

Technical Disclosure Commons

Defensive Publications Series

August 29, 2019

Private Conversion During a Conference Using Stereo Audio Mixing Capabilities

Jeremy ZRIHEN
ALE International

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

ZRIHEN, Jeremy, "Private Conversion During a Conference Using Stereo Audio Mixing Capabilities", Technical Disclosure Commons, (August 29, 2019)
https://www.tdcommons.org/dpubs_series/2452



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Docket Number	FR82019002
Title	“Private conversion during a conference using stereo audio mixing capabilities”
Contributor	Jeremy ZRIHEN
Company	ALE International

Description of the technical solution:

A. INTRODUCTION

During a conference, we want to have the possibility to initiate a discreet conversation with another participant of the conference with following facilities.

- As a participant in the conference, I can hear at the same time the conference and my private discussions.
- Only participant(s) of my private discussion can hear me.

This solution can be implicitly extended with all following use cases:

- I can manage several private discussions in parallel and I can come back and interact at any time in the conference.
- I can merge several private discussions in one private conference - so all participants of each private discussion will have the opportunity to have discussion in a private conference and to continue to ear main conference discussion.
- Optionally, I can also adjust the audio level and the spatial position of each of the two audio streams representing each discussion – the conference and the private discussion.
- The solution also gives me the possibility to end my participation in each private discussion where I am involved.

When conferencing system implements this concept, the solution is available for all participants with an audio device also implementing necessary commands to drive it in the conferencing system.

This solution avoids therefore the main drawbacks of current alternatives; for example, when a user is in an audio conference, and if he wants to get discreetly some information from another participant of the conference, it is only possible to use external channels such as:

- Chat or E-mail, but voice exchange is not possible in this case.
- Initiate another media communication with another audio/video device without the possibility to hear the two discussions at the same time.

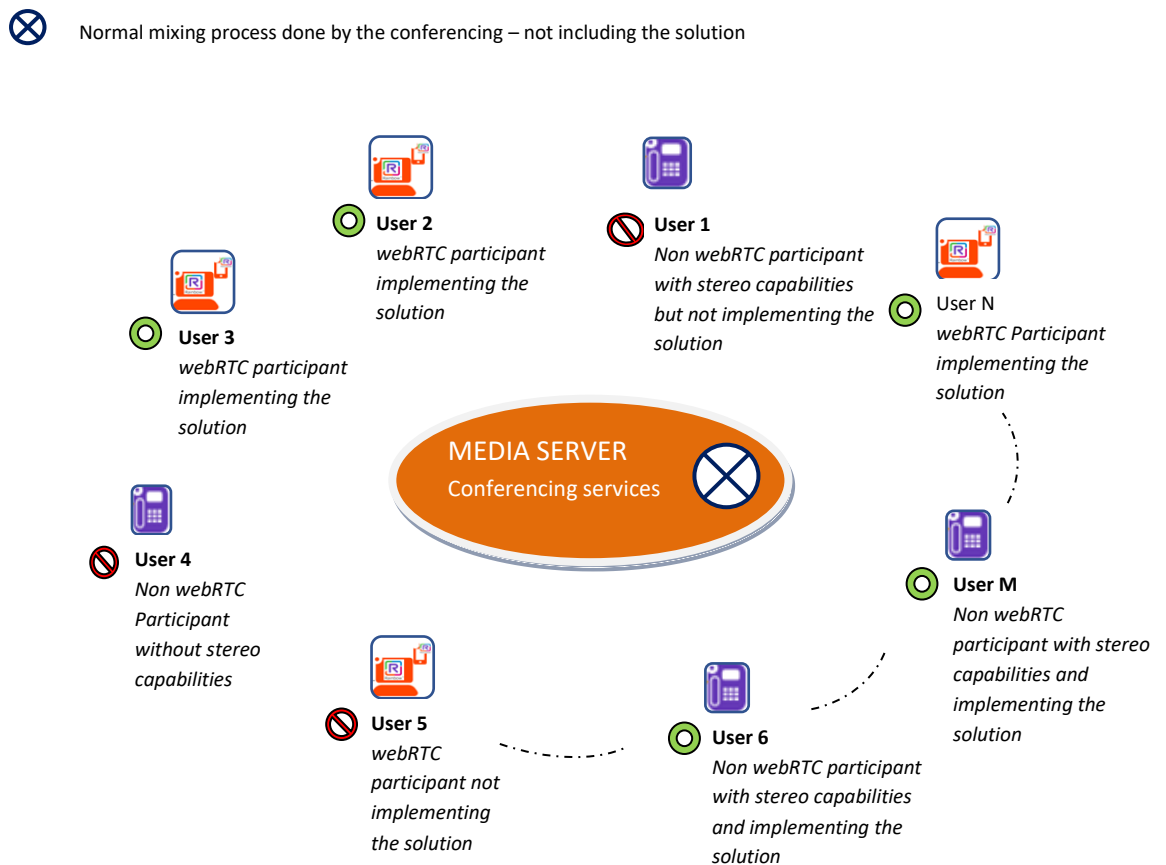
The proposed solution allows to have a comfortable solution limiting impacts when participant of a conference needs to have discreet discussion with another participant.

This solution is available for all participant of a conference as soon as conferencing system and audio devices of the participants implement it.

In the example below, N users participate in a conference using conferencing system with stereo capabilities and implementing the solution.

Some of the participants use webRTC communication facilities with stereo capabilities and implementing the solution (browser plugin). Those participants will benefit of a specific API on the webRTC application that allows them to call use cases proposed by this solution

Some other participants use proprietary softphone, SIP deskphone, mobile application or other type of devices to connect to the conference. Those participants will benefit from the solution only if audio device has stereo capabilities and implement the solution.

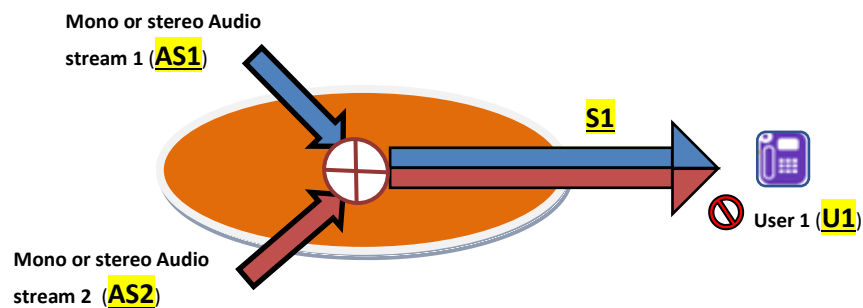


B. THE SOLUTION

Core of this solution is to use stereo capabilities of a conferencing system to give to end users the ability to receive several streams of discussion at the same time, and to be able to distinguish them by using spatialization effects.

For example, in the scheme below, we can see that a mixing stereo output **S1** for user **U1** can contain a spatial rendering of the two audio inputs **AS1** and **AS2**. Independently with input streams mode (mono or stereo), it is always possible for mixing processes to render the two audio streams information in a single stereo output **S1** using spatialization effects.

⊕ Supplementary mixing process and audio rendering proposed by the solution



This stereo mixing process, that consists on the mix of two input audio streams to generate a single stereo stream with a spatial separation of the two inputs, needs to be distinguished with the classical mixing process done by a conferencing server and that acts in the way to render an audio containing the addition of the main actors of the conference (speakers with the higher level).

The idea to add this mixing process in the way to elaborate private conversation in a conferencing server is the main purpose of this concept.

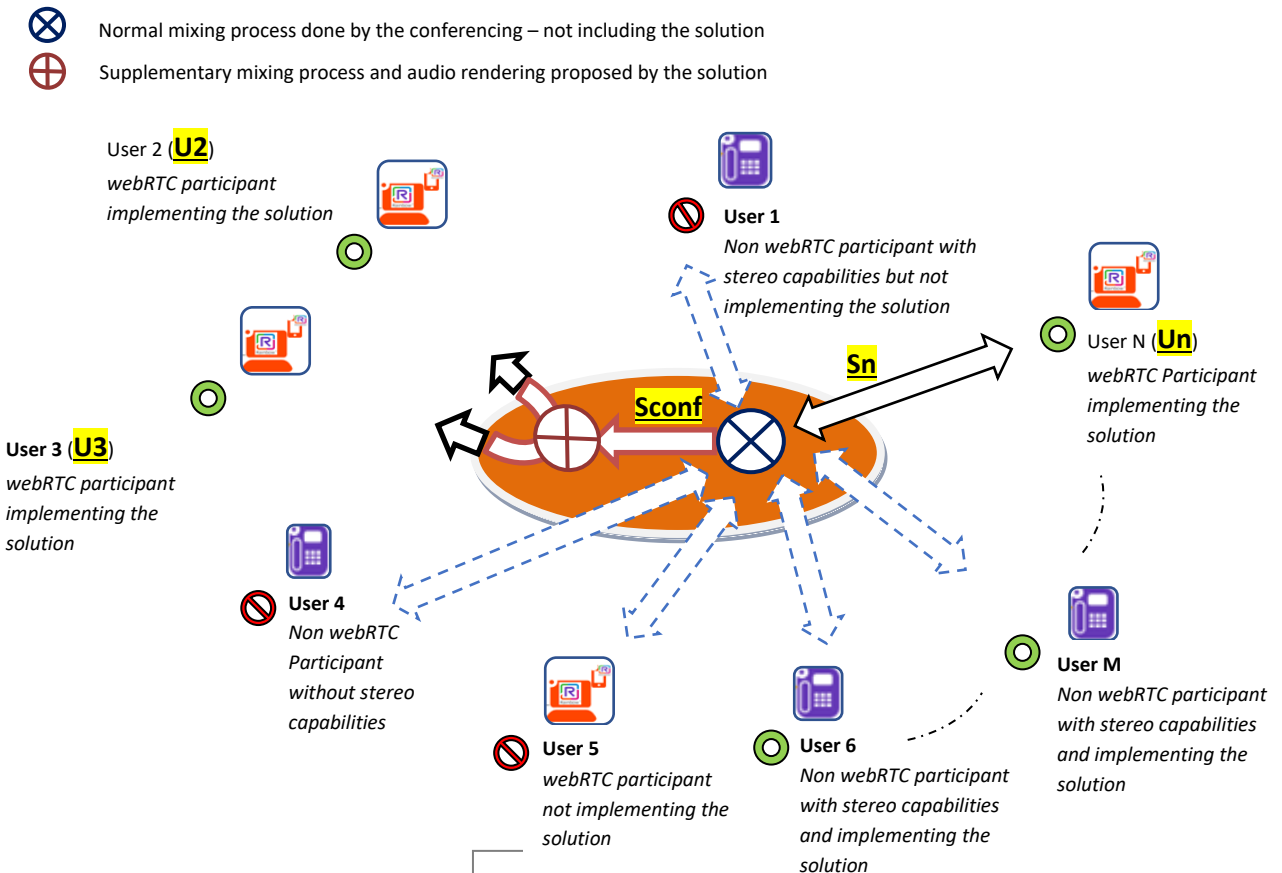
Considering a conferencing server that implements such stereo mixing capabilities, we can easily imagine that each end user audio devices can be able to:

- Drive conferencing server for initiating some private communications, and still listen to the main conference discussion.
- Ensure that, at the same time, users that are not in the private conversation cannot hear the audio of the private conversation.
- Allow private conversation users to switch at any time between the main conference and the private discussion. End user can also modify level and spatial position of each audio stream.



For that, the solution proposes to add the following 3 stereo mixing capabilities in the conferencing server.

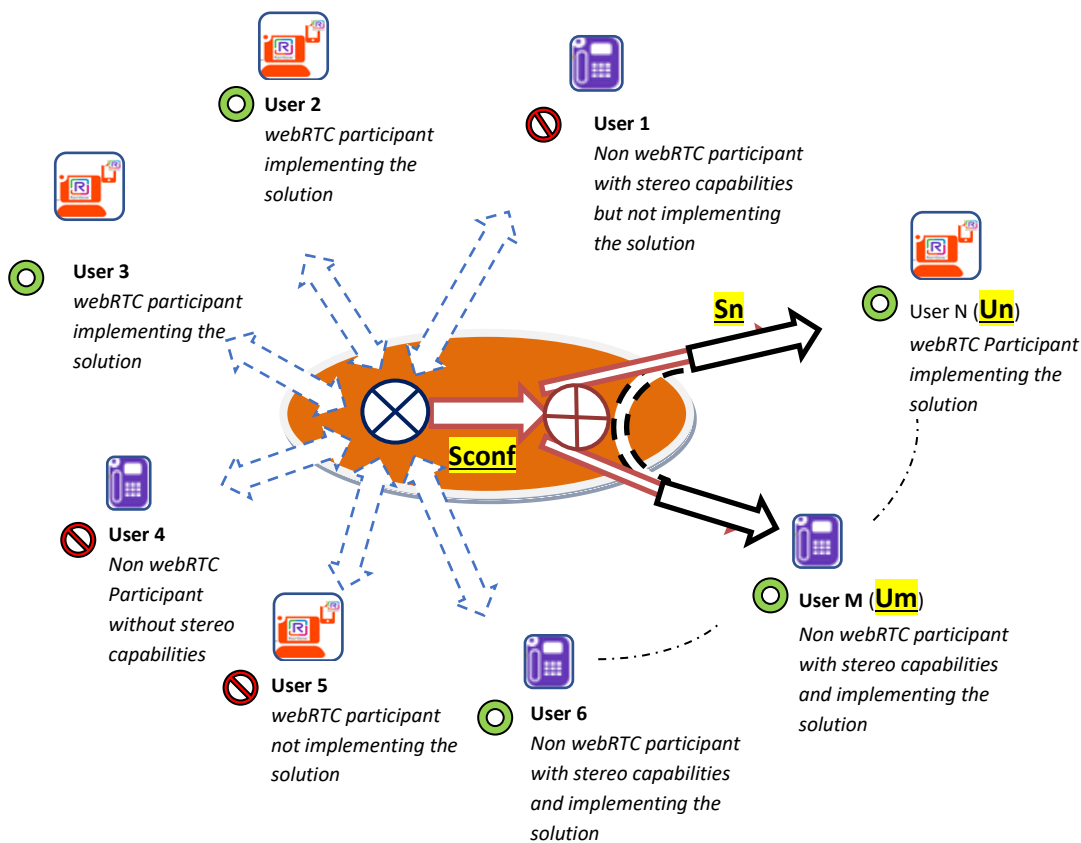
1. Mixing processes done by the conferencing server for a user (**Un**) that does not participate in any active private communication and that benefits from stereo capabilities will produce one stereo audio stream. This stereo audio stream (**Sn**) will be the simple stereo restitution of the mixing process already done by the conferencing server in a normal case but not including all users currently in an active private conversation. In this example audio streams of users **U2**, and **U3** are independent with the main conference and so not included in (**Sn**).

In a general way, if several users are active in several private communications, all of them will not be included in the main conference, but will have the possibility to ear audio of the conference at the same time of their own private conversation.



2. Mixing process done by the conferencing server for a user (U_n) actively in a private communication with user (U_m) will produce one stereo audio stream (S_n). This stereo audio stream is the restitution of a spatialized separation of the two-following audio streams:
 - The mixing output already processed by the conferencing server in a normal case (S_{conf}).
 - The restitution of the voice of the user (U_m) currently active in the private conversation.

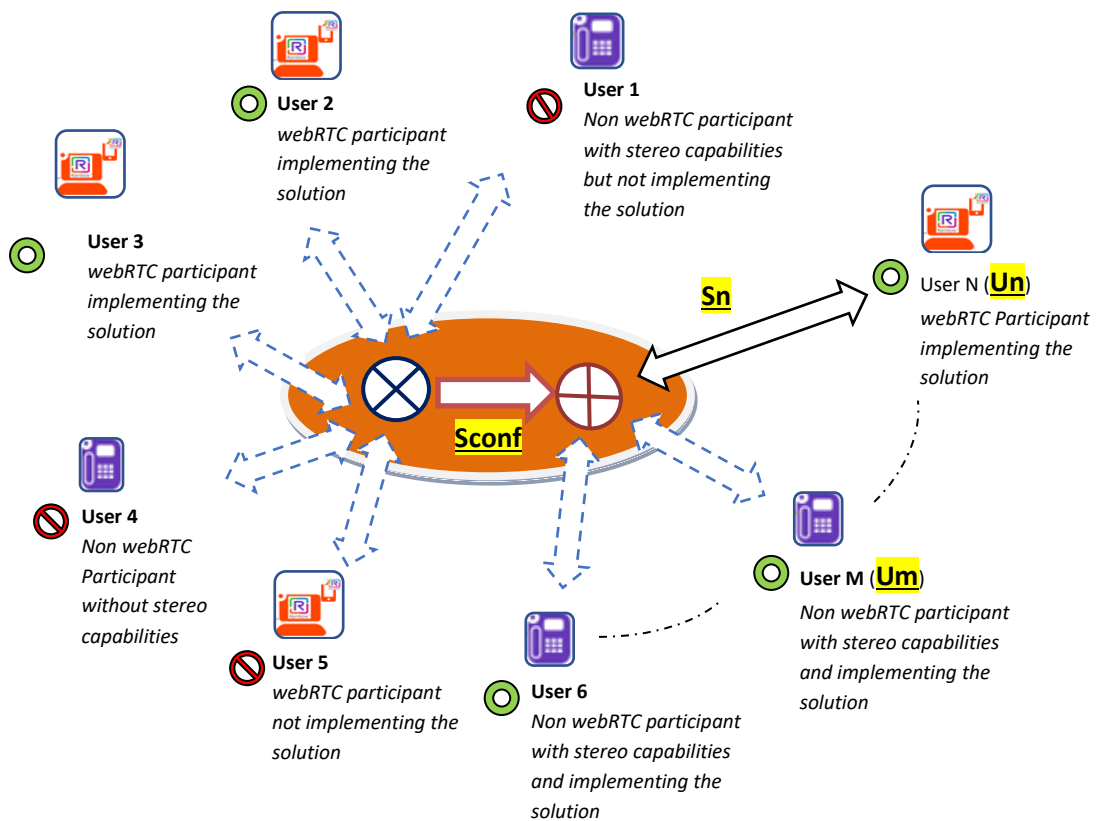
-  Normal mixing process done by the conferencing – not including the solution
-  Supplementary mixing process and audio rendering proposed by the solution



3. A private conversation can include more than two users and can be designed as a “private conference”. In this case, mixing process done by the conferencing server for a user (**Un**) active in a private conference will produce one stereo audio stream (**Sn**). This stereo audio stream is the restitution of a spatialized separation of the two following audio stream:

- The mixing process already done by the conferencing server in a normal case (**Sconf**).
- The result of another mixing process done with, as input, all other audio streams of users currently active in the private conversation. In the example below user (**U6**) and user (**Um**) are participant of the private conference.

- ⊗ Normal mixing process done by the conferencing – not including the solution
- ⊕ Supplementary mixing process and audio rendering proposed by the solution



C. **IMPLEMENTATION**

This section details the implementation that can be proposed for a conferencing system (conferencing server + end user audio devices) to benefit from the solution.

First, conferencing server must implement mixing processes exposed in the previous section. Those capabilities will be driven by end user using specific API exposed by the conferencing server.

Conferencing server and audio communication devices implementing this solution must have capabilities to exchange some specific messages to interact with. In many systems, a protocol is already existing to initiate media communication between end user audio devices and conferencing system, and allows to transport some information to interact with. SIP is for example the most popular one and offer mandatory facilities for the solution.

Those facilities will be used to allow some new interactions between each audio communication devices implementing the solution and the conferencing system.

A Man Machine Interface (MMI) on audio communication devices implementing the solution allows end users to benefit from the solution giving the possibility to trigger following additional use cases and scenarios:

- **UC1: REQUEST PRIVATE COM:**
 - Request to initiate a (or join existing) private communication sent to another participant of the main conference
- **UC2: ACCEPT PRIVATE COM / DENY PRIVATE COM:**
 - Accept or deny a request to join a private communication, and initiate the private stream with the requester
- **UC3: MODIFY AUDIO LEVEL MAIN CONF:**
 - Modify level of the main conference audio stream
- **UC4: MODIFY AUDIO LEVEL PRIVATE COM:**
 - Modify level of a private communication audio stream
- **UC5: MODIFY POSITION MAIN CONF:**
 - Modify spatial position of the main conference audio stream
- **UC6: MODIFY POSITION PRIVATE COM:**
 - Modify spatial position of a private communication audio stream
- **UC7: SWITCH TO MAIN CONF:**
 - Switch my own voice stream to interact with main conference
- **UC8: SWITCH TO PRIVATE COM:**
 - Switch my own voice stream to interact in private discussion or private conference
- **UC9: MERGE PRIVATE COM:**
 - Merge two private discussion and/or conference in only one.
- **UC10: EXIT PRIVATE CONV:**
 - End an active private communication, if any

An implementation of this solution can consist on following scenarios / sequence diagram:

[1] REQUIRE ESTABLISHMENT OR ENDING OF A PRIVATE COMMUNICATION

The four following commands / use case can be implemented in the end user application to manage start or stop of a private communication.

REQUEST_PRIVATE_COM:

Initiates a REQUEST for private communication between two participants of a conference.

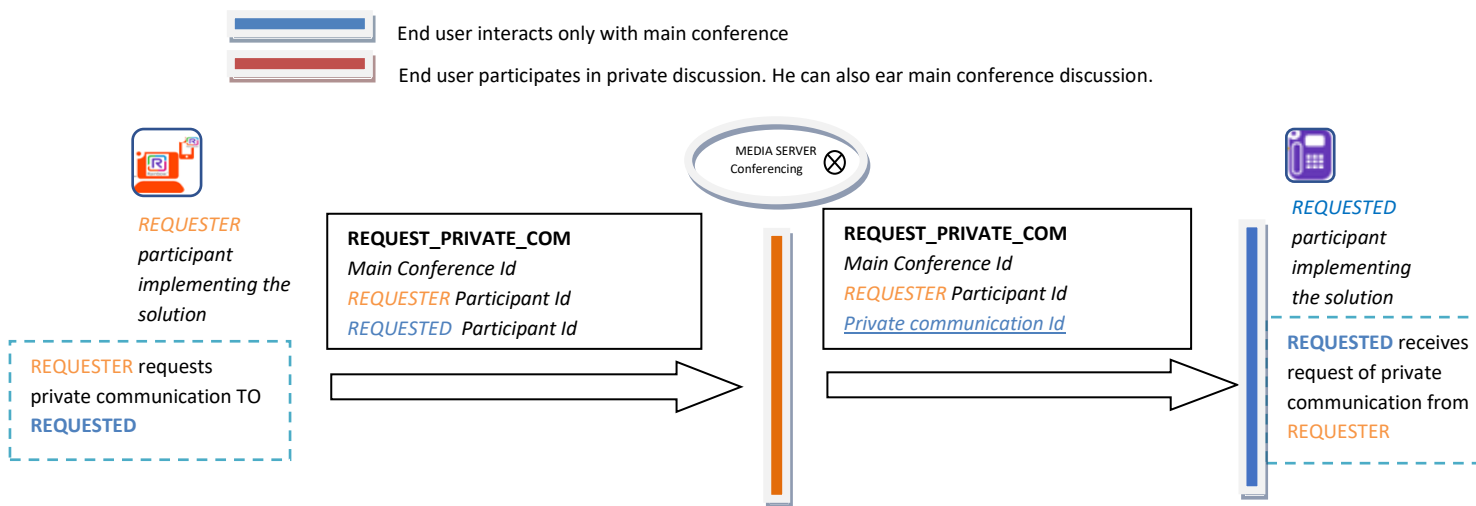
Parameters of the command are following:

- Main Conference Id
- REQUESTER Participant Id
- REQUESTED Participant Id

To establish a private communication with another user of the conference, audio communication devices of the user that wants to initiate the private discussion (**REQUESTER**) and the user that will receive the request (**REQUESTED**) must implement the solution.

At this effect, all audio communication device MMI implementing the solution must give the capability to **REQUESTER** to request for a private communication with any of all users implementing this solution.

When **REQUESTER** choose to request for a private communication, *application send a "REQUEST_PRIVATE_COM" message to REQUESTED as illustrated in the below sequence diagram;*



REQUEST_PRIVATE_COM message is send to conferencing system that will relay it to REQUESTED end user device adding a parameter to indicate associated private communication Id. This Id will be reused during all exchanges of messages to manage associated private conference.

Once final message is received by the *REQUESTED* end user device, an alert is displayed (pop up, avatar of requester that change or blink ...) and *REQUESTED* user can accept or deny the request.

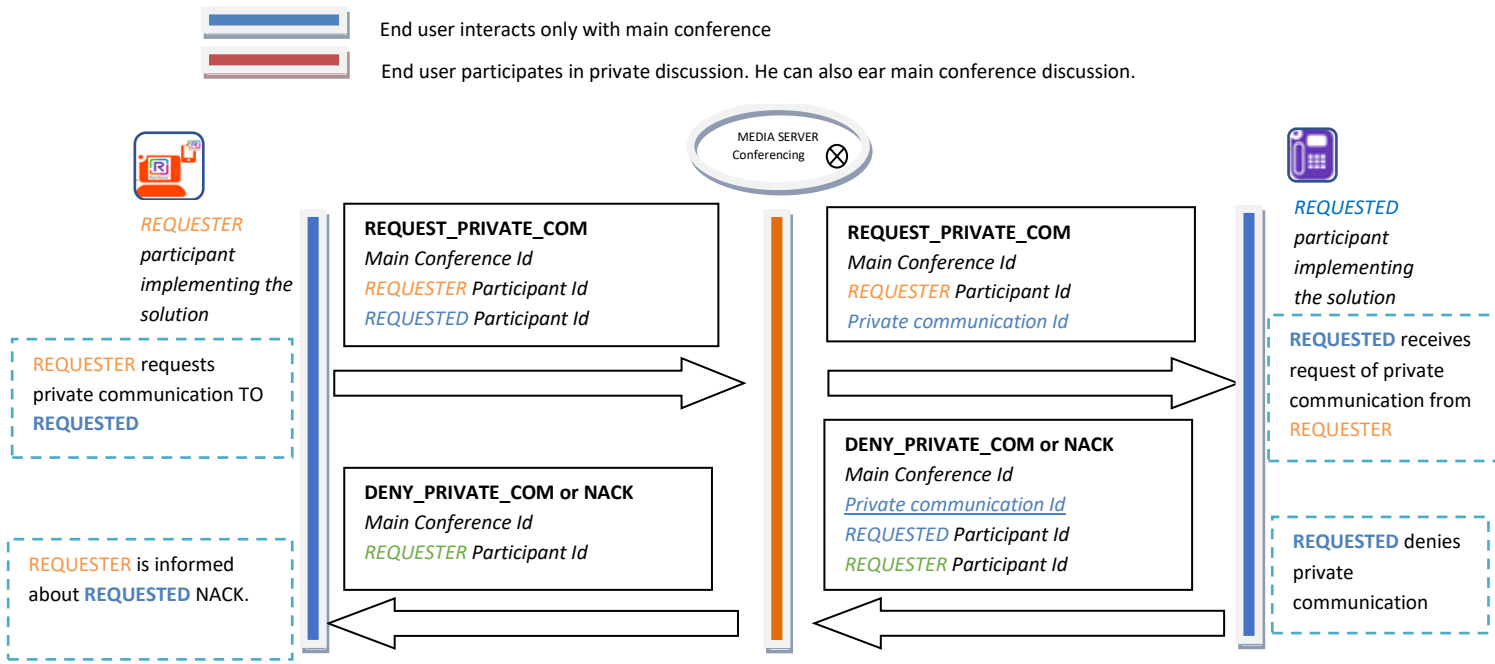
DENY_PRIVATE_COM:

Denies a request received to initiate a private communication in a conference

Parameters of the command are following:

- *Main Conference Id*
- *Private communication Id*
- *REQUESTER Participant Id*
- *REQUESTED Participant Id*

As exposed in sequence diagram below, If *REQUESTED* user denies the *REQUEST_PRIVATE_COM* from *REQUESTER* user, application replies a *DENY_PRIVATE_COM* or *NACK* the request. This information is send to conferencing server, and relayed to *REQUESTER*. No modification of the mixing process of the conferencing server are necessary in this case.



ACCEPT_PRIVATE_COM:

Accepts a request received to initiate a private communication in a conference

Parameters of the command are following:

- *Main Conference Id*
- *Private communication Id*
- *REQUESTER Participant Id*
- *REQUESTED Participant Id*

As exposed in scheme below, if a **REQUESTED** user accepts the **REQUEST_PRIVATE_COM** from a **REQUESTER** user, application replies with an acknowledgement or an **ACCEPT_PRIVATE_COM** or **ACK** message including the *Private Communication Id* information. This information is sent to conferencing server, and relayed to **REQUESTER**. Modification of the mixing process in the conferencing server must be considered to build the private connection of the **REQUESTED** with the **REQUESTER** – Refer to section B of this document for more details.



EXIT_PRIVATE_COM:

Allows end users associated with one or several private communications to end each of them.

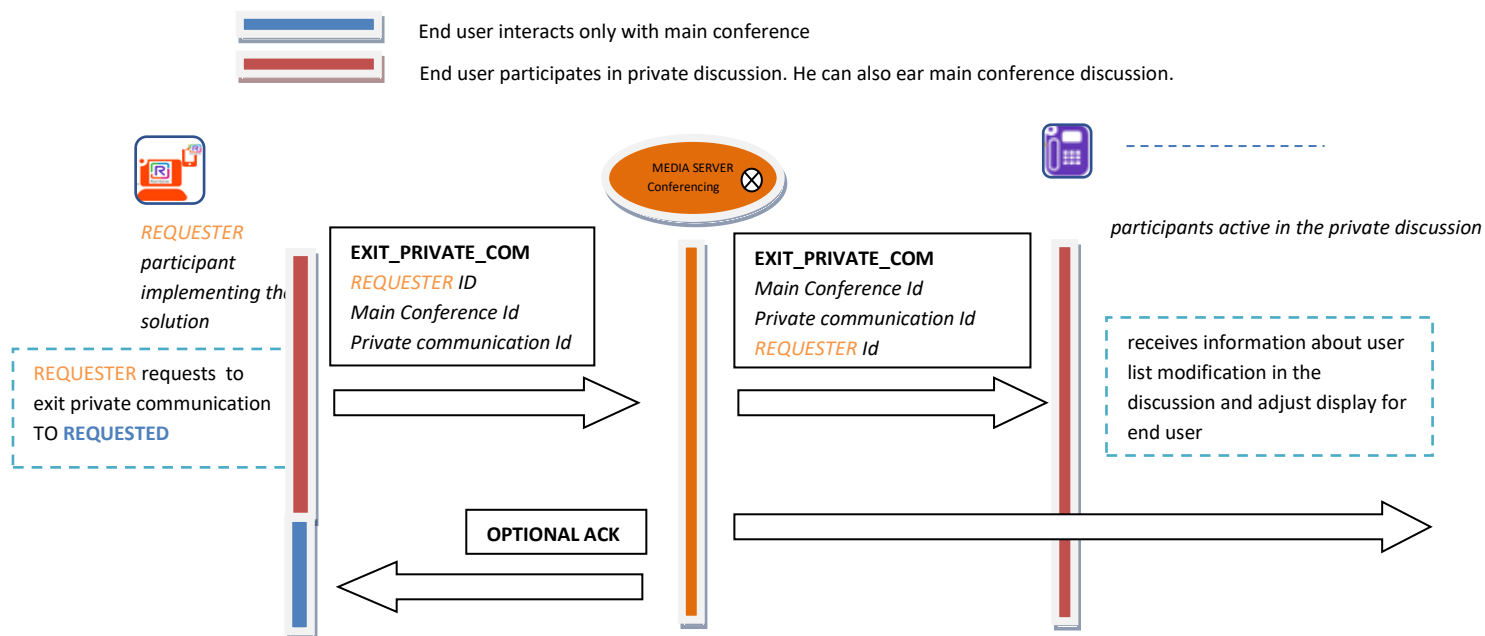
Parameters of the command are following:

- *REQUESTER Participant Id*
- *Main Conference Id*
- *Private communication Id*

As described in sequence diagram below, the **REQUESTER** application sends a message to inform the conferencing server about the requirement of removal in the private discussion.

Conference server adapts mixing characteristics to answer **REQUESTER** removal in the discussion – Refer to section B of this document for more details.

Conferencing server also informs all end user application involved in the discussion that **REQUESTER** has exited the private discussion.



[2] SWITCH MIXING CONNECTIONS

Following use cases are commands that end user audio communication device can implement to drive the switch of participation of end user in the main conference, or in each private communication where he is involved.

SWITCH_TO_CONF

Allows a user to switch from a private communication to the main conference.

Parameters of the command are following:

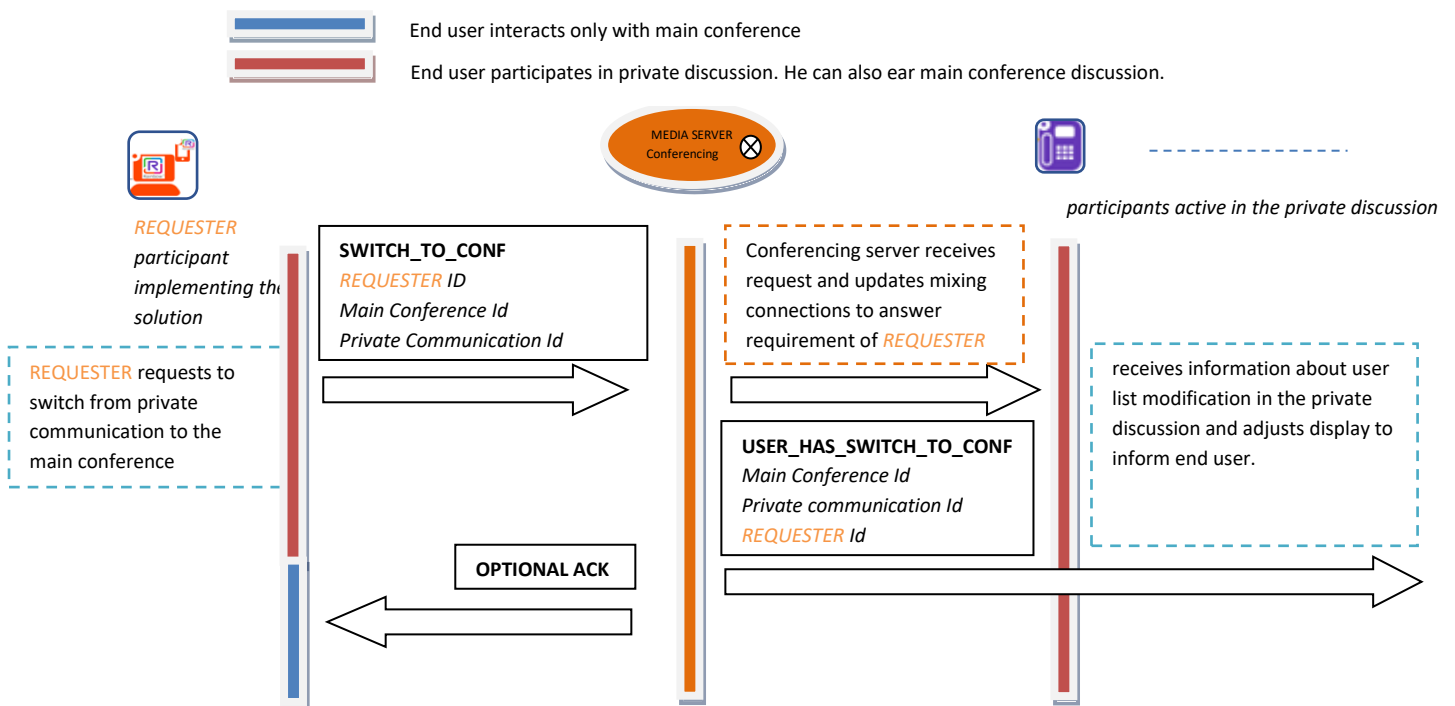
- *REQUESTER Participant Id*
- *Main Conference Id*
- *Private Communication Id - targeted Communication Id*

In sequence diagram below, the *REQUESTER* is initially connected in a private conversation and asks to switch to the main conference.

Once command received, conference server adapts mixing connections to answer switch request – Refer to section B of this document for more details.

Conferencing server also inform all participants of the private discussion about the switch using a specific *USER_HAS_SWITCH_TO_CONF* message.

No relay of the information is done to other end users. This to ensure the confidentiality of the private discussion.



SWITCH_TO_PRIVATE_COM

Allow a user to switch from the main conference to a private communication.

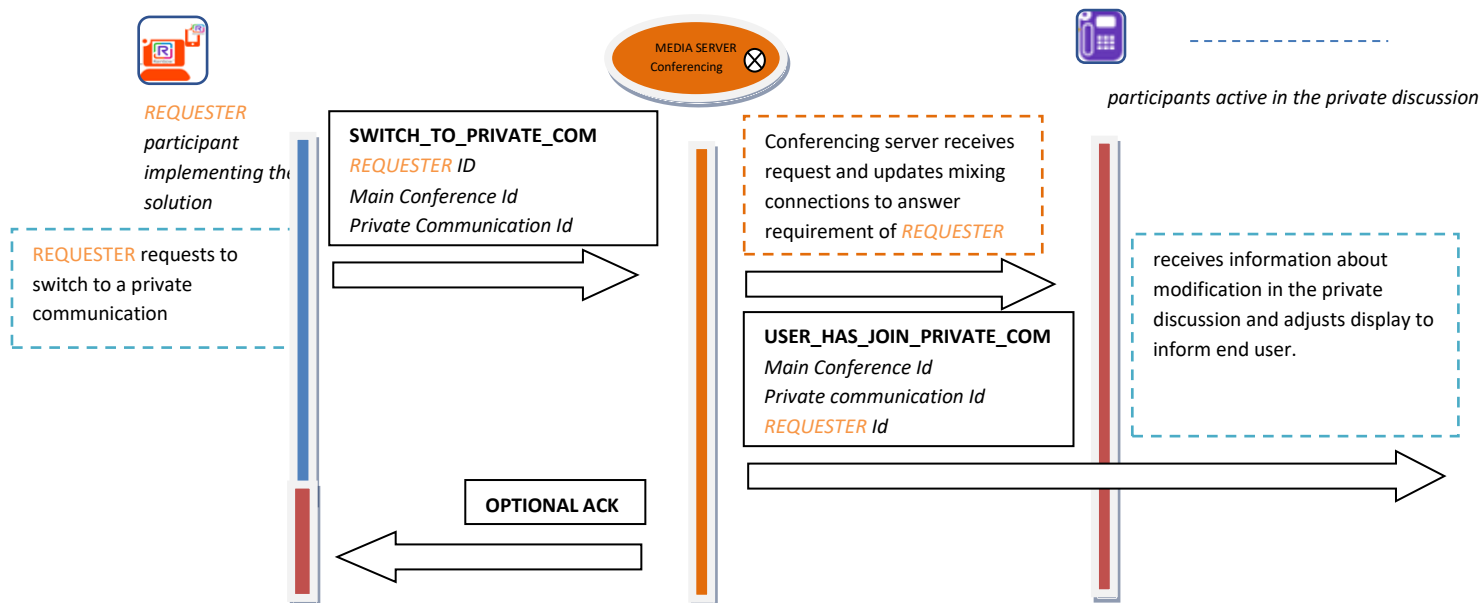
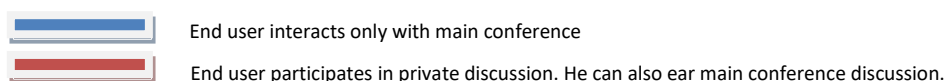
Parameters of the command are following:

- *REQUESTER Participant Id*
- *Main Conference Id*
- *Private Communication Id - targeted Communication Id*

Sequence diagram below illustrates the case where a *REQUESTER* is initially connected to the main conference and asks to switch to a private communication where he is already register. Once command received, conference server adapts mixing connections to answer switch request – Refer to section B of this document for more details.

Conferencing server also inform all participants of the private discussion about the switch using a specific USER_HAS_JOIN_PRIVATE_COM message.

No relay of the information is done to other end uses. This to ensure the confidentiality of the private discussion.



We can also imagine the possibility for a participant involved in several private communications, to join them together.

MERGE_PRIVATE_COM

Merge two private discussions and/or conference in only one.

Parameters of the command are following:

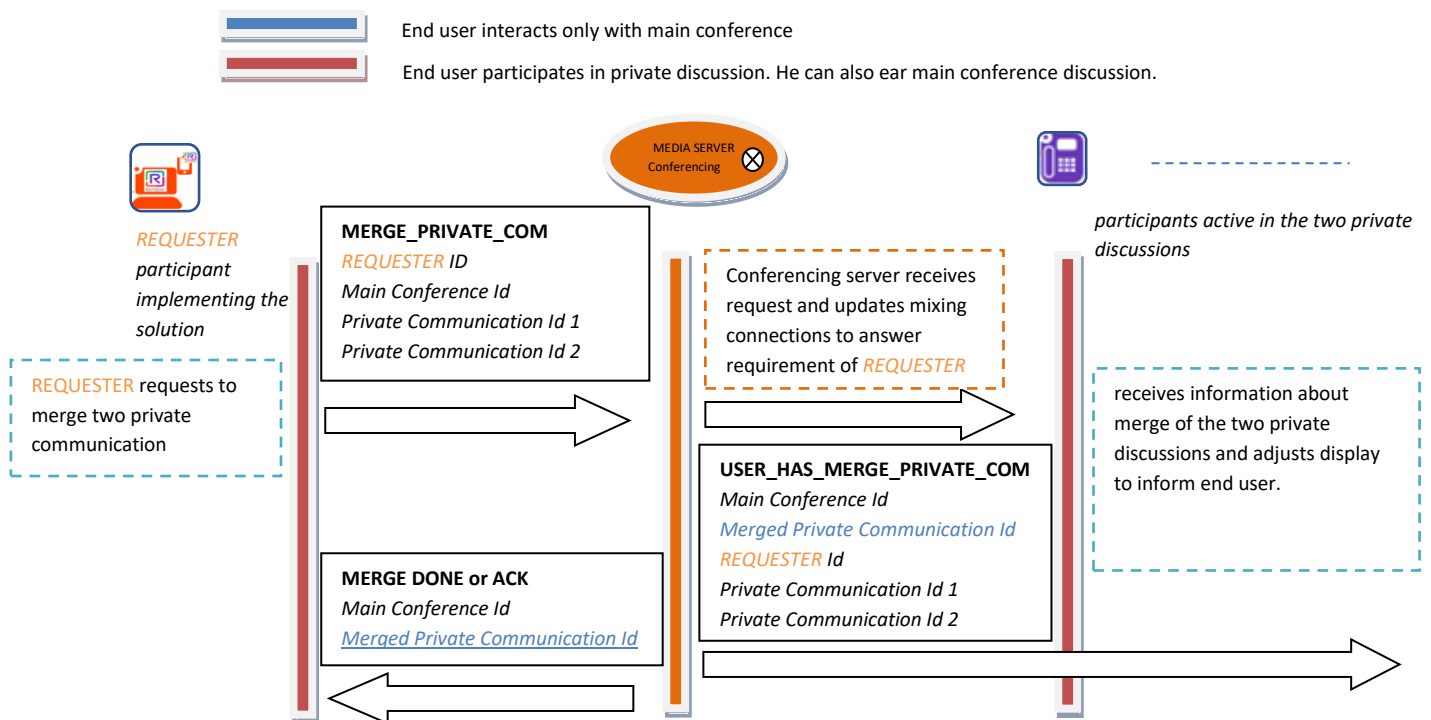
- *REQUESTER Participant Id*
- *Main Conference Id*
- *Private communication Id 1*
- *Private communication Id 2*

In sequence diagram below, the **REQUESTER** that is register in several private discussions can ask conferencing server to merge the private discussion where he is active with another private discussion where he is registered.

Once command received, conference server adapts mixing connections to answer the merge request – Refer to section B of this document for more details.

Conferencing server informs all participants of the two private discussions using a specific **USER_HAS_MERGE_PRIVATE_COM** message that contains a Merged Private Communication Id.

In the same way, conferencing server also answer with a **MERGE_DONE** or a **ACK** to inform **REQUESTER** about the success of the command and to give him the Merged Private Communication Id.



[3] MANAGE AUDIO RENDERING

Following use cases are commands send from end user audio communication device to conferencing server.

Those commands allow to manage audio rendering (level and spatial position) of the main conference and of any private discussion where end user is involved.

MODIFY_AUDIO_LEVEL_MAIN_CONF:

Modify level of the main conference audio stream.

Parameters of the command are following:

Required audio level

MODIFY_AUDIO_LEVEL_PRIVATE_COM:

Modify level of a private communication audio stream.

Parameters of the command are following:

Communication ID - created when a user accepts a private communication
(*ACCEPT_PRIVATE_COM*)

Required audio level

MODIFY_POSITION_MAIN_CONF:

Modify spatial position of the main conference audio stream

Parameters of the command are following:

Required position – xyz spatial adaptation to do from current position

MODIFY_POSITION_PRIVATE_COM:

Modify spatial position of a private communication audio stream

Parameters of the command are following:



Communication ID - created when a user accepts a private communication
(*ACCEPT_PRIVATE_COM*)

Required position – xyz spatial adaptation to do from current position

They are implemented only for end users when they are using a private communication, and allow to drive the audio level or the spatial position of one private communication, or the main conference.

When received by conferencing server, mixing processes is adapted to consider new required value.

Sequence diagram below illustrate commands where cknowledgement of commands is optional:

-  End user interacts only with main conference
-  End user participates in private discussion. He can also ear main conference discussion.

