AUDIO PROCESSING METHOD, CLOUD DESKTOP SERVER, CLIENT, AND SYSTEM

[0001] The present application claims priority to Chinese Patent Application No. 202310033481.5, entitled "AUDIO PROCESSING METHOD, CLOUD DESKTOP SERVER, CLIENT, AND SYSTEM" and filed with the China National Intellectual Property Administration on January 10, 2023, which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] The present application relates to the field of computer technologies and, in particular, to an audio processing method, a cloud desktop server, a client, and a system.

BACKGROUND

[0003] In some scenarios, users can log in to a cloud desktop server via a client to conduct an audio and video meeting through the cloud desktop server.

[0004] In the related art, a plurality of clients can be supported to log in to the cloud desktop server. However, since there is typically one recording channel between the cloud desktop server and the clients, the audio of one of the clients can be obtained through the recording channel each time, and the audio of the plurality of clients cannot be obtained concurrently, resulting in a poor user experience.

SUMMARY

[0005] Multiple aspects of the present application provide an audio processing method, a cloud desktop server, a client, and a system, to improve the usage experience of users.

[0006] In a first aspect, an embodiment of the present application provides an audio processing method, applied to a cloud desktop server, where a plurality of clients access the

cloud desktop server, and a recording channel is provided between each client and the cloud desktop server, and the method includes:

receiving, through recording channels between the plurality of clients and the cloud desktop server, at least one piece of first audio sent by the plurality of clients;

receiving second audio sent by a meeting server, where the second audio is obtained by the meeting server processing audio from at least one meeting terminal;

determining, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and the meeting terminal, and sending the corresponding mixed audio to each client and the meeting server.

[0007] In a second aspect, an embodiment of the present application provides an audio processing method, applied to a cloud desktop server, where the cloud desktop server is configured to connect to a plurality of clients, and the method includes:

receiving, through recording channels between the plurality of clients and the cloud desktop server, at least one piece of first audio sent by the plurality of clients;

receiving second audio sent by a meeting server;

determining, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and a meeting terminal, and sending the corresponding mixed audio to each client and the meeting server.

[0008] In a possible implementation, the determining, according to the at least one piece of first audio and the second audio, the mixed audio corresponding to each client and the meeting terminal, and sending the corresponding mixed audio to each client and the meeting server includes:

performing mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and sending the corresponding first mixed audio to each client;

performing mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server, and sending the second mixed audio to the meeting server.

[0009] In a possible implementation, for any client, the performing mixing processing on the at least one piece of first audio and the second audio to obtain the first mixed audio

corresponding to the client includes:

determining a first client identifier of the client and a client identifier corresponding to each piece of first audio;

determining target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio, where a client identifier corresponding to the target audio is different from the first client identifier;

performing mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

[0010] In a possible implementation, the receiving, through the recording channels between the plurality of clients and the cloud desktop server, the at least one piece of first audio sent by the plurality of clients includes:

acquiring channel statuses of the recording channels between the cloud desktop server and the plurality of clients, where the channel statuses are a muted state or an unmuted state;

determining at least one target recording channel from the recording channels according to the channel statuses of the plurality of recording channels, where the at least one target recording channel is in the unmuted state;

receiving the at least one piece of first audio through the at least one target recording channel.

[0011] In a possible implementation, the method further includes:

receiving a mute request sent by a first client of the plurality of clients, and setting the channel status of the recording channel between the first client and the cloud desktop server to the muted state according to the mute request; or,

receiving an unmute request sent by the first client, and setting the channel status of the recording channel between the first client and the cloud desktop server to the unmuted state according to the unmute request.

[0012] In a possible implementation, before receiving, through the recording channels between the plurality of clients and the cloud desktop server, the at least one piece of first audio sent by the plurality of clients, the method further includes:

receiving a login request sent by a second client of the plurality of clients, where the

login request includes verification information;

performing verification processing on the verification information, and creating a recording channel between the cloud desktop server and the second client after the verification of the verification information is successful, where the recording channel is a socket channel.

[0013] In a third aspect, an embodiment of the present application provides an audio processing method, applied to a client, where the client is provided with a microphone thereon, and the method includes:

performing audio collection through the microphone to obtain first audio;

determining a channel status of a recording channel between the client and a cloud desktop server, where the channel status is a muted state or an unmuted state;

when the channel status is the unmuted state, sending the first audio to the cloud desktop server through the recording channel.

[0014] In a possible implementation, the method further includes:

displaying a first page, where the first page includes a status control of the recording channel;

in response to a mute operation on the status control, sending a mute request to the cloud desktop server; or, in response to an unmute operation on the status control, sending an unmute request to the cloud desktop server.

[0015] In a possible implementation, the method further includes:

when the channel status is the unmuted state, storing the first audio into a preset storage space, or discarding the first audio.

[0016] In a possible implementation, the method further includes:

receiving a mixing result sent by the cloud desktop server, where the mixing result is obtained by the cloud desktop server performing mixing on audio from a further client and a meeting server.

[0017] In a fourth aspect, an embodiment of the present application provides an audio processing apparatus, applied to a cloud desktop server, where the audio processing apparatus includes: a first receiving module, a second receiving module and a determining module, where,

the first receiving module is configured to receive, through recording channels between a plurality of clients and the cloud desktop server, at least one piece of first audio sent

by the plurality of clients;

the second receiving module is configured to receive second audio sent by a meeting server, where the second audio is obtained by the meeting server processing audio from at least one meeting terminal;

the determining module is configured to determine, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and the meeting terminal, and send the corresponding mixed audio to each client and the meeting server.

[0018] In a possible implementation, the determining module is specifically configured to:

perform mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client;

perform mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server, and send the second mixed audio to the meeting server.

[0019] In a possible implementation, the determining module is specifically configured to:

determine a first client identifier of the client and a client identifier corresponding to each piece of first audio;

determine target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio, where a client identifier corresponding to the target audio is different from the first client identifier;

perform mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

[0020] In a possible implementation, the determining module is specifically configured to:

acquire channel statuses of the recording channels between the cloud desktop server and the plurality of clients, where the channel statuses are a muted state or an unmuted state;

determine at least one target recording channel from the recording channels according to the channel statuses of the plurality of recording channels, where the at least one target recording channel is in the unmuted state;

receive the at least one piece of first audio through the at least one target recording channel.

[0021] In a possible implementation, the first receiving module is further configured to:

receive a mute request sent by a first client of the plurality of clients, and set the channel status of the recording channel between the first client and the cloud desktop server to the muted state according to the mute request; or,

receive an unmute request sent by the first client, and set the channel status of the recording channel between the first client and the cloud desktop server to the unmuted state according to the unmute request.

[0022] In a possible implementation, the audio processing apparatus further includes a third receiving module and a creating module;

the third receiving module is configured to receive a login request sent by a second client of the plurality of clients, where the login request includes verification information;

the creating module is configured to perform verification processing on the verification information, and create a recording channel between the cloud desktop server and the second client after the verification of the verification information is successful, where the recording channel is a socket channel.

[0023] In a fifth aspect, an embodiment of the present application provides an audio processing apparatus, applied to a client, and the audio processing apparatus includes: a collecting module, a determining module and a sending module, where,

the collecting module is configured to perform audio collection through a microphone to obtain first audio;

the determining module is configured to determine a channel status of a recording channel between the client and a cloud desktop server, where the channel status is a muted state or an unmuted state;

the sending module is configured to send the first audio to the cloud desktop server through the recording channel when the channel status is in the unmuted state.

[0024] In a possible implementation, the audio processing apparatus further includes a displaying module:

the displaying module is configured to display a first page, where the first page includes a status control of the recording channel;

the sending module is configured to send a mute request to the cloud desktop server

in response to a mute operation on the status control; or send an unmute request to the cloud desktop server in response to an unmute operation on the status control.

[0025] In a possible implementation, the audio processing apparatus further includes a storage module:

the storage module is configured to, when the channel status is in the unmuted state, store the first audio into a preset storage space, or discard the first audio.

[0026] In a possible implementation, the audio processing apparatus further includes a receiving module:

the receiving module is configured to receive a mixing result sent by the cloud desktop server, where the mixing result is obtained by the cloud desktop server performing mixing on audio from a further client and a meeting server.

[0027] In a sixth aspect, an embodiment of the present application provides a cloud desktop server, including: a memory and a processor;

the memory stores computer-executable instructions;

the processor executes the computer-executable instructions stored in the memory, so that the processor performs the audio processing method according to any one of the first aspect or the second aspect.

[0028] In a seventh aspect, an embodiment of the present application provides a client, including: a memory, a processor, and a microphone;

where the microphone is configured to collect first audio;

the memory stores computer-executable instructions;

the processor executes the computer-executable instructions stored in the memory, so that the processor executes the audio processing method according to any of the third aspect.

[0029] In an eighth aspect, an embodiment of the present application provides an audio processing system, including the cloud desktop server according to the sixth aspect, and the at least one client described according to the seventh aspect, where the client is configured to access the cloud desktop server.

[0030] In a ninth aspect, an embodiment of the present application provides a computer-readable storage medium, which stores computer-executable instructions, which, when executed by a processor, are configured to implement the audio processing method according

to any one of the first aspect or the second aspect.

[0031] In a tenth aspect, an embodiment of the present application provides a computer-readable storage medium, where the computer-readable storage medium stores computer-executable instructions, which, when executed by a processor, are configured to implement the audio processing method according to any of the third aspect.

[0032] In an eleventh aspect, an embodiment of the present application provides a computer program product, including a computer program, which, when executed by a processor, implements the audio processing method according to any one of the first aspect or the second aspect.

[0033] In a twelfth aspect, an embodiment of the present application provides a computer program product, including a computer program, which, when executed by a processor, implements the audio processing method according to any of the third aspect.

[0034] The embodiments of the present application provide an audio processing method, a cloud desktop server, a client, and a system. The client can collect audio through a microphone to obtain the first audio and determine the channel status of the recording channel between the client and the cloud desktop server. When the channel status is the unmuted state, the client can send the first audio to the cloud desktop server through the recording channel, so that the cloud desktop server receives the first audio. The cloud desktop server can receive the second audio sent by the meeting server. The cloud desktop server can determine the mixed audio corresponding to each client and the meeting terminal according to the at least one piece of first audio and the second audio, and send the corresponding mixed audio to each client and the meeting server. Since the cloud desktop server can receive at least one piece of audio sent by a plurality of clients through a plurality of recording channels, compared with receiving the audio sent by one of the clients through one recording channel, the usage experience of users is improved.

BRIEF DESCRIPTION OF DRAWINGS

[0035] The drawings described herein are used to provide further understanding of the present application and constitute a part of the present application. The illustrative embodiments

of the present application and their descriptions are intended to explain the present application and do not constitute improper limitations on the present application. In the drawings:

FIG. 1 is a schematic diagram of an application scenario provided by an illustrative embodiment of the present application;

FIG. 2 is a schematic flowchart of an audio processing method provided by an illustrative embodiment of the present application;

FIG. 3 is a schematic flowchart of another audio processing method provided by an illustrative embodiment of the present application;

FIG. 4 is a first schematic diagram of an audio processing system provided by an illustrative embodiment of the present application;

FIG. 5A is a first schematic diagram of a first page provided by an illustrative embodiment of the present application;

FIG. 5B is a second schematic diagram of a first page provided by an illustrative embodiment of the present application;

FIG. 6 is a second schematic diagram of an audio processing system provided by an illustrative embodiment of the present application;

FIG. 7 is a schematic structural diagram of an audio processing apparatus provided by an illustrative embodiment of the present application;

FIG. 8 is a schematic structural diagram of another audio processing apparatus provided by an illustrative embodiment of the present application;

FIG. 9 is a schematic structural diagram of a cloud desktop server provided by an illustrative embodiment of the present application; and

FIG. 10 is a schematic structural diagram of a client provided by an illustrative embodiment of the present application.

DESCRIPTION OF EMBODIMENTS

[0036] In order to make the objectives, technical solutions and advantages of the present application clearer, the technical solutions of the present application will be clearly and completely described below in conjunction with the specific embodiments of the present

application and the corresponding drawings. Obviously, the described embodiments are merely part of the embodiments of the present application, rather than all of the embodiments. Based on the embodiments in the present application, all other embodiments obtained by those of ordinary skill in the art without making creative work shall fall within the protection scope of the present application.

[0037] FIG. 1 is a schematic diagram of an application scenario provided by an illustrative embodiment of the present application. Please refer to FIG. 1, which includes a plurality of clients, a cloud desktop server, and a meeting server. For example, the plurality of clients may include client 1, client 2, ..., client *n*. The cloud desktop server and any client can communicate with each other. The cloud desktop server and the meeting server can communicate with each other.

[0038] For any client, there may be a recording channel between the client and the cloud desktop server. The client can send first audio to the cloud desktop server through a corresponding recording channel. For example, client 1 can send first audio 1 to the cloud desktop server through recording channel 1, client 2 can send first audio 2 to the cloud desktop server through recording channel 2, ..., client *n* can send first audio *n* to the cloud desktop server through recording channel *n*.

[0039] The meeting server can send second audio to the cloud desktop server. The second audio can be obtained through processing audio from at least one meeting terminal by the meeting server.

[0040] After receiving the at least one piece of first audio and the second audio, the cloud desktop server can determine, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and the meeting terminal, and send the corresponding mixed audio to each client and the meeting server. For example, the cloud desktop server can determine, according to the at least one piece of first audio and the second audio, mixed audio 1 corresponding to client 1, mixed audio 2 corresponding to client 2, ..., mixed audio *n* corresponding to client *n*, determine mixed audio *a* corresponding to the meeting server, and send the corresponding mixed audio to each client and the meeting server.

[0041] In the related art, a plurality of clients can be supported to log in to the cloud desktop server. However, since there is typically one recording channel between the cloud desktop

server and the clients, the audio of one of the clients can be obtained through the recording channel each time, and the audio of the plurality of clients cannot be obtained concurrently, resulting in a poor user experience.

[0042] In an embodiment of the present application, there may be a recording channel between each client and the cloud desktop server. For any client, the cloud desktop server can receive the first audio sent by the client through a recording channel corresponding to the client. Since the cloud desktop server can receive at least one piece of first audio sent by a plurality of clients through a plurality of recording channels, compared with receiving the audio sent by one of the clients through one recording channel, the usage experience of users is improved.

[0043] The technical solution illustrated in the present application is described in detail below through specific embodiments. It should be noted that the following embodiments may exist independently or in combination with each other, and the same or similar contents will not be described repeatedly in different embodiments.

[0044] FIG. 2 is a schematic flowchart of an audio processing method provided by an illustrative embodiment of the present application. Please refer to FIG. 2, the method may include steps as follows.

[0045] S201, a client performs audio collection through a microphone to obtain first audio.

[0046] The client may be a lightweight electronic device with a display screen and a microphone. For example, the client may be a mobile phone or a television.

[0047] A user can input voice into the client, and the client can perform audio collection through the microphone to obtain the first audio.

[0048] For example, the user may input a voice “The weather is nice today” into the client, and the client can collect the audio through the microphone to obtain the first audio, which may include the voice data of “The weather is nice today”.

[0049] S202, the client determines a channel status of a recording channel between the client and a cloud desktop server.

[0050] The recording channel may be a channel for transmitting audio between the client and the cloud desktop server. There may be a recording channel between each client and the cloud desktop server.

[0051] The channel status of the recording channel may include a muted state or an unmuted

state.

[0052] Optionally, the client can determine the channel status of the recording channel between the client and the cloud desktop server. When the channel status of the recording channel is the muted state, the client can store the first audio to a preset storage space, or discard the first audio.

[0053] For example, it is assumed that there are two clients, client 1 and client 2, respectively. If a recording channel between client 1 and the cloud desktop server is recording channel 1, and a recording channel between client 2 and the cloud desktop server is recording channel 2, client 1 can determine that a channel status of the recording channel 1 is the muted state, and client 2 can determine that a channel status of the recording channel 2 is the unmuted state. If client 1 obtains first audio 1 by collecting, since the recording channel 1 is in the muted state, client 1 can store first audio 1 in a preset storage space of client 1 or discard first audio 1.

[0054] S203, when the channel status is in the unmuted state, the client sends the first audio to the cloud desktop server through the recording channel.

[0055] Optionally, a plurality of clients can send at least one piece of first audio to the cloud desktop server through corresponding recording channels, so that the cloud desktop server receives the at least one piece of first audio.

[0056] For example, it is assumed that there are two clients, client 1 and client 2, respectively. If a recording channel between client 1 and the cloud desktop server is recording channel 1, a recording channel between client 2 and the cloud desktop server is recording channel 2, client 1 obtains first audio 1 by collecting, client 2 obtains first audio 2 by collecting, and a channel status of recording channel 1 and a channel status of recording channel 2 are both the unmuted state, client 1 can send first audio 1 to the cloud desktop server through recording channel 1, and client 2 can send first audio 2 to the cloud desktop server through recording channel 2.

[0057] S204, the cloud desktop server receives second audio sent by a meeting server.

[0058] The meeting server can be configured to process audio from meeting terminal(s).

[0059] The meeting terminal may be an electronic device. For example, the meeting terminal may be a computer.

[0060] The second audio may be audio obtained by processing audio from the at least one

meeting terminal by the meeting server.

[0061] In an optional embodiment, the meeting server can receive the audio sent by the at least one meeting terminal, and process the audio to obtain the second audio. The meeting server can send the second audio to the cloud desktop server, so that the cloud desktop server receives the second audio from the meeting server.

[0062] For example, user 1 can input the voice “The weather is nice today” into meeting terminal 1, so that meeting terminal 1 obtains audio 1 by collecting; user 2 can input the voice “Indeed, the sunshine is splendid” into meeting terminal 2, so that meeting terminal 2 obtains audio 2 by collecting. Meeting terminal 1 can send audio 1 to the meeting server, and meeting terminal 2 can send audio 2 to the meeting server. After receiving audio 1 and audio 2, the meeting server can process audio 1 and audio 2, to obtain the second audio. The second audio may include “The weather is nice today” and “Indeed, the sunshine is splendid”. The meeting server can send the second audio to the cloud desktop server, so that the cloud desktop server receives the second audio.

[0063] S205, the cloud desktop server determines mixed audio corresponding to each client and the meeting terminal according to the at least one piece of first audio and the second audio, and sends the corresponding mixed audio to each client and the meeting server.

[0064] The mixed audio refers to audio obtained by processing the at least one piece of first audio and the second audio. The mixed audio may include the at least one piece of first audio and the second audio. For example, if there are first audio 1 and first audio 2, and the second audio includes audio 1 and audio 2, mixed audio 1 may include first audio 2, audio 1, and audio 2.

[0065] In an optional embodiment, it is possible to determine the mixed audio corresponding to each client and the meeting terminal and send the corresponding mixed audio to each client and the meeting server, in the following manner: performing mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and sending the corresponding first mixed audio to each client; performing mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server, and sending the second mixed audio to the meeting server.

[0066] For example, it is assumed that there are two clients, client 1 and client 2, respectively. If the cloud desktop server can receive first audio 1 sent by client 1 and first audio 2 sent by client 2, and the cloud desktop server can receive the second audio sent by the meeting server that includes audio 1 and audio 2, the cloud desktop server can perform mixing processing on the two pieces of first audio and the second audio, determine that first mixed audio 1 corresponding to client 1 includes first audio 2, audio 1 and audio 2, and send first mixed audio 1 to client 1; determine that first mixed audio 2 corresponding to client 2 includes first audio 1, audio 1 and audio 2, and send first mixed audio 2 to client 2. The cloud desktop server can perform mixing processing on the two pieces of first audio, determine that the second mixed audio corresponding to the meeting server includes first audio 1 and first audio 2, and send the second mixed audio to the meeting server.

[0067] In the embodiments of the present application, the client can collect audio through a microphone to obtain the first audio, and determine the channel status of the recording channel between the client and the cloud desktop server. When the channel status is the unmuted state, the client can send the first audio to the cloud desktop server through the recording channel, so that the cloud desktop server receives the first audio. The cloud desktop server can receive the second audio sent by the meeting server. The cloud desktop server can determine the mixed audio corresponding to each client and the meeting terminal according to the at least one piece of first audio and the second audio, and send the corresponding mixed audio to each client and the meeting server. Since the cloud desktop server can receive the at least one piece of audio sent by a plurality of clients through a plurality of recording channels, compared with receiving the audio sent by one of the clients through one recording channel, the usage experience of users is improved.

[0068] The aforementioned audio processing method is further described in detail below based on the embodiment illustrated in FIG. 2 and in combination with FIG. 3.

[0069] FIG. 3 is a schematic flowchart of another audio processing method provided by an illustrative embodiment of the present application. Please refer to FIG. 3, the method can include steps as follows.

[0070] S301, the client sends a login request to the cloud desktop server.

[0071] For any client, the client can send the login request to the cloud desktop server. The

login request may include verification information, which may include a user account and a password.

[0072] For example, if there are two clients, client 1 and client 2, respectively, client 1 can send login request 1 to the cloud desktop server, where login request 1 may include verification information 1, i.e., user account 1 and password 1; and client 2 can send login request 2 to the cloud desktop server, and login request 2 may include verification information 2, i.e., user account 2 and password 2.

[0073] S302, the cloud desktop server performs verification processing on the verification information, and creates the recording channel between the cloud desktop server and the client after the verification of the verification information is successful.

[0074] Optionally, the recording channel may be a socket channel.

[0075] In an optional embodiment, the cloud desktop server can determine a user account and a corresponding preset password. If the verification information includes a user account and a password, the cloud desktop server can perform verification processing on the user account and the password, that is, the cloud desktop server can determine whether the preset password corresponding to the user account is consistent with the password in the verification information. If the preset password corresponding to the user account is consistent with the password in the verification information, the verification is successful; if the preset password corresponding to the user account is inconsistent with the password in the verification information, the verification fails.

[0076] For example, it is assumed that login request 1 sent by client 1 to the cloud desktop server includes verification information 1, and verification information 1 includes user account 1 and password 1. If the cloud desktop server determines that a preset password corresponding to user account 1 is password 1, the verification is successful since the preset password corresponding to user account 1 is consistent with password 1 in verification information 1; if the cloud desktop server determines that the preset password corresponding to user account 1 is password 2, the verification fails since the preset password corresponding to user account 1 is inconsistent with password 1 in verification information 1.

[0077] Optionally, after the verification of the verification information is successful, the cloud desktop server can create the recording channel between the cloud desktop server and the

client. For example, it is assumed that login request 1 sent by client 1 to the cloud desktop server includes verification information 1. If the verification of verification information 1 by the cloud desktop server is successful, the cloud desktop server can create recording channel 1 with the client, and recording channel 1 may be a socket channel.

[0078] S303, the client performs audio collection through a microphone to obtain first audio.

[0079] After the client successfully logs into the cloud desktop server, a user can input a voice into the client, and the client can perform audio collection through a microphone to obtain the first audio.

[0080] For example, user 1 may input a voice "The weather is nice today" into client 1, and client 1 can collect the audio through a microphone to obtain first audio 1, which may include the voice data of "The weather is nice today".

[0081] S304, the client determines a channel status of the recording channel between the client and the cloud desktop server.

[0082] It should be noted that a reference for the execution process of step S304 can be made to the execution process of step S202, which will not be repeated here.

[0083] S305, when the channel status is an unmuted state, the client sends the first audio to the cloud desktop server through the recording channel.

[0084] It should be noted that a reference for the execution process of step S305 can be made to the execution process of step S203, which will not be repeated here.

[0085] S306, the meeting server sends second audio to the cloud desktop server.

[0086] Optionally, the meeting server can receive audio sent by at least one meeting terminal, and process the audio to obtain the second audio. The meeting server can send the second audio to the cloud desktop server, so that the cloud desktop server receives the second audio from the meeting server.

[0087] For example, meeting terminal 1 can send audio 1 to the meeting server, and meeting terminal 2 can send audio 2 to the meeting server. After receiving audio 1 and audio 2, the meeting server can perform processing on audio 1 and audio 2, to obtain the second audio, which may include audio 1 and audio 2. The meeting server can send the second audio to the cloud desktop server, so that the cloud desktop server receives the second audio.

[0088] S307, the cloud desktop server performs mixing processing on the at least one piece

of first audio and the second audio to obtain first mixed audio corresponding to each client.

[0089] For any client, mixing processing can be performed on the at least one piece of first audio and the second audio in the following manner to obtain the first mixed audio corresponding to each client: determining a first client identifier of the client and a client identifier corresponding to each piece of first audio; determining target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio; performing mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

[0090] The plurality of clients may include a first client, and the cloud desktop server can determine a first client identifier of the first client. For example, it is assumed that there are three clients, client 1, client 2 and client 3, respectively. If the first client is client 1, the cloud desktop server can determine that the first client identifier is client 1.

[0091] For any first audio, the first audio may have a corresponding client identifier. The cloud desktop server can determine the client identifier corresponding to the first audio. For example, if client 1 sends first audio 1 to the cloud desktop server through recording channel 1, the cloud desktop server can determine that the client identifier corresponding to first audio 1 is client 1.

[0092] Optionally, the cloud desktop server can receive the at least one piece of first audio sent by the at least one client, and determine the target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio. The client identifier corresponding to the target audio may be different from the first client identifier.

[0093] For example, it is assumed that the first client identifier is client 1. If the cloud desktop server can receive first audio 1 sent by client 1, first audio 2 sent by client 2, and first audio 3 sent by client 3, the cloud desktop server can determine that the client identifier corresponding to first audio 1 is client 1, the client identifier corresponding to first audio 2 is client 2, and the client identifier corresponding to first audio 3 is client 3. Since the first client identifier is client 1, the cloud desktop server can determine that target audio 1 includes first audio 2 and first audio 3 among first audio 1, first audio 2, and first audio 3. The client identifier corresponding to target audio 1 includes client 2 and client 3, which are different from the first

client identifier, i.e., client 1.

[0094] Optionally, after determining the target audio, the cloud desktop server can perform mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

[0095] For example, it is assumed that the cloud desktop server receives first audio 1 sent by client 1, first audio 2 sent by client 2, and first audio 3 sent by client 3, and the cloud desktop server receives the second audio sent by the meeting server that includes audio 1 and audio 2. If the first client identifier is client 1, the cloud desktop server determines that target audio 1 includes first audio 2 and first audio 3, the cloud desktop server can perform mixing processing on the target audio 1 and the second audio to obtain first mixed audio 1, that is, the cloud desktop server can determine that first mixed audio 1 includes first audio 2, first audio 3, audio 1 and audio 2; if the first client identifier is client 2, the cloud desktop server determines that target audio 2 includes first audio 1 and first audio 3. The cloud desktop server can perform mixing processing on target audio 2 and the second audio to obtain first mixed audio 2, that is, the cloud desktop server can determine that first mixed audio 2 includes first audio 1, first audio 3, audio 1, and audio 2.

[0096] S308, the cloud desktop server sends the corresponding first mixed audio to each client.

[0097] For example, if the cloud desktop server determines that first mixed audio 1 includes first audio 2, first audio 3, audio 1, and audio 2, the cloud desktop server can send first mixed audio 1 to client 1; if the cloud desktop server determines that first mixed audio 2 includes first audio 1, first audio 3, audio 1, and audio 2, the cloud desktop server can send first mixed audio 2 to client 2.

[0098] S309, the cloud desktop server performs mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server.

[0099] For example, if the cloud desktop server receives first audio 1 sent by client 1, first audio 2 sent by client 2, and first audio 3 sent by client 3, the cloud desktop server can perform mixing processing on the three pieces of first audio to obtain the second mixed audio, that is, the cloud desktop server can determine that the second mixed audio includes first audio 1, first audio 2, and first audio 3.

[0100] S310, the cloud desktop server sends the second mixed audio to the meeting server.

[0101] For example, if the cloud desktop server determines that the second mixed audio includes first audio 1, first audio 2, and first audio 3, the cloud desktop server sends the second mixed audio to the meeting server.

[0102] Optionally, after receiving the second mixed audio, the meeting server can perform mixing processing on the second mixed audio and the second audio to determine the mixed audio corresponding to each meeting terminal.

[0103] For example, if there are two meeting terminals, meeting terminal 1 and meeting terminal 2, respectively, and the meeting server can receive audio 1 sent by meeting terminal 1 and audio 2 sent by meeting terminal 2, the second audio includes audio 1 and audio 2. If the second mixed audio includes first audio 1, first audio 2, and first audio 3, the meeting server can perform mixing processing on the second mixed audio and the second audio to determine the mixed audio corresponding to each meeting terminal, that is, the meeting server can determine that the mixed audio corresponding to meeting terminal 1 includes first audio 1, first audio 2, first audio 3, and audio 2; that the mixed audio corresponding to meeting terminal 2 includes first audio 1, first audio 2, first audio 3, and audio 1.

[0104] In the embodiments of the present application, the client can send the login request to the cloud desktop server, and the cloud desktop server can perform verification processing on the verification information and create the recording channel between the cloud desktop server and the client after the verification of the verification information is successful. The client can perform audio collection through a microphone to obtain the first audio, and determine the channel status of the recording channel between the client and the cloud desktop server. When the channel status is the unmuted state, the client sends the first audio to the cloud desktop server through the recording channel. The meeting server can send the second audio to the cloud desktop server. The cloud desktop server can perform mixing processing on the at least one piece of first audio and the second audio to obtain the first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client; the cloud desktop server can perform mixing processing on the at least one piece of first audio to obtain the second mixed audio corresponding to the meeting server, and send the second mixed audio to the meeting server. Since the cloud desktop server can receive at least one piece of audio sent by a

plurality of clients through a plurality of recording channels, compared with receiving the audio sent by one of the clients through one recording channel, the usage experience of users is improved.

[0105] An audio processing system is provided below in conjunction with FIG. 4 based on any one of the aforementioned embodiments.

[0106] FIG. 4 is a first schematic diagram of an audio processing system provided by an illustrative embodiment of the present application. Please refer to FIG. 4, which includes a cloud desktop server and at least one client, e.g., the at least one client may be client 1, client 2, ..., and client *n*.

[0107] For any client, the client can send a login request to the cloud desktop server, and the login request may include verification information. After receiving the login request, the cloud desktop server can perform verification processing on the verification information in the login request. If the verification is successful, the cloud desktop server can create a recording channel with the client, and the recording channel may be a socket channel. For example, client 1 can send login request 1 to the cloud desktop server, and login request 1 may include verification information 1. After receiving login request 1, the cloud desktop server can perform verification processing on verification information 1. If the verification is successful, the cloud desktop server can create recording channel 1 with client 1.

[0108] For any client, after the client successfully logs in to the cloud desktop server, it can participate in a meeting through the cloud desktop server.

[0109] Optionally, during the meeting, the client can display a first page, which includes a status control of the recording channel.

[0110] The first page will be described below with reference to FIG. 5A and FIG. 5B.

[0111] FIG. 5A is a first schematic diagram of a first page provided by an illustrative embodiment of the present application. Please refer to FIG. 5A, the client can display the first page, which may include the status control of the recording channel. A user may perform a mute operation on the status control on the client, as shown in FIG. 5A. The client can send a mute request to the cloud desktop server in response to the mute operation on the status control. The cloud desktop server can receive the mute request sent by the client, and set, according to the mute request, the channel status of the recording channel between the client and the cloud

desktop server to a muted state.

[0112] FIG. 5B is a second schematic diagram of a first page provided by an illustrative embodiment of the present application. Please refer to FIG. 5B, the client can display the first page, which may include the status control of the recording channel. A user may perform an unmute operation on the status control on the client, as shown in FIG. 5B. The client can send an unmute request to the cloud desktop server in response to the unmute operation on the status control. The cloud desktop server can receive the unmute request from the client, and set, according to the unmute request, the channel status of the recording channel between the client and the cloud desktop server to an unmuted state.

[0113] For example, client 1 can send unmute request 1 to the cloud desktop server in response to an unmute operation by user 1 on the status control. After receiving unmute request 1, the cloud desktop server can set, according to unmute request 1, a channel status of recording channel 1 between client 1 and the cloud desktop server to the unmuted state; client 2 can send mute request 2 to the cloud desktop server in response to a mute operation by user 2 on the status control. After receiving mute request 2, the cloud desktop server can set, according to mute request 2, a channel status of recording channel 2 between client 2 and the cloud desktop server to the muted state; ...; client *n* can send unmute request *n* to the cloud desktop server in response to an unmute operation by user *n* on the status control. After receiving unmute request *n*, the cloud desktop server can set, according to unmute request *n*, a channel status of recording channel *n* between client *n* and the cloud desktop server to the unmuted state.

[0114] For any client, a microphone may be provided on the client. During a meeting, the client can collect audio through the microphone to obtain the first audio.

[0115] It should be noted that during the audio collection process, the microphone of the client can remain active all the times. The cloud desktop server is capable of setting the channel status of the recording channel, to enable the switching of the recording channel between the muted state and the unmuted state, thereby meeting user needs. Since the microphone of the client is active all the time, when the recording channel switches from the muted state to the unmuted state, there is no need to turn on the microphone, which reduces the client's recording delay.

[0116] Since the cloud desktop server can set, according to the mute request sent by any

client, the channel status of the recording channel between the client and the cloud desktop server to the muted state, or can set, according to the unmute request sent by any client, the channel status of the recording channel between the client and the cloud desktop server to the unmuted state, the channel status of the recording channels between a plurality of clients and the cloud desktop server can include the muted state and the unmuted state.

[0117] Optionally, the cloud desktop server can acquire the channel statuses of the plurality of recording channels with the plurality of clients, where the channel statuses are a muted state or an unmuted state; determine at least one target recording channel from the plurality of recording channels according to the channel statuses of the plurality of recording channels, where the target recording channel is in the unmuted state; and receive the at least one piece of first audio through the at least one target recording channel. When the channel status of the recording channel is the muted state, the client can store the first audio to a preset storage space, or discard the first audio.

[0118] For example, if the channel status of recording channel 1 between client 1 and the cloud desktop server is the unmuted state, and the channel status of recording channel 2 between client 2 and the cloud desktop server is the muted state, ..., and the channel status of recording channel *n* between client *n* and the cloud desktop server is the unmuted state, since the channel statuses of recording channel 1, ..., and recording channel *n* is the unmuted state, the cloud desktop server can determine that the target recording channels include recording channel 1, ..., and recording channel *n*. The cloud desktop server can receive first audio 1 sent by client 1 through recording channel 1, ..., can receive first audio *n* sent by client *n* through recording channel *n*. Since the channel status of recording channel 2 is the muted state, client 2 can store first audio 2 into the preset storage space, or discard first audio 2, without sending first audio 2 to the cloud desktop server.

[0119] As the cloud desktop server can determine the channel status of each recording channel before obtaining the first audio, the at least one piece of first audio can be obtained through a recording channel in the unmuted state, thereby reducing power consumption.

[0120] Optionally, the cloud desktop server can also receive the second audio sent by the meeting server. The second audio may be audio obtained through processing audio from at least one meeting terminal by the meeting server.

[0121] After receiving the at least one piece of first audio and the second audio, the cloud desktop server can perform mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client.

[0122] Specifically, the cloud desktop server can determine a first client identifier of the client and a client identifier corresponding to each piece of first audio, and determine target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio. The cloud desktop server can perform mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client, and send the corresponding first mixed audio to the client, to enable the client to receive a mixing result sent by the cloud desktop server.

[0123] For example, if the cloud desktop server can receive first audio 1 sent by client 1, ..., and first audio *n* sent by client *n*, and the cloud desktop server receives the second audio sent by the meeting server that includes audio 1 and audio 2, the cloud desktop server can determine that the client identifier corresponding to first audio 1 is client 1, ..., and the client identifier corresponding to first audio *n* is client *n*. If the first client identifier is client 1, the cloud desktop server can determine target audio 1 in first audio 1, ..., and first audio *n*, and perform mixing processing on target audio 1 and the second audio to obtain first mixed audio 1, and send first mixed audio 1 to client 1; if the first client identifier is client 2, the cloud desktop server determines target audio 2, and performs mixing processing on target audio 2 and the second audio to obtain first mixed audio 2, and sends first mixed audio 2 to client 2; ...; if the first client identifier is client *n*, the cloud desktop server determines target audio *n*, and performs mixing processing on target audio *n* and the second audio to obtain first mixed audio *n*, and sends the first mixed audio *n* to client *n*.

[0124] Optionally, the cloud desktop server can further perform mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server and send the second mixed audio to the meeting server.

[0125] For example, if the cloud desktop server receives first audio 1 sent by client 1, ..., and first audio *n* sent by client *n*, the cloud desktop server can perform mixing processing on first audio 1, ..., and first audio *n* to obtain the second mixed audio, and send the second mixed

audio to the meeting server.

[0126] In the embodiments of the present application, the client can send the login request to the cloud desktop server, and the cloud desktop server can verify the verification information and create the recording channel between the cloud desktop server and the client after the verification of the verification information is successful. The client can collect audio through a microphone to obtain the first audio, and determine the channel status of the recording channel between the client and the cloud desktop server. When the channel status is the unmuted state, the cloud desktop server can receive the first audio sent by the client through the recording channel. The meeting server can send the second audio to the cloud desktop server. The cloud desktop server can perform mixing processing on the at least one piece of first audio and the second audio to obtain the first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client; the cloud desktop server can perform mixing processing on the at least one piece of first audio to obtain the second mixed audio corresponding to the meeting server, and send the second mixed audio to the meeting server. Since the cloud desktop server can receive at least one piece of audio sent by a plurality of clients through a plurality of recording channels, compared with receiving the audio sent by one of the clients through one recording channel, the usage experience of users is improved.

[0127] The audio processing system is further described in detail below in combination with FIG. 6 based on any of the aforementioned embodiments.

[0128] FIG. 6 is a second schematic diagram of an audio processing system provided by an illustrative embodiment of the present application. Please refer to FIG. 6, which includes at least one client, a cloud desktop server, a meeting server and at least one meeting terminal. For example, the at least one client may be client 1, client 2, ..., and client *n*; the at least one meeting terminal may be meeting terminal 1, meeting terminal 2, ..., and meeting terminal *m*.

[0129] For any client, the client can send a login request to the cloud desktop server, where the login request may include verification information. After receiving the login request, the cloud desktop server can perform verification processing on the verification information in the login request. If the verification is successful, the cloud desktop server can create a recording channel between the client and the cloud desktop server, where the recording channel may be a socket channel. For example, the cloud desktop server can establish recording channel 1

corresponding to client 1, recording channel 2 corresponding to client 2, ..., and recording channel *n* corresponding to client *n*.

[0130] For any client, after the client successfully logs in to the cloud desktop server, it can participate in a meeting through the cloud desktop server.

[0131] Optionally, during the meeting, the client can display a first page, which includes a status control of the recording channel. The client can send a mute request to the cloud desktop server in response to a mute operation by a user on the status control; or send an unmute request to the cloud desktop server in response to an unmute operation by a user on the status control.

[0132] The cloud desktop server can receive the mute request sent by the client, and set, according to the mute request, the channel status of the recording channel between the client and the cloud desktop server to the muted state; or, the cloud desktop server can receive the unmute request sent by the client, and set, according to the unmute request, the channel status of the recording channel between the client and the cloud desktop server to the unmuted state.

[0133] For example, the cloud desktop server can set recording channel 1 to the unmuted state, recording channel 2 to the muted state, and recording channel *n* to the unmuted state.

[0134] For any client, a microphone may be provided on the client. During a meeting, the client can collect audio through the microphone to obtain the first audio.

[0135] The cloud desktop server can acquire the channel statuses of the plurality of recording channels with the plurality of clients, and determine at least one target recording channel from the plurality of recording channels according to the channel statuses of the plurality of recording channels, and then receive, through the at least one target recording channel, at least one piece of first audio sent by the client.

[0136] For example, the cloud desktop server can determine that a channel status of recording channel 1 is the unmuted state, a channel status of recording channel 2 is the muted state, and a channel status of recording channel *n* is the unmuted state. Then, the cloud desktop server can receive, through recording channel 1, first audio 1 sent by client 1, ..., and receive, through recording channel *n*, first audio *n* sent by client *n*. Since the channel status of recording channel 2 is the muted state, client 2 can store first audio 2 into the preset storage space, or discard first audio 2, without sending first audio 2 to the cloud desktop server.

[0137] Optionally, the meeting server can receive at least one piece of audio sent by at least

one meeting terminal, and perform processing on the at least one piece of audio to obtain the second audio. The meeting server can send the second audio to the cloud desktop server, so that the cloud desktop server receives the second audio.

[0138] For example, the meeting server can receive audio 1 that meeting terminal 1 can send to the meeting server, audio 2 that meeting terminal 2 can send to the meeting server, ..., and audio *m* that meeting terminal *m* can send to the meeting server. The meeting server can perform processing on the *m* pieces of audio to obtain the second audio, which may include audio 1, audio 2, ..., and audio *m*. The meeting server can send the second audio to the cloud desktop server, so that the cloud desktop server receives the second audio.

[0139] After receiving the at least one piece of first audio and the second audio, the cloud desktop server can perform mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client.

[0140] For example, the cloud desktop server can perform mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio 1 corresponding to client 1, first mixed audio 2 corresponding to client 2, ..., and first mixed audio *n* corresponding to client *n*, and send the corresponding first mixed audio to each client.

[0141] Optionally, the cloud desktop server can perform mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server and send the second mixed audio to the meeting server.

[0142] Optionally, after receiving the second mixed audio, the meeting server can perform mixing processing on the second mixed audio and the second audio, determine mixed audio corresponding to each meeting terminal, and send the corresponding mixed audio to each meeting terminal.

[0143] For example, after receiving the second mixed audio, the meeting server can perform mixing processing on the second mixed audio and the second audio, to obtain mixed audio 1 corresponding to meeting terminal 1, mixed audio 2 corresponding to meeting terminal 2, ..., and mixed audio *m* corresponding to meeting terminal *m*, and send the corresponding mixed audio to each meeting terminal.

[0144] In the embodiments of the present application, the client can send the login request

to the cloud desktop server, and the cloud desktop server can perform verification processing on the verification information and create the recording channel between the cloud desktop server and the client after the verification of the verification information is successful. The client can collect audio through a microphone to obtain the first audio, and determine the channel status of the recording channel between the client and the cloud desktop server. When the channel status is the unmuted state, the cloud desktop server can receive the first audio sent by the client through the recording channel. The meeting server can send the second audio to the cloud desktop server. The cloud desktop server can perform mixing processing on the at least one piece of first audio and the second audio to obtain the first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client; the cloud desktop server can perform mixing processing on the at least one piece of first audio to obtain the second mixed audio corresponding to the meeting server, and send the second mixed audio to the meeting server. Since the cloud desktop server can receive at least one piece of audio sent by a plurality of clients through a plurality of recording channels, compared with receiving the audio sent by one of the clients through one recording channel, the usage experience of users is improved.

[0145] FIG. 7 is a schematic structural diagram of an audio processing apparatus provided by an illustrative embodiment of the present application, which may be applied to a cloud desktop server. Please refer to FIG. 7, the audio processing apparatus 10 includes: a first receiving module 11, a second receiving module 12, and a determining module 13, where,

the first receiving module 11 is configured to receive, through recording channels between a plurality of clients and the cloud desktop server, at least one piece of first audio sent by the plurality of clients;

the second receiving module 12 is configured to receive second audio sent by a meeting server, where the second audio is obtained by the meeting server processing audio from at least one meeting terminal;

the determining module 13 is configured to determine, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and the meeting terminal, and send the corresponding mixed audio to each client and the meeting server.

[0146] The audio processing apparatus provided in the embodiment of the present

application can execute the technical solutions illustrated in the aforementioned method embodiments, and their implementation principle and beneficial effects are similar, which will not be repeated here.

[0147] In a possible implementation, the determining module 13 is specifically configured to:

perform mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client;

perform mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server, and send the second mixed audio to the meeting server.

[0148] In a possible implementation, the determining module 13 is specifically configured to:

determine a first client identifier of the client and a client identifier corresponding to each piece of first audio;

determine target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio, where a client identifier corresponding to the target audio is different from the first client identifier;

perform mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

[0149] In a possible implementation, the determining module 13 is specifically configured to:

acquire channel statuses of the plurality of recording channels between the cloud desktop server and the plurality of clients, where the channel statuses are a muted state or an unmuted state;

determine at least one target recording channel from the plurality of recording channels according to the channel statuses of the plurality of recording channels, where the target recording channel is the unmuted state;

receive the at least one piece of first audio through the at least one target recording channel.

[0150] In a possible implementation, the first receiving module 11 is further configured to:

receive a mute request sent by the client, and set the channel status of the recording channel between the client and the cloud desktop server to the muted state according to the mute request; or,

receive an unmute request sent by the client, and set the channel status of the recording channel between the client and the cloud desktop server to the unmuted state according to the unmute request.

[0151] In a possible implementation, the audio processing apparatus further includes a third receiving module 14 and a creating module 15.

the third receiving module 14 is configured to receive a login request sent by a client, where the login request includes verification information;

the creating module 15 is configured to perform verification processing on the verification information, and create a recording channel between the cloud desktop server and the client after the verification of the verification information is successful, where the recording channel is a socket channel.

[0152] The audio processing apparatus provided in the embodiment of the present application can execute the technical solutions illustrated in the aforementioned method embodiments, and their implementation principle and beneficial effects are similar, which will not be repeated here.

[0153] FIG. 8 is a schematic structural diagram of another audio processing apparatus provided by an illustrative embodiment of the present application, which may be applied to a client. Please refer to FIG. 8, the audio processing apparatus 20 includes: a collecting module 21, a determining module 22, and a sending module 23; where,

the collecting module 21 is configured to perform audio collection through a microphone to obtain first audio;

the determining module 22 is configured to determine a channel status of a recording channel between the client and a cloud desktop server, where the channel status is a muted state or an unmuted state;

the sending module 23 is configured to send first audio to the cloud desktop server through the recording channel when the channel status is the unmuted state.

[0154] The audio processing apparatus provided in the embodiment of the present application can execute the technical solutions illustrated in the aforementioned method embodiments, and their implementation principle and beneficial effects are similar, which will not be repeated here.

[0155] In a possible implementation, the audio processing apparatus further includes a displaying module 24, where,

the displaying module 24 is configured to display a first page, where the first page includes a status control of the recording channel;

the sending module 23 is configured to send a mute request to the cloud desktop server in response to a mute operation on the status control; or send an unmute request to the cloud desktop server in response to an unmute operation on the status control.

[0156] In a possible implementation, the audio processing apparatus further includes a storage module 25, and

the storage module 25 is configured to, when the channel status is the unmuted state, store the first audio into a preset storage space, or discard the first audio.

[0157] In a possible implementation, the audio processing apparatus further includes a receiving module 26, and

the receiving module 26 is configured to receive a mixing result sent by the cloud desktop server, where the mixing result is obtained by the cloud desktop server performing mixing on audio from a further client and a meeting server.

[0158] The audio processing apparatus provided in the embodiment of the present application can execute the technical solution illustrated in the aforementioned method embodiments, and their implementation principle and beneficial effects are similar, which will not be repeated here.

[0159] FIG. 9 is a schematic structural diagram of a cloud desktop server provided by an illustrative embodiment of the present application. Please refer to FIG. 9, the cloud desktop server 30 may include a processor 31 and a memory 32. Illustratively, the processor 31 and the memory 32 are interconnected to each other via a bus 33.

[0160] The memory 32 stores computer-executable instructions; and

the processor 31 executes the computer-executable instructions stored in the

memory 32, so that the processor 31 performs the audio processing method as illustrated in the aforementioned method embodiments.

[0161] FIG. 10 is a schematic structural diagram of a client provided by an illustrative embodiment of the present application. Please refer to FIG. 10, the client 40 may include a processor 41, a memory 42, and a microphone 43. Illustratively, the processor 41, the memory 42, and the microphone 43 are interconnected to each other via a bus 44.

[0162] The microphone 43 is configured to collect first audio;

the memory 42 stores computer-executable instructions; and

the processor 41 executes the computer-executable instructions stored in the memory 42, so that the processor 41 performs the audio processing method as illustrated in the aforementioned method embodiments.

[0163] Accordingly, an embodiment of the present application provides a computer-readable storage medium storing computer-executable instructions that, when executed by a processor, are used to implement the audio processing method described in the aforementioned method embodiments.

[0164] Accordingly, an embodiment of the present application may further provide a computer program product, including a computer program, which, when executed by a processor, can implement the audio processing method illustrated in the aforementioned method embodiments.

[0165] Those skilled in the art should understand that the embodiments of the present invention can be provided as methods, systems or computer program products. Accordingly, the present invention can take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment combining software and hardware aspects. Moreover, the present invention can take the form of a computer program product implemented on one or more computer-usable storage media (including but not limited to disk storage, CD-ROM, optical storage, etc.) containing therein computer-usable program codes.

[0166] The present invention is described with reference to flowcharts and/or block diagrams of methods, devices (systems), and computer program products according to the embodiments of the present invention. It should be understood that each procedure and/or block in the flowcharts and/or block diagrams, and combinations of procedures and/or blocks in the

flowcharts and/or block diagrams can be implemented by computer program instructions. These computer program instructions can be provided to a processor of a general-purpose computer, a special-purpose computer, an embedded processor, or other programmable data processing devices to produce a machine, such that the instructions executed by the processor of the computer or other programmable data processing devices produce means for implementing the functions specified in one or more procedures of the flowcharts and/or one or more blocks of the block diagrams.

[0167] These computer program instructions may also be stored in a computer-readable memory that can guide a computer or other programmable data processing device to operate in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including the instruction means. The instruction means implements the functions specified in one or more procedures of the flowcharts and/or one or more blocks of the block diagrams.

[0168] These computer program instructions can also be loaded onto a computer or other programmable data processing device, so that a series of operating steps are performed on the computer or other programmable device to produce computer-implemented processing, thereby the instructions executed on the computer or other programmable device provide steps for implementing the functions specified in one or more procedures of the flowcharts and/or one or more blocks of the block diagrams.

[0169] In a typical configuration, a computing device includes one or more processors (CPU (Central Processing Unit)), input/output interfaces, network interfaces, and memory.

[0170] Memory may include forms such as non-permanent storage (e.g., random access memory (RAM)) and/or non-volatile memory (e.g., read-only memory (ROM) or flash memory (flash RAM)) in computer-readable media. Memory is an example of a computer-readable medium.

[0171] Computer-readable media include both permanent and non-permanent, removable and non-removable media that can implement information storage using any method or technology. The information may be computer-readable instructions, data structures, program modules, or other data. Examples of computer storage media include, but are not limited to, phase change memory (PRAM), static random access memory (SRAM), dynamic random

access memory (DRAM), other types of random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disk read-only memory (CD-ROM), digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic disk storage, or other magnetic storage devices, or any other non-transmission media that can be used to store information that can be accessed by a computing device. According to the definition provided herein, computer-readable media does not include temporary computer-readable media (transitory media), such as modulated data signals and carrier waves.

[0172] It should also be noted that the terms “comprise”, “include,” or any other variants thereof, are intended to cover non-exclusive inclusions, such that a process, method, product, or device that includes a series of elements includes not only those elements, but also other elements not explicitly listed or inherent to such process, method, product, or device. In the absence of more limitations, the elements defined by the phrase “comprising a...” do not exclude the presence of additional identical element(s) in the process, method, product, or device that includes the said element.

[0173] The above description is merely embodiments of the present application and is not intended to limit the present application. For those skilled in the art, various modifications and variations may be made to the present application. Any modifications, equivalent substitutions, improvements, and the like, made within the spirit and principles of the present application shall be included in the scope of the claims of the present application.

CLAIMS

1. An audio processing method, applied to a cloud desktop server, wherein a plurality of clients access the cloud desktop server, and a recording channel is provided between each client and the cloud desktop server, and the method comprises:

receiving, through recording channels between the plurality of clients and the cloud desktop server, at least one piece of first audio sent by the plurality of clients;

receiving second audio sent by a meeting server, wherein the second audio is obtained by the meeting server processing audio from at least one meeting terminal;

determining, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and the meeting terminal, and sending the corresponding mixed audio to each client and the meeting server.

2. An an audio processing method, applied to a cloud desktop server, wherein the cloud desktop server is configured to connect to a plurality of clients, and the method comprises:

receiving, through recording channels between the plurality of clients and the cloud desktop server, at least one piece of first audio sent by the plurality of clients;

receiving second audio sent by a meeting server;

determining, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and a meeting terminal, and sending the corresponding mixed audio to each client and the meeting server.

3. The method according to claim 1 or 2, wherein the determining, according to the at least one piece of first audio and the second audio, the mixed audio corresponding to each client and the meeting terminal, and sending the corresponding mixed audio to each client and the meeting server comprises:

performing mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and sending the corresponding first mixed audio to each client;

performing mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server, and sending the second mixed audio to the meeting server.

4. The method according to claim 3, wherein for any client, the performing mixing processing on the at least one piece of first audio and the second audio to obtain the first mixed audio corresponding to the client comprises:

determining a first client identifier of the client and a client identifier corresponding to each piece of first audio;

determining target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio, wherein a client identifier corresponding to the target audio is different from the first client identifier;

performing mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

5. The method according to any one of claims 1 to 4, wherein the receiving, through the recording channels between the plurality of clients and the cloud desktop server, the at least one piece of first audio sent by the plurality of clients comprises:

acquiring channel statuses of the recording channels between the cloud desktop server and the plurality of clients, wherein the channel statuses are a muted state or an unmuted state;

determining at least one target recording channel from the recording channels according to the channel statuses of the recording channels, wherein the at least one target recording channel is in the unmuted state;

receiving the at least one piece of first audio through the at least one target recording channel.

6. The method according to claim 5, wherein the method further comprises:

receiving a mute request sent by a first client of the plurality of clients, and setting the channel status of the recording channel between the first client and the cloud desktop server to the muted state according to the mute request; or,

receiving an unmute request sent by the first client, and setting the channel status of the recording channel between the first client and the cloud desktop server to the unmuted state according to the unmute request.

7. The method according to any one of claims 1 to 6, before receiving, through the recording channels between the plurality of clients and the cloud desktop server, the at least one piece of first audio sent by the plurality of clients, further comprising:

receiving a login request sent by a second client of the plurality of clients, wherein the login request comprises verification information;

performing verification processing on the verification information, and creating a recording channel between the cloud desktop server and the second client after the verification of the verification information is successful, wherein the recording channel is a socket channel.

8. An audio processing method, applied to a client, wherein the client is provided with a microphone thereon, and the method comprises:

performing audio collection through the microphone to obtain first audio;

determining a channel status of a recording channel between the client and a cloud desktop server, wherein the channel status is a muted state or an unmuted state;

when the channel status is the unmuted state, sending the first audio to the cloud desktop server through the recording channel.

9. The method according to claim 8, further comprising:

displaying a first page, wherein the first page comprises a status control of the recording channel;

in response to a mute operation on the status control, sending a mute request to the cloud desktop server; or, in response to an unmute operation on the status control, sending an unmute request to the cloud desktop server.

10. The method according to claim 8 or 9, further comprising:

when the channel status is the unmuted state, storing the first audio into a preset storage space, or discarding the first audio.

11. The method according to any one of claims 8 to 10, further comprising:

receiving a mixing result sent by the cloud desktop server, wherein the mixing result is obtained by the cloud desktop server performing mixing on audio from a further client and a meeting server.

12. A cloud desktop server, comprising: a memory and a processor;

wherein the memory stores computer-executable instructions; and

the processor executes the computer-executable instructions stored in the memory, so that the processor executes the audio processing method according to any one of claims 1 to 7.

13. A client, comprising: a memory, a processor, and a microphone;

wherein the microphone is configured to collect first audio;

the memory stores computer-executable instructions;

the processor executes the computer-executable instructions stored in the memory, so that the processor executes the audio processing method according to any one of claims 8 to 11.

14. An audio processing system, comprising the cloud desktop server according to claim 12 and the at least one client according to claim 13, wherein the client is configured to access the cloud desktop server.

15. An audio processing apparatus, applied to a cloud desktop server, comprising: a first receiving module, a second receiving module and a determining module; wherein,

the first receiving module is configured to receive, through recording channels between a plurality of clients and the cloud desktop server, at least one piece of first audio sent by the plurality of clients;

the second receiving module is configured to receive second audio sent by a meeting server, wherein the second audio is obtained by the meeting server processing audio from at least one meeting terminal;

the determining module is configured to determine, according to the at least one piece of first audio and the second audio, mixed audio corresponding to each client and the meeting terminal, and send the corresponding mixed audio to each client and the meeting server.

16. The audio processing apparatus according to claim 15, wherein the determining module is specifically configured to:

perform mixing processing on the at least one piece of first audio and the second audio to obtain first mixed audio corresponding to each client, and send the corresponding first mixed audio to each client;

perform mixing processing on the at least one piece of first audio to obtain second mixed audio corresponding to the meeting server, and send the second mixed audio to the meeting server;

wherein for any client, when performing mixing processing on the at least one piece of first audio and the second audio to obtain the first mixed audio corresponding to the client, the determining module is specifically configured to:

determine a first client identifier of the client and a client identifier corresponding to each piece of first audio;

determine target audio in the at least one piece of first audio according to the first client identifier and the client identifier corresponding to each piece of first audio, wherein a client identifier corresponding to the target audio is different from the first client identifier;

perform mixing processing on the target audio and the second audio to obtain the first mixed audio corresponding to the client.

17. The audio processing apparatus according to claim 15 or 16, wherein,

the determining module is specifically configured to:

acquire channel statuses of the recording channels between the cloud desktop server and the plurality of clients, wherein the channel statuses are a muted state or an unmuted state;

determine at least one target recording channel from the recording channels according to the channel statuses of the plurality of recording channels, wherein the at least one target recording channel is in the unmuted state;

receive the at least one piece of first audio through the at least one target recording channel;

the first receiving module is further configured to:

receive a mute request sent by a first client of the plurality of clients, and set the channel status of the recording channel between the first client and the cloud desktop server to the muted state according to the mute request; or,

receive an unmute request sent by the first client, and set the channel status of the recording channel between the first client and the cloud desktop server to the unmuted state according to the unmute request.

18. The audio processing apparatus according to any one of claims 15 to 17, further comprising a third receiving module and a creating module; wherein,

the third receiving module is configured to receive a login request sent by a second client of the plurality of clients, wherein the login request comprises verification information;

the creating module is configured to perform verification processing on the verification information, and create a recording channel between the cloud desktop server and the second client after the verification of the verification information is successful, wherein the recording channel is a socket channel.

19. An audio processing apparatus, applied to a client, comprising: a collecting module, a determining module, and a sending module, wherein,

the collecting module is configured to perform audio collection through a microphone to obtain first audio;

the determining module is configured to determine a channel status of a recording channel between the client and a cloud desktop server, wherein the channel status is a muted state or an unmuted state;

the sending module is configured to send the first audio to the cloud desktop server through the recording channel when the channel status is in the unmuted state.

20. The audio processing apparatus according to claim 19, further comprising: a displaying module, a storage module, and a receiving module, wherein,

the displaying module is configured to display a first page, and the first page comprises a status control of the recording channel;

the sending module is configured to send a mute request to the cloud desktop server in response to a mute operation on the status control; or send an unmute request to the cloud desktop server in response to an unmute operation on the status control;

wherein the storage module is configured to, when the channel status is in the unmuted state, store the first audio into a preset storage space, or discard the first audio;

wherein the receiving module is configured to receive a mixing result sent by the cloud desktop server, wherein the mixing result is obtained by the cloud desktop server performing mixing on audio from a further client and a meeting server.