Expo*, Beijing, July 2007.
- [2] M. S. Alam, M. Cohen, and A. Ahmed. Narrowcasting—Controlling Media Privacy in SIP Multimedia Conferencing. In *Proc. IEEE CCNC: 4th Consumer Communications and Networking Conf.*, Las Vegas, Jan. 2007.

- [3] M. S. Alam, M. Cohen, and A. Ahmed. Narrowcasting: Implementation of Privacy Control in SIP Conferencing. *JVRB: J. of Virtual Reality and Broadcasting*, 4(9), 2007.
- [4] L. Berc, H. Gajewska, and M. Manasse. Pssst: Side Conversations in the Argo Telecollaboration System. In *Proc. UIST: 8th Annual ACM Symp. on User Interface and Software Technology*, pages 155–156, New York, NY, Nov. 1995. ACM Press. ISBN 0-89791-709-X.
- [5] Y.-H. Cho, M.-S. Jeong, J.-T. Park, and W.-H. Lee. Distributed Management Architecture for Multimedia Conferencing Using SIP. In *Proc. DFMA: 1st Int. Conf. on Distributed Frameworks for Multimedia Applications*, Besancon, France, Feb. 2005.
- [6] M. Cohen. Throwing, pitching, and catching sound: Audio windowing models and modes. *IJMMS: the J. of Person-Computer Interaction*, 39(2):269–304, Aug. 1993. ISSN 0020-7373; www.u-aizu.ac.jp/~mcohen/welcome/publications/tpc.ps.
- [7] M. Cohen. Quantity of presence: Beyond person, number, and pronouns. In T. L. Kunii and A. Luciani, editors, *Cyberworlds*, chapter 19, pages 289–308. Springer-Verlag, Tokyo, 1998. ISBN 4-431-70207-5; www.u-aizu.ac.jp/~mcohen/welcome/publications/bi1.pdf.
- [8] M. Cohen. Exclude and include for audio sources and sinks: Analogs of mute & solo are deafen & attend. *Presence: Teleoperators and Virtual Environments*, 9(1):84–96, Feb. 2000. ISSN 1054-7460; www.u-aizu.ac.jp/~mcohen/welcome/publications/ie1.pdf.
- [9] M. Cohen. Emerging exotic auditory interfaces. In *114th Convention of the AES*, Amsterdam, Mar. 2003. Preprint #5819.
- [10] M. Cohen, J. Herder, and W. L. Martens. Cyberspatial Audio Technology. *J. Acous. Soc. Jap. (English)*, 20(6), Nov. 1999.
- [11] M. Cohen and N. Koizumi. Virtual gain for audio windows. *Presence: Teleoperators and Virtual Environments*, 7(1):53–66, Feb. 1998. ISSN 1054-7460.
- [12] M. Cohen and E. M. Wenzel. The design of multidimensional sound interfaces. In W. Barfield and T. A. Furness III, editors, *Virtual Environments and Advanced Interface Design*, chapter 8, pages 291–346. Oxford University Press, 1995. ISBN 0-19-507555-2.
- [13] H.-P. Dommel and J. Garcia-Luna-Aceves. Floor Control for Activity Coordination in Networked Multimedia Applications. In *Proc. APCC: 2nd Asian-Pacific Conference on Communications*, Osaka, Japan, June 1995.
- [14] J. V. Dyke, E. Burger, and A. Spitzer. RFC 4722— Media Server Control Markup Language (MSCML) and Protocol, Nov. 2006.
- [15] R. Even and N. Ismail. RFC 4825— Conferencing Scenarios, July 2006.
- [16] O. N. N. Fernando, K. Adachi, U. Duminduwardena, M. Kawaguchi, and M. Cohen. Audio Narrowcasting and Privacy for Multipresent Avatars on Workstations and Mobile Phones. *IEICE Trans. on Information and Systems*, E89-D(1):73–87, Jan. 2006.
- [17] P. Koskelainen, H. Schulzrinne, and X. Wu. A SIP-based Conference Control Framework. In *Proc. NOSSDAV: 12th Int. Wkshp. on Network and Operating Systems Support for Digital Audio and Video*, pages 53–61, New York, NY, 2002. ACM Press. ISBN 1-58113-512-2.
- [18] A. B. Roach. RFC 3265— Session Initiation Protocol (SIP) – Specific Event Notification, June 2002.

- [19] J. Rosenberg. RFC 4353— A Framework for Conferencing with the Session Initiation Protocol, Feb. 2006.
- [20] J. Rosenberg. RFC 4897— The Extensible Markup Language (XML) Configuration Access Protocol (XCAP), May 2007.
- [21] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. RFC 3261— SIP: Session Initiation Protocol, June 2002.
- [22] J. Rosenberg, H. Schulzrinne, and O. Levin. RFC 4575— A Session Initiation Protocol (SIP) Event Package for Conference State, Aug. 2006.
- [23] H. Sowizral, K. Rushforth, and M. Deering. *The Java 3D API Specification*. Addison-Wesley, second edition, 2000. ISBN 0-201-71041-2.
- [24] R. A. Spinello. *Cyberethics Morality and Law in Cyberspace*. Jones and Bartlett Publishers, Sudbury, 2004. ISBN 0-7637-0064-9.
- [25] A. E. Walsh and D. Gehringer. *Java 3D API Jump-Start*. Prentice-Hall, 2002. ISBN 0-13-034076-6.
- [26] C. C. Wang, J. Cahnbley, and J. W. Richardson. US Patent—Method and System for Providing a Private Conversation Channel in a Videoconference System, June 2003. Publication No. WO 2003/053034 A1.
- [27] N. Yankelovich, J. McGinn, M. Wessler, J. Kaplan, J. Provino, and H. Fox. Private communications in public meetings. In *Proc. CHI: Human Factors in Computing Systems*, pages 1873–1876, Portland, OR, Apr. 2005. ACM Press. ISBN 1-59593-002-7.

List of Figures

1.1	Media Privacy: A Call Center Application Scenario	4
1.2	Typical Conference Architecture: F is the focus, M are media mixers, and P_i are the participants.	5
1.3	Privacy: Freedom from disturbance.	7
1.4	Media Privacy (Narrowcasting Features)	8
1.5	Formalization of narrowcasting and selection functions in predicate calculus notation, where ' \neg ' means "not," ' \wedge ' means conjunction (logical "and"), ' \exists ' means "there exists," ' \Rightarrow ' means "implies," and ' \Leftrightarrow ' means mutual implication (equivalence).	9
1.6	Media Server Component Model with Collaborative Virtual Environment Integration	13
1.7	Media Mixing and Delivery (P_1 mutes P_2 and deafens P_4)	15
1.8	Communication Flow Between SIP Entities: A default configuration (1.) establishes a normal session (2.), but it can be adjusted (3.) to reconfigure (4.) the mixes returned to the participants (5.).	16
1.9	Narrowcasting Control in Virtual Environment: P_1 (avatar 0, right) mutes P_2 (avatar 1, middle) and deafens P_3 (avatar 2, left).	19

1.10	Theme-based Discussion in Articulated ChatSPACE.	21
1.11	<p> Sidebar sidebar sidebar: This separated figure (“sidebar”) uses a special outline display that graphically represents nested hierarchical elements (“sidebars”) to show the position within a database schema for conferencing where a side-channel (“sidebar”) would be specified. The <code>conference-info</code> document format shown can be extended by a newly defined event template-package. A template-package has all the properties of a regular SIP event package. It is always associated with some other event package, and can always be applied to any event package. In this case, the template-package inherits the status of the conference event package. In many cases, the information can be dynamically learned from the call signaling and can also be manually populated by an administrator— all subject to local policies. Some portions of the information are intended for processing by automata; others are for human consumption only. For example, the <code><display-text></code> sub-elements of elements <code><conf-uris></code>, <code><service-uris></code>, <code><available-media></code>, <code><host-info></code>, <code><endpoint></code>, and <code><media></code> (some of which are not shown above) are intended for display to human subscribers only. The template-package (perhaps named something like <code>conference.narrowcasting</code>) would define a matrix of narrowcasting status (<code>mute</code>, <code>deafen</code>, <code>select</code>, <code>attend</code>) along with the controller information. </p>	23

List of Tables

1.1	Three different Mute operations	3
1.2	Conferencing Models: ‘P’ indicates participant, ‘F’ indicates focus, ‘M’ indicates media mixer, and (in the last model) ‘PF’ indicates Primary Focus. Dotted lines indicate signaling, dashed lines indicate media transmission, and solid lines indicate mixer control.	6
1.3	Narrowcasting Commands	10
1.4	Policy Matrix $P = [p_{ij}]$	14
1.5	MSCML Configuration: P_1 deafens P_2	16