(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 December 2022 (29.12.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/271295 A1

(51) International Patent Classification:
H04L 65/10 (2022.01)

(21) International Application Number:
PCT/US2022/029087

(22) International Filing Date:
13 May 2022 (13.05.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
202111027986 22 June 2021 (22.06.2021) IN

(71) Applicant: MICROSOFT TECHNOLOGY LICENSING, LLC [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventor: SRIVASTAVA, Alok; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(74) Agent: CHATTERJEE, Aaron C. et al.; Microsoft Technology Licensing, LLC, One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM,

(54) Title: DISTRIBUTED PROCESSING OF COMMUNICATION SESSION SERVICES

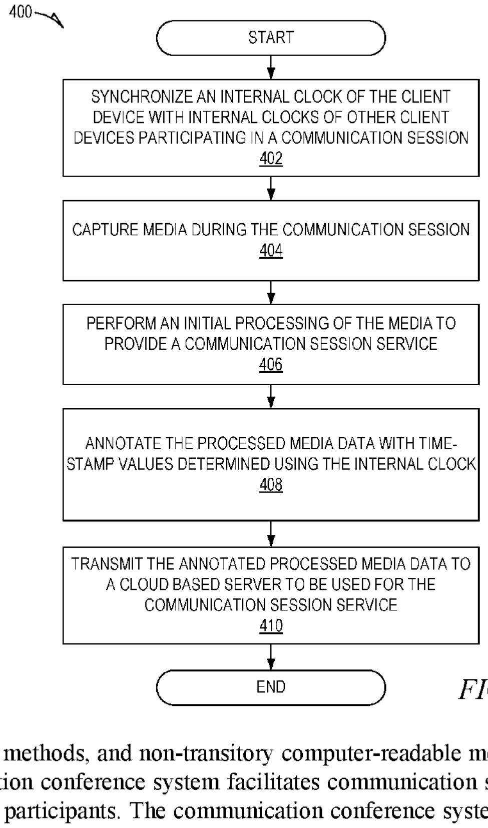
400
START
SYNCHRONIZE AN INTERNAL CLOCK OF THE CLIENT DEVICE WITH INTERNAL CLOCKS OF OTHER CLIENT DEVICES PARTICIPATING IN A COMMUNICATION SESSION
402
CAPTURE MEDIA DURING THE COMMUNICATION SESSION
404
PERFORM AN INITIAL PROCESSING OF THE MEDIA TO PROVIDE A COMMUNICATION SESSION SERVICE
406
ANNOTATE THE PROCESSED MEDIA DATA WITH TIME-STAMP VALUES DETERMINED USING THE INTERNAL CLOCK
408
TRANSMIT THE ANNOTATED PROCESSED MEDIA DATA TO A CLOUD BASED SERVER TO BE USED FOR THE COMMUNICATION SESSION SERVICE
410
END

FIG. 4

(57) Abstract: Disclosed are systems, methods, and non-transitory computer-readable media for distributed processing of communication session services. A communication conference system facilitates communication sessions (e.g., videoconferences, conference calls, etc.) between groups of meeting participants. The communication conference system uses a distributed architecture to provide communication session services. In the distributed architecture, at least a portion of the tasks related to providing a communication session service are performed at the individual client devices, rather than each task being performed at a centralized system. For example, to provide a transcription service, audio data (e.g., speech) captured at each client device may be converted to text at the client devices prior to being transmitted to the communication conference system. Similarly, additional tasks may also be performed at the client device, such as translating the text into another language, performing sentiment analysis, and the like.

WO 2022/271295 A1

[Continued on next page]

TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

DISTRIBUTED PROCESSING OF COMMUNICATION SESSION SERVICES

TECHNICAL FIELD

An embodiment of the present subject matter relates generally to communication sessions and, more specifically, to distributed processing of communication session services.

BACKGROUND

Communication conference systems are commonly used to conduct meetings and provide presentations online and via telephones. Communication conference systems receive media (e.g., video, audio) captured at each meeting participants device and share the data with the other meeting participants. This facilitates communication amongst the meeting participants as they can see and hear each other in real-time.

Some conference systems also provide various services in relation to the shared media. For example, a conference system may provide any of transcription services (e.g., speech to text), translation services, sentiment analysis, and the like. A central system performs the service based on media received from the various devices. For example, a centralized conference system may provide a transcription service by generating text from audio data received from the device.

Providing these types of services presents several technical challenges. For example, these services can be resource intensive as the centralized conference system is tasked with identifying the actors associated with the received media, synchronizing media, and performing any analysis and/or transformation based on the received media. These tasks become even more challenging when multiple languages are used by the meeting participants and/or the service is to be provided in multiple languages. Accordingly, improvements are needed.

BRIEF DESCRIPTION OF THE DRAWINGS

In the drawings, which are not necessarily drawn to scale, like numerals may describe similar components in different views. Like numerals having different letter suffixes may represent different instances of similar components. Some embodiments are illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which:

FIG. 1 shows a system for distributed processing of communication session services, according to some example embodiments.

FIG. 2. is a block diagram of a client-side application for distributed processing of communication session services, according to some example embodiments.

FIG. 3 is a block diagram of a communication conference system for distributed processing of communication session services, according to some example embodiments.

FIG. 4 is a flowchart showing a method for a client device providing distributed processing of communication session services, according to certain example embodiments.

FIG. 5 is a flowchart showing a method for a communication conference system providing distributed processing of communication session services, according to certain example embodiments.

FIGS. 6A and 6B are flowcharts showing client-side and server-side methods for providing distributed processing of a transcription service during a communication session, according to certain example embodiments.

FIG. 7 is a block diagram illustrating a representative software architecture, which may be used in conjunction with various hardware architectures herein described.

FIG. 8 is a block diagram illustrating components of a machine, according to some example embodiments, able to read instructions from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein.

DETAILED DESCRIPTION

In the following description, for purposes of explanation, various details are set forth in order to provide a thorough understanding of some example embodiments. It will be apparent, however, to one skilled in the art, that the present subject matter may be practiced without these specific details, or with slight alterations.

Reference in the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present subject matter. Thus, the appearances of the phrase “in one embodiment” or “in an embodiment” appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

For purposes of explanation, specific configurations and details are set forth in order to provide a thorough understanding of the present subject matter. However, it will be apparent to one of ordinary skill in the art that embodiments of the subject matter described may be practiced without the specific details presented herein, or in various combinations, as described herein. Furthermore, well-known features may be omitted or simplified in order not to obscure the described embodiments. Various examples may be given throughout this description. These are merely descriptions of specific embodiments. The scope or meaning of the claims is not limited to the examples given.

Disclosed are systems, methods, and non-transitory computer-readable media for distributed processing of communication session services. A communication conference system facilitates communication sessions (e.g., videoconferences, conference calls, etc.) between groups of meeting participants. A communication session is a conference established between meeting participants in which media (e.g., video, audio, other data) captured at each meeting participants client devices is shared with the client devices of the other meeting participants. The communication conference system allows users to schedule a communication session, and select meeting participants, as well as facilitates the transmission of media streams (e.g., video data, audio data) among the meeting participants during the communication session.

As explained earlier, current systems use a centralized framework to provide services in relation to communication sessions, which presents several technical problems. For example, a centralized system is tasked with receiving and processing media received from each of the client devices participating in the communication session. This may involve performing multiple individual tasks, such as identifying the actors associated with the received media, synchronizing media, and performing any analysis and/or transformation based on the received media. Performing each of these individual tasks at a centralized system can place a strain on the available computing resources and cause system latency. Accordingly, minimizing computing resource usage and latency is a technical problem faced when providing communication session services.

To alleviate these issues, the communication conference system of the present disclosure uses a distributed architecture to provide communication session services. In the distributed architecture, at least a portion of the tasks related to providing a communication session service are performed at the individual client devices, rather than each task being performed at a centralized system. For example, to provide a transcription service, audio data (e.g., speech) captured at each client device may be converted to text at the client devices prior to being transmitted to the communication conference system. Similarly, additional tasks may also be performed at the client device, such as translating the text into another language, performing sentiment analysis, and the like.

The client devices provide the processed media data to the communication conference system. For example, the client devices may provide the communication conference system with text generated from speech captured at the client device. The communication conference system may perform additional tasks in relation to the processed media data to provide the communication session service. For example, the communication conference system may aggregate the text received from the client devices into a singular transcript of the communication session.

Performing a portion of the tasks at the client devices reduces the number of operations that are performed by the communication conference system. This provides a technical solution by reducing the computing resources usage at the communication conference system, thereby reducing the strain on the available computing resources of the communication conference system and overall system latency.

Performing a portion of the tasks at the client devices may also simplify processing of the media captured by each client device. In some cases, meeting participants may be speaking different languages and/or may with to have the media processed based on a different language and/or alphabet. For example, a user may wish to have to have their speech transcribed into a specified language. To provide this type of functionality using a centralized architecture, the centralized system is tasked with correctly identifying the user associated with the captured speech and processing the data based on the selected configurations. This task is both difficult and prone to errors. In contrast, a client device can easily process media captured at the client device based on the defined configurations because the client device does not have to identify the source of the media or choose from multiple configurations for processing the data.

FIG. 1 shows a system 100 for distributed processing of communication session services, according to some example embodiments. As shown, multiple devices (i.e., client device 102, client device 104, and communication conference system 106) are connected to a communication network 108 and configured to communicate with each other through use of the communication network 108. The communication network 108 is any type of network, including a local area network (LAN), such as an intranet, a wide area network (WAN), such as the internet, or any combination thereof. Further, the communication network 108 may be a public network, a private network, or a combination thereof. The communication network 108 is implemented using any number of communication links associated with one or more service providers, including one or more wired communication links, one or more wireless communication links, or any combination thereof. Additionally, the communication network 108 is configured to support the transmission of data formatted using any number of protocols.

Multiple computing devices can be connected to the communication network 108. A computing device is any type of general computing device capable of network communication with other computing devices. For example, a computing device can be a personal computing device such as a desktop or workstation, a business server, or a portable computing device, such as a laptop, smart phone, or a tablet personal computer (PC). A computing device can include some or all of the features, components, and peripherals of the machine 800 shown in FIG. 8.

To facilitate communication with other computing devices, a computing device includes a communication interface configured to receive a communication, such as a request, data, and the like, from another computing device in network communication with the computing device and pass the communication along to an appropriate module running on the computing device. The communication interface also sends a communication to another computing device in network communication with the computing device.

In the system 100, users interact with and utilize the functionality of the communication conference system 106 by using the client devices 102 and 104 that are connected to the communication network 108 by direct and/or indirect communication. Although the system 100 includes only two client devices 102, 104, this is for ease of explanation and is not meant to be limiting. One skilled in the art would appreciate that the system 100 can include any number of client devices 102, 104. Further, the communication conference system 106 may concurrently accept connections from and interact with any number of client devices 102, 104. The communication conference system 106 also supports connections from a variety of different types of client devices 102, 104, such as desktop computers; mobile computers; mobile communications devices, (e.g., mobile phones, smart phones, tablets); smart televisions; set-top boxes; and/or any other network enabled computing devices. Hence, the client devices 102 and 104 may be of varying type, capabilities, operating systems, and so forth.

A user interacts with the communication conference system 106 via a client-side application 110 installed on the client devices 102 and 104. In some embodiments, the client-side application 110 includes a component specific to the communication conference system 106. For example, the component may be a stand-alone application, one or more application plug-ins, and/or a browser extension. However, the client-side application 110 may also be a third-party application, such as a web browser, that resides on the client devices 102 and 104 and is configured to communicate with the communication conference system 106. In either case, the client-side application 110 presents a user interface (UI) for the user to interact with the communication conference system 106. For example, the user interacts with the communication conference system 106 via a client-side application 110 integrated with the file system or via a webpage displayed using a web browser application.

The communication conference system 106 is one or more computing devices configured to facilitate and manage communication session between various meeting participants. For example, the communication conference system 106 can facilitate a communication session between client devices 102 and 104, where a meeting participant using one client device 102 can send and receive media (e.g., audio, video, shared data) with a meeting participant using another client device 104 and vice versa.

The communication conference system 106 allows users to schedule a communication session, and select meeting participants, as well as facilitates the transmission of media streams (e.g., video data, audio data) among the meeting participants during the communication session. For example, the communication conference system 106 established a connection with each of the client devices 102, 104, such as a WebSocket connection, that allows the communication conference system 106 to initiate media streams between the communication conference system 106 and the client devices 102, 104. The media streams allow media captured at a client device 102, 104 to be provided to the communication conference system 106 as well as allows the communication conference system 106 to provide media to the client devices 102, 104.

To facilitate a communication session, the communication conference system 106 receives media streams, including audio data, video data, etc., from one of the client devices 102, and transmits the received media streams to the other client device 104, where it can be presented by client device 104, and vice versa. This allows the meeting participants at each client device 102, 104 to receive and share data, including audio and/or video data, thereby enabling the meeting participants to engage in a real time meeting even though the two participants may be in different geographic locations.

The communication conference system 106 also provides for communication session services in relation to a communication session. A communication session service is a service provided based on the media shared during a communication session. For example, a communication session service may include generating a text transcript of speech spoken during a communication session, translating speech spoken during a communication session, performing sentiment analysis, performing a machine learning analysis, and the like.

As explained earlier, current systems use a centralized framework to provide services in relation to communication sessions, which presents several technical problems. For example, a centralized system is tasked with receiving and processing media received from each of the client devices participating in the communication session. This may involve performing multiple individual tasks, such as identifying the actors associated with the received media, synchronizing media, and performing any analysis and/or transformation based on the received media. Performing each of these individual tasks at a centralized system can place a strain on the available computing resources and cause system latency. Accordingly, minimizing computing resource usage and latency is a technical problem faced when providing communication session services.

To alleviate these issues, the communication conference system 106 uses a distributed architecture to provide communication session services. In the distributed architecture, at least a portion of the tasks related to providing a communication session service are performed at the individual client devices 102, 104, rather than each of the tasks being performed at a communication conference system 106. For example, to provide a transcription service, audio data (e.g., speech) captured at each client device 102, 104 may be converted to text at the client devices 102, 104 prior to being transmitted to the communication conference system 106. Similarly, additional tasks may also be performed at the client device 102, 104, such as translating the text into another language, performing sentiment analysis, and the like.

The client devices 102, 104 provide the processed media data to the communication conference system 106. For example, the client devices 102, 104 may provide the communication conference system 106 with text generated from speech captured at the client devices 102, 104. The communication conference system 106 may perform additional tasks in relation to the processed media data to provide the communication session service. For example, the communication conference system 106 may aggregate the text received from the client devices 102, 104 into a singular transcript of the communication session.

The communication conference system 106 may aggregate or otherwise perform additional tasks in relation to the processed media based on time-stamp values provided with the processed media. The time-stamp values may indicate times at which speech was detected by the client devices 102, 104. The communication conference system 106 may use the time-stamp values to aggregate the processed media into a chronological order, such as by generating a transcript of a communication session that includes speech spoken by the meeting participants in the order in which the speech was captured at each client device 102, 104. The client devices 102, 104 may generate the time-stamp values using an internal clock available to each client device 102, 104.

In some embodiments, the communication conference system 106 may synchronize the internal clocks of the client devices 102, 104 participating in a communication session. For example, the communication conference system 106 may use any of a variety of known device synchronization techniques. Synchronizing the internal clocks of the client devices 102, 104 provides for time-stamp values that are relatively accurate to each other, thereby allowing the communication conference system 106 to properly aggregate the processed media data in the correct chronological order.

FIG. 2 is a block diagram of a client-side application 110 for distributed processing of communication session services, according to some example embodiments. To avoid obscuring the inventive subject matter with unnecessary detail, various functional components (e.g., modules) that are not germane to conveying an understanding of the inventive subject matter have been omitted from FIG. 2. However, a skilled artisan will readily recognize that various additional functional components may be supported to facilitate additional functionality that is not specifically described herein. Furthermore, the various functional modules depicted in FIG. 2 may reside on a single computing device or may be distributed across several computing devices in various arrangements such as those used in cloud-based architectures.

As shown, the client-side application 110 includes a communication session management component 202, a clock synchronization component 204, a media capturing component 206, a media processing component 208, a time-stamp component 210, a media stream component 212, and a data storage 214.

The communication session management component 202 provides functionality associated with engaging in communication sessions. For example, the communication session management component 202 provides a user interface that enables a user to schedule, join, and/of configure a communication session. The user interface may include user interface elements, such a buttons, text boxes, and the like, that enable a user to provide input and utilize the functionality of the communication conference system 106.

Configuring a communication session may include configuring communication session services provided in relation to a communication session. For example, the user interface may enable a user to select a language that the user will be using during the communication session. The user interface may also enable a user to select a language and/or alphabet that should be used when providing a communication session service. For example, a user may select to have speech converted to text in a specified language, such as French, German, or Spanish, and using the corresponding alphabet and/or or special characters associated with each.

The communication session management component 202 may provide data provided by a user to the communication conference system 106, the other components of the client-side application 110 and/or store the data in data storage 214.

The clock synchronization component 204 provides functionality for synchronizing an internal clock of a client device 102, 104. For example, the clock synchronization component 204 communicates with the communication conference system 106 to receive data used to synchronize the internal clock as well as configures the internal clock of the client device 102, 104 based on the received data.

The media capturing component 206 captures media as part of a communication session. The media may include any of a variety of types of media, such as image data (e.g., pictures, video), audio data, and/or other types of data, such as shared data (e.g., screen share). The media capturing component 206 may capture the media using one or more sensors of the client device 102, 104. For example, the media capturing component 206 may capture media using image sensors (e.g., cameras), audio sensors (e.g., microphones), and the like. The media capturing component 206 may provide the captured media to the other components of the client-side application 110 and/or store the media in data storage 214.

The media processing component 208 processes media captured by the media capturing component 206 to provide a portion of a communication session service. As explained earlier, the communication conference system 106 provides communication session services using a distributed architecture to alleviate the technical problems associated with current systems that utilize a centralized architecture. For example, using distributed architecture reduces strain on computing resources at the communication conference system 106 and thereby decreases system latency.

In a distributed architecture, a portion of the tasks associated with providing a communication session service, such as a transcription service, translation service, and the like, are performed at the client device 102, 104. The media processing component 208 processes media captured by the media capturing component 206 to provide the portion of a communication session service that is distributed to the client devices 102, 104. For example, to provide a transcription service the media processing component 208 generates text from speech captured at the client device 102, 104 by the media capturing component 206. To provide a translation service, the media processing component 208 may both generate text from speech captured at the client device 102, 104 by the media capturing component 206 as well as translate the text into a specified language. These are just two examples and are not meant to be limiting. The media processing component 208 may process media to provide any of a number of communication session services, such as sentiment analysis, translation, transcription, and the like.

The media processing component 208 processes the media based on the configurations defined by the user of the client device 102, 104. For example, the media processing component 208 may process speech based on the language and/or alphabet defined by the user. This allows for each client device 102, 104 that is engaged in a communication session to process the media captured at the respective client device 102, 104 based on the specific configurations defined by the user of the client device 102, 104. As a result, the overall complexity of providing the communication session service is reduced and the operating speed at which the communication session service is provided in increased.

The media processing component 208 may access the configurations defined by the user from the data storage 214 and/or from the communication session management component 202.

The time-stamp component 210 annotates the processed media data generated by the media processing component 208 with times stamp values. The times stamp values indicate the times at which associated processed media was captured at the client device 102, 104. For example, the time-stamp values may indicate times at which speech converted to text was captured by the client device 102, 104. As another example, the time-stamp value may indicate times at which images processed into other data were captured by the client device 102, 104.

The time-stamp component 210 determines the time-stamp values using the internal clock of the client device 102, 104. For example, the time-stamp component uses a time provide by the internal clock at which sensor data (e.g., speech) is captured to determine the time-stamp value associated with the speech and/or text generated from the captured speech. The time-stamp values may be annotated as metadata to text generated by the media processing component 208.

The media stream component 212 facilitates transfer of media and other data between a client device 102, 104 and the communication conference system 106 and/or other client devices 102, 104 engaged in a communication session. The media may include video, audio, shared data, and the like, that is captured during a communication session. The media stream component 212 may also provide the communication conference system 106 with processed sensor data generated by the media processing component 208 and annotated by the time-stamp component 210.

FIG. 3 is a block diagram of a communication conference system 106, according to some example embodiments To avoid obscuring the inventive subject matter with unnecessary detail, various functional components (e.g., modules) that are not germane to conveying an understanding of the inventive subject matter have been omitted from FIG. 3. However, a skilled artisan will readily recognize that various additional functional components may be supported by the communication conference system 106 to facilitate additional functionality that is not specifically described herein. Furthermore, the various functional modules depicted in FIG. 3 may reside on a single computing device or may be distributed across several computing devices in various arrangements such as those used in cloud-based architectures.

As shown, the communication conference system 106 includes a communication session management component 302, a media stream forwarding component 304, a synchronization component 306, a processed media data receiving component 308, a communication session service component 310, an output component 312, and a data storage 314.

The communication session management component 302 facilitates management and initialization of communication sessions. For example, the communication session management component 302 provides server-side functionality associated with the communication session management component 202 operating on the client device 102, 104. This includes providing a user interface enabling users to schedule and configure communication session, as well as define communication session services to be provided in relation to a communication session. The communication session management component 302 may store data associated with a scheduled communication session in the data storage 314. For example, the data may include the day/time of the communication session, invited meeting participants, meeting password, associated configurations, selected communication session services, and the like.

The communication session management component 302 may allocate resources to the scheduled communication session, such as by defining a contact identifier (e.g., phone number, web address) for joining the communication session, as well as reserving and/or initializing computing resources to facilitate the communication session. The communication session management component 302 uses the stored data to initiate a communication session. For example, the communication session management component 302 may use the stored data to authenticate requests to join the communication session, such as by confirming that the request is associated with an invited participant, valid password, and the like.

The communication session management component 302 also establishes connections with each of the client devices 102, 104 participating in a communication session. The connections allow for media and other data to be shared between the client devices 102, 104 and the communication conference system 106. For example, a connection may be a WebSocket connection that allows for a media stream between the communication conference system 106 and the client devices 102, 104.

The communication session management component 302 also communicates with the other components of the communication conference system 106 to initiate the communication session. For example, the communication session management component 302 may provide the media stream forwarding component 304 with data used to properly forward media received as part of a communication session. Similarly, the communication session management component 302 may notify the synchronization component 306 to synchronize the internal clocks of the client devices 102, 104, the communication session service component 310 to provide specified communication session services, and the like.

The media stream forwarding component 304 receives media from the client device 102, 104 that was captured as part of a communication session and forwards the media to the other client devices 102, 104 participating in the communication session. For example, the media stream forwarding component 304 may receive media, such as audio, video, and/or shared data, captured at each client device 102, 104 participating in the communication session and forward the received media to the other client devices 102, 104 participating in the communication session, where it may be presented to the other meeting participants. This allows the meeting participants to communicate with each other to see, hear and share data with each other in near real-time, thereby facilitating communication amongst the meeting participants.

The synchronization component 306 facilitates synchronization of internal clocks of the client devices 102, 104. Synchronizing the internal clocks of the client devices 102, 104 causes the internal clocks to operate at synchronized times such that that time-stamp values recorded using the internal clocks of each client device 102, 104 are accurate relative to each other. The synchronization component 306 may synchronize the internal clocks of the client devices 102, 104 using any suitable methods. For example, the synchronization component 306 may provide each client device 102, 104 with instructions for determining a moment at which to synchronize the internal clock of the client device 102, 104 to a designated value.

The processed media data receiving component 308 receives processed media data from the client devices 102, 104. The processed media data is data that has been partially processed at a client device 102, 104 to provide a communication session service. In some embodiments, the processed media data may include text generated at the client device 102, 104 based on speech captured at the client device 102, 104. The text may be a text representation of the speech and/or the speech translated into a selected language. In some embodiments, the text may be anonymized, normalized, and/or otherwise modified.

The processed media data may be annotated with data describing the source of the processed media data, such as an identifier identifying the communication session associated with the processed media data, the client device 102, 104 from which the processed media data is received, the user associated with the received processed media data, and the like. The processed media data may also be annotated with time-stamp values indicating times associated with the processed media data. For example, the time-stamp values may indicate times at which speech converted to text was captured by the client device 102, 104.

The processed media data receiving component 308 provides the received processed media data to the other component of the communication conference system 106 and/or stores the processed media data in the data storage 314.

The communication session service component 310 further processes the processed media data received from the client devices 102, 104 to provide a communication session service. As explained earlier, the communication conference system 106 utilizes a distributed architecture to provide communication session services in which a portion of the tasks associated with providing the communication session service are performed at the client devices 102, 104 and a portion the remaining tasks are performed at the communication conference system 106. For example, the provide a transcription service, speech is converted to text at the client devices 102, 104 and then aggregated into a full transcript at the communication conference system 106.

The communication session service component 310 performs the additional tasks associated with providing a communication session service in a distributed architecture. As part of this process, the communication session service component 310 accesses the processed media data received from the client devices 102, 104 engaged in the communication session. For example, the communication session service component 310 accessed the processed media data from the processed media data receiving component 308 and/or from the data storage 314.

The communication session service component 310 may identify the processed media data associated with the communication session based on identifying data associated with the processed media data. For example, the communication session service component 310 may identify the processed media data based on a unique identifier for the communication session that is associated with the processed media data.

The communication session service component 310 may further process the processed media data received from the client devices 102, 104 based on the time-stamp values annotated to the processed media data. For example, the communication session service component 310 may aggregate text received from each of the client devices 102, 104 into a singular text document such that the text is ordered sequentially based on the time-stamp values. This may provide for a transcript and/or translation of the communication session that is ordered based on the times at which the converted speech of each meeting participant was captured.

The communication session service component 310 may perform additional functionality based on the aggregated text, such as generating an input and/or training data for a machine learning models, performing natural language processing, generating automated response message or suggestions, and the like. The communication session service component 310 may provide any generated output associated with the provided communication session service to the other components of the communication conference system 106 and/or store the output in the data storage 314.

The output component 312 provides an output generated by the communication session service component 310 to an authorized and/or otherwise designated user. For example, the output component 312 may facilitate access to the output to via a client device 102, 104 that has been authenticated to access the output, such as by providing an appropriate username, password, security code, and the like. The output component 312 may also transmit a message providing access to the output to designated users. For example, the output component 312 may transmit messages, such as email, text messages, and the like, to designated contact identifiers (e.g., email addresses, phone numbers) associated with the communication session. In some embodiments, the messages may include the output, such as by including the output in the body of the message and/or as an attachment to the message. In some embodiments, the message may include a link or other type of data that may be used to access the output.

FIG. 4 is a flowchart showing a method for a client device 102, 104 providing distributed processing of communication session services, according to certain example embodiments. The method 400 may be embodied in computer readable instructions for execution by one or more processors such that the operations of the method 400 may be performed in part or in whole by the client-side application 110; accordingly, the method 400 is described below by way of example with reference thereto. However, it shall be appreciated that at least some of the operations of the method 400 may be deployed on various other hardware configurations and the method 400 is not intended to be limited to the client-side application 110.

At operation 402, the clock synchronization component 204 synchronizes an internal clock of the client device 102, 104 with internal clocks of other client devices 102, 104 participating in a communication session. The clock synchronization component 204 provides functionality for synchronizing an internal clock of a client device 102, 104. For example, the clock synchronization component 204 communicates with the communication conference system 106 to receive data used to synchronize the internal clock as well as configures the internal clock of the client device 102, 104 based on the received data.

At operation 404, the media capturing component 206 captures media during the communication session. The media may include any of a variety of types of media, such as image data (e.g., pictures, video), audio data, and/or other types of data, such as shared data (e.g., screen share).

The media capturing component 206 may capture the media using one or more sensors of the client device 102, 104. For example, the media capturing component 206 may capture media using image sensors (e.g., cameras), audio sensors (e.g., microphones), and the like. The media capturing component 206 may provide the captured media to the other components of the client-side application 110 and/or store the media in data storage 214.

At operation 406, the media processing component 208 performs an initial processing of the media to provide a communication session service. As explained earlier, the communication conference system 106 provides communication session services using a distributed architecture to alleviate the technical problems associated with current systems that utilize a centralized architecture. For example, using distributed architecture reduces strain on computing resources at the communication conference system 106 and thereby decreases system latency.

In a distributed architecture, a portion of the tasks associated with providing a communication session service, such as a transcription service, translation service, and the like, are performed at the client device 102, 104. The media processing component 208 processes media captured by the media capturing component 206 to provide the portion of a communication session service that is distributed to the client devices 102, 104. For example, to provide a transcription service the media processing component 208 generates text from speech captured at the client device 102, 104 by the media capturing component 206. To provide a translation service, the media processing component 208 may both generate text from speech captured at the client device 102, 104 by the media capturing component 206 as well as translate the text into a specified language. These are just two examples and are not meant to be limiting. The media processing component 208 may process media to provide any of a number of communication session services, such as sentiment analysis, translation, transcription, and the like.

The media processing component 208 processes the media based on the configurations defined by the user of the client device 102, 104. For example, the media processing component 208 may process speech based on the language and/or alphabet defined by the user. This allows for each client device 102, 104 that is engaged in a communication session to process the media captured at the respective client device 102, 104 based on the specific configurations defined by the user of the client device 102, 104. As a result, the overall complexity of providing the communication session service is reduced and the operating speed at which the communication session service is provided in increased.

The media processing component 208 may access the configurations defined by the user from the data storage 214 and/or from the communication session management component 202.

At operation 408, the time-stamp component 210 annotates the processed media data with time stamp values determined using the internal clock. The times stamp values indicate the times at which associated processed media was captured at the client device 102, 104. For example, the time-stamp values may indicate times at which speech converted to text was captured by the client device 102, 104. As another example, the time-stamp value may indicate times at which images processed into other data were captured by the client device 102, 104.

The time-stamp component 210 determines the time-stamp values using the internal clock of the client device 102, 104. For example, the time-stamp component uses a time provide by the internal clock at which sensor data (e.g., speech) is captured to determine the time-stamp value associated with the speech and/or text generated from the captured speech. The time-stamp values may be annotated as metadata to text generated by the media processing component 208.

At operation 410, the media stream component 212 transmits the annotated processed media data to a cloud-based server to be used for the communication session service. The media stream component 212 facilitates transfer of media and other data between a client device 102, 104 and the communication conference system 106 and/or other client devices 102, 104 engaged in a communication session. The media may include video, audio, shared data, and the like, that is captured during a communication session. The media stream component 212 may also provide the communication conference system 106 with processed sensor data generated by the media processing component 208 and annotated by the time-stamp component 210.

FIG. 5 is a flowchart showing a method 500 for generating a communication session providing configurable group-based media streams, according to certain example embodiments. The method 500 may be embodied in computer readable instructions for execution by one or more processors such that the operations of the method 500 may be performed in part or in whole by the communication conference system 106; accordingly, the method 500 is described below by way of example with reference thereto. However, it shall be appreciated that at least some of the operations of the method 500 may be deployed on various other hardware configurations and the method 500 is not intended to be limited to the communication conference system 106.

At operation 502, the synchronization component 306 transmits a synchronization input to synchronize internal clocks of client devices 102, 104 engaged in a communication session. The synchronization component 306 facilitates synchronization of internal clocks of the client devices 102, 104. Synchronizing the internal clocks of the client devices 102, 104 causes the internal clocks to operate at synchronized times such that that time-stamp values recorded using the internal clocks of each client device 102, 104 are accurate relative to each other. The synchronization component 306 may synchronize the internal clocks of the client devices 102, 104 using any suitable methods. For example, the synchronization component 306 may provide each client device 102, 104 with a synchronization input providing instructions for determining a moment at which to synchronize the internal clock of the client device 102, 104 to a designated value.

At operation 504, the processed media data receiving component 308 receives annotated processed media data from the client devices 102, 104 engaged in the communication session. The processed media data is data that has been partially processed at a client device 102, 104 to provide a communication session service. In some embodiments, the processed media data may include text generated at the client device 102, 104 based on speech captured at the client device 102, 104. The text may be a text representation of the speech and/or the speech translated into a selected language. In some embodiments, the text may be anonymized, normalized, and/or otherwise modified.

The processed media data may be annotated with data describing the source of the processed media data, such as an identifier identifying the communication session associated with the processed media data, the client device 102, 104 from which the processed media data is received, the user associated with the received processed media data, and the like. The processed media data may also be annotated with time-stamp values indicating times associated with the processed media data. For example, the time-stamp values may indicate times at which speech converted to text was captured by the client device 102, 104.

The processed media data receiving component 308 provides the received processed media data to the other component of the communication conference system 106 and/or stores the processed media data in the data storage 314.

At operation 506, the communication session service component 310 performs a subsequent processing of the annotated processed media data to provide a communication session service. As explained earlier, the communication conference system 106 utilizes a distributed architecture to provide communication session services in which a portion of the tasks associated with providing the communication session service are performed at the client devices 102, 104 and a portion the remaining tasks are performed at the communication conference system 106. For example, the provide a transcription service, speech is converted to text at the client devices 102, 104 and then aggregated into a full transcript at the communication conference system 106.

The communication session service component 310 performs the additional tasks associated with providing a communication session service in a distributed architecture. As part of this process, the communication session service component 310 accesses the processed media data received from the client devices 102, 104 engaged in the communication session. For example, the communication session service component 310 accessed the processed media data from the processed media data receiving component 308 and/or from the data storage 314.

The communication session service component 310 may identify the processed media data associated with the communication session based on identifying data associated with the processed media data. For example, the communication session service component 310 may identify the processed media data based on a unique identifier for the communication session that is associated with the processed media data.

The communication session service component 310 may further process the processed media data received from the client devices 102, 104 based on the time-stamp values annotated to the processed media data. For example, the communication session service component 310 may aggregate text received from each of the client devices 102, 104 into a singular text document such that the text is ordered sequentially based on the time-stamp values. This may provide for a transcript and/or translation of the communication session that is ordered based on the times at which the converted speech of each meeting participant was captured.

The communication session service component 310 may perform additional functionality based on the aggregated text, such as generating an input and/or training data for a machine learning models, performing natural language processing, generating automated response message or suggestions, and the like. The communication session service component 310 may provide any generated output associated with the provided communication session service to the other components of the communication conference system 106 and/or store the output in the data storage 314.

At operation 508, the output component 312 provide an output based on the subsequent processing of the of the annotated processed media data. The output component 312 provides an output generated by the communication session service component 310 to an authorized and/or otherwise designated user. For example, the output component 312 may facilitate access to the output to via a client device 102, 104 that has been authenticated to access the output, such as by providing an appropriate username, password, security code, and the like. The output component 312 may also transmit a message providing access to the output to designated users. For example, the output component 312 may transmit messages, such as email, text messages, and the like, to designated contact identifiers (e.g., email addresses, phone numbers) associated with the communication session. In some embodiments, the messages may include the output, such as by including the output in the body of the message and/or as an attachment to the message. In some embodiments, the message may include a link or other type of data that may be used to access the output.

FIGS. 6A and 6B are flowcharts showing client-side and server-side methods 600, 650 for providing distributed processing of a transcription service during a communication session, according to certain example embodiments. The methods 600, 650 may be embodied in computer readable instructions for execution by one or more processors such that the operations of the methods 600, 650 may be performed in part or in whole by the client-side application 110 and the communication conference system 106; accordingly, the methods 600, 650 are described below by way of example with reference thereto. However, it shall be appreciated that at least some of the operations of the methods 600, 650 may be deployed on various other hardware configurations and the methods 600, 650 is not intended to be limited to the client-side application and the communication conference system 106.

FIG. 6A shows a client-side method 600 for providing distributed processing of a transcription service during a communication session, according to certain example embodiments.

At operation 602, the clock synchronization component 204 receives a synchronization input from a cloud based server (e.g., the communication conference system 106) to synchronize an internal clock of the a client device 102, 104 with internal clocks of other client devices 102, 104 participating in a communication session. The synchronization input may provide instructions for determining a moment at which to synchronize the internal clock of the client device 102, 104 to a designated value. The clock synchronization component 204 configures the internal clock of the client device 102, 104 based on the instructions provided in the synchronization input.

At operation 604, the media capturing component 206 captures, with a microphone, speech spoken during the communication session.

At operation 606, the time-stamp component 210 records time-stamp values at which the speech was captured.

At operation 608, the media processing component 208 generates text content from the speech that was captured and the time-stamp component 210 associates the time-stamp values with the text content. For example, the time-stamp may annotate the text content with the time-stamp values.

At operation 610, the media stream component 212 transmits the text content to the cloud-based sever to be used for generating a transcript of the communication session.

FIG. 6B shows a server-side method 650 for providing distributed processing of a transcription service during a communication session, according to certain example embodiments.

At operation 652, the processed media data receiving component 308 receives a communication session identifier, first text content, and a first-time stamp value from a first client device 102 engaged in a communication session. The first text content corresponds to speech captured at the first client device 102. For example, the first text content was generated at the first client device 102 to provide a transcript of the communication session. The first time-stamp value indicate a time at which the speech was captured at the first client device 102.

At operation 654, the processed media data receiving component 308 receives the communication session identifier, second text content, and a second time-stamp value from a second client device 104 engaged in a communication session. The second text content corresponds to speech captured at the second client device 104. For example, the second text content was generated at the second client device 104 to provide the transcript of the communication session. The second time-stamp value indicate a time at which the speech was captured at the second client device 104.

At operation 656, the communication session service component 310 aggregates the first text and the second text into a transcript of the communication session based on the first time-stamp value and the second time-stamp value. For example, the communication session service component 310 may aggregate the first text and the second text into chronological order based on the first time-stamp value and the second time-stamp value. The resulting transcript presents text content representing speech spoken by participants on the communication session in the chronological order in which the speech was spoken.

SOFTWARE ARCHITECTURE

FIG. 7 is a block diagram illustrating an example software architecture 706, which may be used in conjunction with various hardware architectures herein described. FIG. 7 is a non-limiting example of a software architecture 706 and it will be appreciated that many other architectures may be implemented to facilitate the functionality described herein. The software architecture 706 may execute on hardware such as machine 800 of FIG. 8 that includes, among other things, processors 804, memory 814, and (input/output) I/O components 818. A representative hardware layer 752 is illustrated and can represent, for example, the machine 800 of FIG. 8. The representative hardware layer 752 includes a processing unit 754 having associated executable instructions 704. Executable instructions 704 represent the executable instructions of the software architecture 706, including implementation of the methods, components, and so forth described herein. The hardware layer 752 also includes memory and/or storage modules 756, which also have executable instructions 704. The hardware layer 752 may also comprise other hardware 758.

In the example architecture of FIG. 7, the software architecture 706 may be conceptualized as a stack of layers where each layer provides particular functionality. For example, the software architecture 706 may include layers such as an operating system 702, libraries 720, frameworks/middleware 718, applications 716, and a presentation layer 714. Operationally, the applications 716 and/or other components within the layers may invoke application programming interface (API) calls 708 through the software stack and receive a response such as messages 712 in response to the API calls 708. The layers illustrated are representative in nature and not all software architectures have all layers. For example, some mobile or special purpose operating systems may not provide a frameworks/middleware 718, while others may provide such a layer. Other software architectures may include additional or different layers.

The operating system 702 may manage hardware resources and provide common services. The operating system 702 may include, for example, a kernel 722, services 724, and drivers 726. The kernel 722 may act as an abstraction layer between the hardware and the other software layers.

For example, the kernel 722 may be responsible for memory management, processor management (e.g., scheduling), component management, networking, security settings, and so on. The services 724 may provide other common services for the other software layers. The drivers 726 are responsible for controlling or interfacing with the underlying hardware. For instance, the drivers 726 include display drivers, camera drivers, Bluetooth® drivers, flash memory drivers, serial communication drivers (e.g., Universal Serial Bus (USB) drivers), Wi-Fi® drivers, audio drivers, power management drivers, and so forth, depending on the hardware configuration.

The libraries 720 provide a common infrastructure that is used by the applications 716 and/or other components and/or layers. The libraries 720 provide functionality that allows other software components to perform tasks in an easier fashion than to interface directly with the underlying operating system 702 functionality (e.g., kernel 722, services 724, and/or drivers 726). The libraries 720 may include system libraries 744 (e.g., C standard library) that may provide functions such as memory allocation functions, string manipulation functions, mathematical functions, and the like. In addition, the libraries 720 may include API libraries 746 such as media libraries (e.g., libraries to support presentation and manipulation of various media format such as MPEG4, H.264, MP3, AAC, AMR, JPG, PNG), graphics libraries (e.g., an OpenGL framework that may be used to render 2D and 3D in a graphic content on a display), database libraries (e.g., SQLite that may provide various relational database functions), web libraries (e.g., WebKit that may provide web browsing functionality), and the like. The libraries 720 may also include a wide variety of other libraries 748 to provide many other APIs to the applications 716 and other software components/modules.

The frameworks/middleware 718 (also sometimes referred to as middleware) provide a higher-level common infrastructure that may be used by the applications 716 and/or other software components/modules. For example, the frameworks/middleware 718 may provide various graphical user interface (GUI) functions, high-level resource management, high-level location services, and so forth. The frameworks/middleware 718 may provide a broad spectrum of other APIs that may be used by the applications 716 and/or other software components/modules, some of which may be specific to a particular operating system 702 or platform.

The applications 716 include built-in applications 738 and/or third-party applications 740. Examples of representative built-in applications 738 may include, but are not limited to, a contacts application, a browser application, a book reader application, a location application, a media application, a messaging application, and/or a game application. Third-party applications 740 may include an application developed using the ANDROID™ or IOS™ software development kit (SDK) by an entity other than the vendor of the particular platform, and may be mobile software running on a mobile operating system such as IOS™, ANDROID™, WINDOWS® Phone, or other mobile operating systems. The third-party applications 740 may invoke the API calls 708 provided by the mobile operating system (such as operating system 702) to facilitate functionality described herein.

The applications 716 may use built in operating system functions (e.g., kernel 722, services 724, and/or drivers 726), libraries 720, and frameworks/middleware 718 to create UIs to interact with users of the system. Alternatively, or additionally, in some systems, interactions with a user may occur through a presentation layer, such as presentation layer 714. In these systems, the application/component "logic" can be separated from the aspects of the application/component that interact with a user.

FIG. 8 is a block diagram illustrating components of a machine 800, according to some example embodiments, able to read instructions 704 from a machine-readable medium (e.g., a machine-readable storage medium) and perform any one or more of the methodologies discussed herein. Specifically, FIG. 8 shows a diagrammatic representation of the machine 800 in the example form of a computer system, within which instructions 810 (e.g., software, a program, an application, an applet, an app, or other executable code) for causing the machine 800 to perform any one or more of the methodologies discussed herein may be executed. As such, the instructions 810 may be used to implement modules or components described herein. The instructions 810 transform the general, non-programmed machine 800 into a particular machine 800 programmed to carry out the described and illustrated functions in the manner described. In alternative embodiments, the machine 800 operates as a standalone device or may be coupled (e.g., networked) to other machines. In a networked deployment, the machine 800 may operate in the capacity of a server machine or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine 800 may comprise, but not be limited to, a server computer, a client computer, a PC, a tablet computer, a laptop computer, a netbook, a set-top box (STB), a personal digital assistant (PDA), an entertainment media system, a cellular telephone, a smart phone, a mobile device, a wearable device (e.g., a smart watch), a smart home device (e.g., a smart appliance), other smart devices, a web appliance, a network router, a network switch, a network bridge, or any machine 800 capable of executing the instructions 810, sequentially or otherwise, that specify actions to be taken by machine 800. Further, while only a single machine 800 is illustrated, the term "machine" shall also be taken to include a collection of machines that individually or jointly execute the instructions 810 to perform any one or more of the methodologies discussed herein.

The machine 800 may include processors 804, memory/storage 806, and I/O components 818, which may be configured to communicate with each other such as via a bus 802. The memory/storage 806 may include a memory 814, such as a main memory, or other memory storage, and a storage unit 816, both accessible to the processors 804 such as via the bus 802. The storage unit 816 and memory 814 store the instructions 810 embodying any one or more of the methodologies or functions described herein. The instructions 810 may also reside, completely or partially, within the memory 814, within the storage unit 816, within at least one of the processors 804 (e.g., within the processor's cache memory), or any suitable combination thereof, during execution thereof by the machine 800. Accordingly, the memory 814, the storage unit 816, and the memory of processors 804 are examples of machine-readable media.

The I/O components 818 may include a wide variety of components to receive input, provide output, produce output, transmit information, exchange information, capture measurements, and so on. The specific I/O components 818 that are included in a particular machine 800 will depend on the type of machine. For example, portable machines such as mobile phones will likely include a touch input device or other such input mechanisms, while a headless server machine will likely not include such a touch input device. It will be appreciated that the I/O components 818 may include many other components that are not shown in FIG. 8. The I/O components 818 are grouped according to functionality merely for simplifying the following discussion and the grouping is in no way limiting. In various example embodiments, the I/O components 818 may include output components 826 and input components 828. The output components 826 may include visual components (e.g., a display such as a plasma display panel (PDP), a light emitting diode (LED) display, a liquid crystal display (LCD), a projector, or a cathode ray tube (CRT)), acoustic components (e.g., speakers), haptic components (e.g., a vibratory motor, resistance mechanisms), other signal generators, and so forth. The input components 828 may include alphanumeric input components (e.g., a keyboard, a touch screen configured to receive alphanumeric input, a photo-optical keyboard, or other alphanumeric input components), point based input components (e.g., a mouse, a touchpad, a trackball, a joystick, a motion sensor, or other pointing instrument), tactile input components (e.g., a physical button, a touch screen that provides location and/or force of touches or touch gestures, or other tactile input components), audio input components (e.g., a microphone), and the like.

In further example embodiments, the I/O components 818 may include biometric components 830, motion components 834, environmental components 836, or position components 838 among a wide array of other components. For example, the biometric components 830 may include components to detect expressions (e.g., hand expressions, facial expressions, vocal expressions, body gestures, or eye tracking), measure biosignals (e.g., blood pressure, heart rate, body temperature, perspiration, or brain waves), identify a person (e.g., voice identification, retinal identification, facial identification, fingerprint identification, or electroencephalogram based identification), and the like. The motion components 834 may include acceleration sensor components (e.g., accelerometer), gravitation sensor components, rotation sensor components (e.g., gyroscope), and so forth. The environmental components 836 may include, for example, illumination sensor components (e.g., photometer), temperature sensor components (e.g., one or more thermometer that detect ambient temperature), humidity sensor components, pressure sensor components (e.g., barometer), acoustic sensor components (e.g., one or more microphones that detect background noise), proximity sensor components (e.g., infrared sensors that detect nearby objects), gas sensors (e.g., gas detection sensors to detect concentrations of hazardous gases for safety or to measure pollutants in the atmosphere), or other components that may provide indications, measurements, or signals corresponding to a surrounding physical environment. The position components 838 may include location sensor components (e.g., a GPS receiver component), altitude sensor components (e.g., altimeters or barometers that detect air pressure from which altitude may be derived), orientation sensor components (e.g., magnetometers), and the like.

Communication may be implemented using a wide variety of technologies. The I/O components 818 may include communication components 840 operable to couple the machine 800 to a network 832 or devices 820 via coupling 824 and coupling 822, respectively. For example, the communication components 840 may include a network interface component or other suitable device to interface with the network 832. In further examples, communication components 840 may include wired communication components, wireless communication components, cellular communication components, near field communication (NFC) components, Bluetooth® components (e.g., Bluetooth® Low Energy), Wi-Fi® components, and other communication components to provide communication via other modalities. The devices 820 may be another machine or any of a wide variety of peripheral devices (e.g., a peripheral device coupled via a USB).

Moreover, the communication components 840 may detect identifiers or include components operable to detect identifiers. For example, the communication components 840 may include radio frequency identification (RFID) tag reader components, NFC smart tag detection components, optical reader components (e.g., an optical sensor to detect one-dimensional bar codes such as Universal Product Code (UPC) bar code, multi-dimensional bar codes such as Quick Response (QR) code, Aztec code, Data Matrix, Dataglyph, MaxiCode, PDF417, Ultra Code, UCC RSS-2D bar code, and other optical codes), or acoustic detection components (e.g., microphones to identify tagged audio signals). In addition, a variety of information may be derived via the communication components 840 such as location via Internet Protocol (IP) geo-location, location via Wi-Fi® signal triangulation, location via detecting a NFC beacon signal that may indicate a particular location, and so forth.

Glossary

"CARRIER SIGNAL" in this context refers to any intangible medium that is capable of storing, encoding, or carrying instructions 810 for execution by the machine 800, and includes digital or analog communications signals or other intangible medium to facilitate communication of such instructions 810. Instructions 810 may be transmitted or received over the network 832 using a transmission medium via a network interface device and using any one of a number of well-known transfer protocols.

"CLIENT DEVICE" in this context refers to any machine 800 that interfaces to a communications network 832 to obtain resources from one or more server systems or other client devices 102, 104. A client device 102, 104 may be, but is not limited to, mobile phones, desktop computers, laptops, PDAs, smart phones, tablets, ultra books, netbooks, laptops, multi-processor systems, microprocessor-based or programmable consumer electronics, game consoles, STBs, or any other communication device that a user may use to access a network 832.

"COMMUNICATIONS NETWORK" in this context refers to one or more portions of a network 832 that may be an ad hoc network, an intranet, an extranet, a virtual private network (VPN), a LAN, a wireless LAN (WLAN), a WAN, a wireless WAN (WWAN), a metropolitan area network (MAN), the Internet, a portion of the Internet, a portion of the Public Switched Telephone Network (PSTN), a plain old telephone service (POTS) network, a cellular telephone network, a wireless network, a Wi-Fi® network, another type of network, or a combination of two or more such networks. For example, a network 832 or a portion of a network 832 may include a wireless or cellular network and the coupling may be a Code Division Multiple Access (CDMA) connection, a Global System for Mobile communications (GSM) connection, or other type of cellular or wireless coupling. In this example, the coupling may implement any of a variety of types of data transfer technology, such as Single Carrier Radio Transmission Technology (1xRTT), Evolution-Data Optimized (EVDO) technology, General Packet Radio Service (GPRS) technology, Enhanced Data rates for GSM Evolution (EDGE) technology, third Generation Partnership Project (3GPP) including 3G, fourth generation wireless (4G) networks, Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Worldwide Interoperability for Microwave Access (WiMAX), Long Term Evolution (LTE) standard, others defined by various standard setting organizations, other long range protocols, or other data transfer technology.

"MACHINE-READABLE MEDIUM" in this context refers to a component, device or other tangible media able to store instructions 810 and data temporarily or permanently and may include, but is not be limited to, random-access memory (RAM), read-only memory (ROM), buffer memory, flash memory, optical media, magnetic media, cache memory, other types of storage (e.g., erasable programmable read-only memory (EEPROM)), and/or any suitable combination thereof. The term "machine-readable medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, or associated caches and servers) able to store instructions 810. The term "machine-readable medium" shall also be taken to include any medium, or combination of multiple media, that is capable of storing instructions 810 (e.g., code) for execution by a machine 800, such that the instructions 810, when executed by one or more processors 804 of the machine 800, cause the machine 800 to perform any one or more of the methodologies described herein. Accordingly, a "machine-readable medium" refers to a single storage apparatus or device, as well as "cloud-based" storage systems or storage networks that include multiple storage apparatus or devices. The term "machine-readable medium" excludes signals per se.

"COMPONENT" in this context refers to a device, physical entity, or logic having boundaries defined by function or subroutine calls, branch points, APIs, or other technologies that provide for the partitioning or modularization of particular processing or control functions. Components may be combined via their interfaces with other components to carry out a machine process. A component may be a packaged functional hardware unit designed for use with other components and a part of a program that usually performs a particular function of related functions. Components may constitute either software components (e.g., code embodied on a machine-readable medium) or hardware components. A "hardware component" is a tangible unit capable of performing certain operations and may be configured or arranged in a certain physical manner. In various example embodiments, one or more computer systems (e.g., a standalone computer system, a client computer system, or a server computer system) or one or more hardware components of a computer system (e.g., a processor or a group of processors 804) may be configured by software (e.g., an application 716 or application portion) as a hardware component that operates to perform certain operations as described herein. A hardware component may also be implemented mechanically, electronically, or any suitable combination thereof. For example, a hardware component may include dedicated circuitry or logic that is permanently configured to perform certain operations. A hardware component may be a special-purpose processor, such as a field-programmable gate array (FPGA) or an application specific integrated circuit (ASIC). A hardware component may also include programmable logic or circuitry that is temporarily configured by software to perform certain operations. For example, a hardware component may include software executed by a general-purpose processor 804 or other programmable processor 804. Once configured by such software, hardware components become specific machines 800 (or specific components of a machine 800) uniquely tailored to perform the configured functions and are no longer general-purpose processors 804. It will be appreciated that the decision to implement a hardware component mechanically, in dedicated and permanently configured circuitry, or in temporarily configured circuitry (e.g., configured by software), may be driven by cost and time considerations. Accordingly, the phrase "hardware component"(or "hardware-implemented component") should be understood to encompass a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner or to perform certain operations described herein. Considering embodiments in which hardware components are temporarily configured (e.g., programmed), each of the hardware components need not be configured or instantiated at any one instance in time. For example, where a hardware component comprises a general-purpose processor 804 configured by software to become a special-purpose processor, the general-purpose processor 804 may be configured as respectively different special-purpose processors (e.g., comprising different hardware components) at different times. Software accordingly configures a particular processor or processors 804, for example, to constitute a particular hardware component at one instance of time and to constitute a different hardware component at a different instance of time. Hardware components can provide information to, and receive information from, other hardware components. Accordingly, the described hardware components may be regarded as being communicatively coupled. Where multiple hardware components exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses 802) between or among two or more of the hardware components. In embodiments in which multiple hardware components are configured or instantiated at different times, communications between such hardware components may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware components have access. For example, one hardware component may perform an operation and store the output of that operation in a memory device to which it is communicatively coupled. A further hardware component may then, at a later time, access the memory device to retrieve and process the stored output. Hardware components may also initiate communications with input or output devices, and can operate on a resource (e.g., a collection of information). The various operations of example methods described herein may be performed, at least partially, by one or more processors 804 that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors 804 may constitute processor-implemented components that operate to perform one or more operations or functions described herein. As used herein, "processor-implemented component" refers to a hardware component implemented using one or more processors 804. Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors 804 being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors 804 or processor-implemented components. Moreover, the one or more processors 804 may also operate to support performance of the relevant operations in a "cloud computing" environment or as a "software as a service" (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines 800 including processors 804), with these operations being accessible via a network 832 (e.g., the Internet) and via one or more appropriate interfaces (e.g., an API). The performance of certain of the operations may be distributed among the processors 804, not only residing within a single machine 800, but deployed across a number of machines 800. In some example embodiments, the processors 804 or processor-implemented components may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors 804 or processor-implemented components may be distributed across a number of geographic locations.

"PROCESSOR" in this context refers to any circuit or virtual circuit (a physical circuit emulated by logic executing on an actual processor 804) that manipulates data values according to control signals (e.g., "commands," "op codes," "machine code," etc.) and which produces corresponding output signals that are applied to operate a machine 800. A processor 804 may be, for example, a central processing unit (CPU), a reduced instruction set computing (RISC) processor, a complex instruction set computing (CISC) processor, a graphics processing unit (GPU), a digital signal processor (DSP), an ASIC, a radio-frequency integrated circuit (RFIC) or any combination thereof. A processor 804 may further be a multi-core processor having two or more independent processors 804 (sometimes referred to as "cores") that may execute instructions 810 contemporaneously.

CLAIMS

1. A computer-implemented method of a cloud server, the method comprising:

receiving, from a first client:

a communication session identifier representing a communication session including the first client and a second client,

a first text content corresponding to first speech from the first client in the communication session, and

a first time-stamp value identifying a first time of the first speech of the first client;

receiving, from a second client:

the communication session identifier,

a second text content corresponding to second speech from the second client in the communication session, and

a second time-stamp value identifying a different second time of the second speech of the second client; and

based on the first time-stamp value and the second time-stamp value, aggregating the first text content and the second text content into a transcript of the communication session.

2. The method of claim 1, further comprising:

synchronizing an internal clock of the first client with an internal clock of the second client, the first time-stamp value being determined based on the internal clock of the first client and the second time-stamp value being determined based on the internal clock of the second client.

3. The method of claim 1, further comprising:

receiving, from a third client in the communication session:

the communication session identifier,

a third text content corresponding to third speech from the third client in the communication session, and

a third time-stamp value identifying a different third time of the third speech of the third client.

4. The method of claim 3, wherein aggregating the first text content and the second text content into the transcript of the communication session further comprises aggregating the third text content based on the third time-stamp value.

5. The method of claim 1, wherein the first text content comprises text output of a natural language processing service.

6. The method of claim 1, wherein the text content includes characters in a first alphabet and the second text content includes different characters in a second alphabet.

7. The method of claim 1, further comprising:

establishing a first connection with the first client and a second connection with the second client, the first text content received via the first connection and the second text content received via the second connection.

8. A cloud server comprising:

one or more computer processors; and

one or more computer-readable mediums storing instructions that, when executed by the one or more computer processors, cause the cloud server to perform operations comprising:

receiving, from a first client:

a communication session identifier representing a communication session including the first client and a second client,

a first text content corresponding to first speech from the first client in the communication session, and

a first time-stamp value identifying a first time of the first speech of the first client;

receiving, from a second client:

the communication session identifier,

a second text content corresponding to second speech from the second client in the communication session, and

a second time-stamp value identifying a different second time of the second speech of the second client; and

based on the first time-stamp value and the second time-stamp value, aggregating the first text content and the second text content into a transcript of the communication session.

9. The cloud server of claim 8, the operations further comprising:

synchronizing an internal clock of the first client with an internal clock of the second client, the first time-stamp value being determined based on the internal clock of the first client and the second time-stamp value being determined based on the internal clock of the second client.

10. The cloud server of claim 8, the operations further comprising:

receiving, from a third client in the communication session:

the communication session identifier,

a third text content corresponding to third speech from the third client in the communication session, and

a third time-stamp value identifying a different third time of the third speech of the third client.

11. The cloud server of claim 10, wherein aggregating the first text content and the second text content into the transcript of the communication session further comprises aggregating the third text content based on the third time-stamp value.

12. The cloud server of claim 8, wherein the first text content comprises text output of a natural language processing service.

13. The cloud server of claim 8, wherein the text content includes characters in a first alphabet and the second text content includes different characters in a second alphabet.

14. The cloud server of claim 8, the operations further comprising:

establishing a first connection with the first client and a second connection with the second client, the first text content received via the first connection and the second text content received via the second connection.

15. A computer-readable medium storing instructions that, when executed by one or more computer processors of a cloud server, cause the cloud server to perform operations comprising:

receiving, from a first client:

a communication session identifier representing a communication session including the first client and a second client,

a first text content corresponding to first speech from the first client in the communication session, and

a first time-stamp value identifying a first time of the first speech of the first client;

receiving, from a second client:

the communication session identifier,

a second text content corresponding to second speech from the second client in the communication session, and

a second time-stamp value identifying a different second time of the second speech of the second client; and

based on the first time-stamp value and the second time-stamp value, aggregating the first text content and the second text content into a transcript of the communication session.

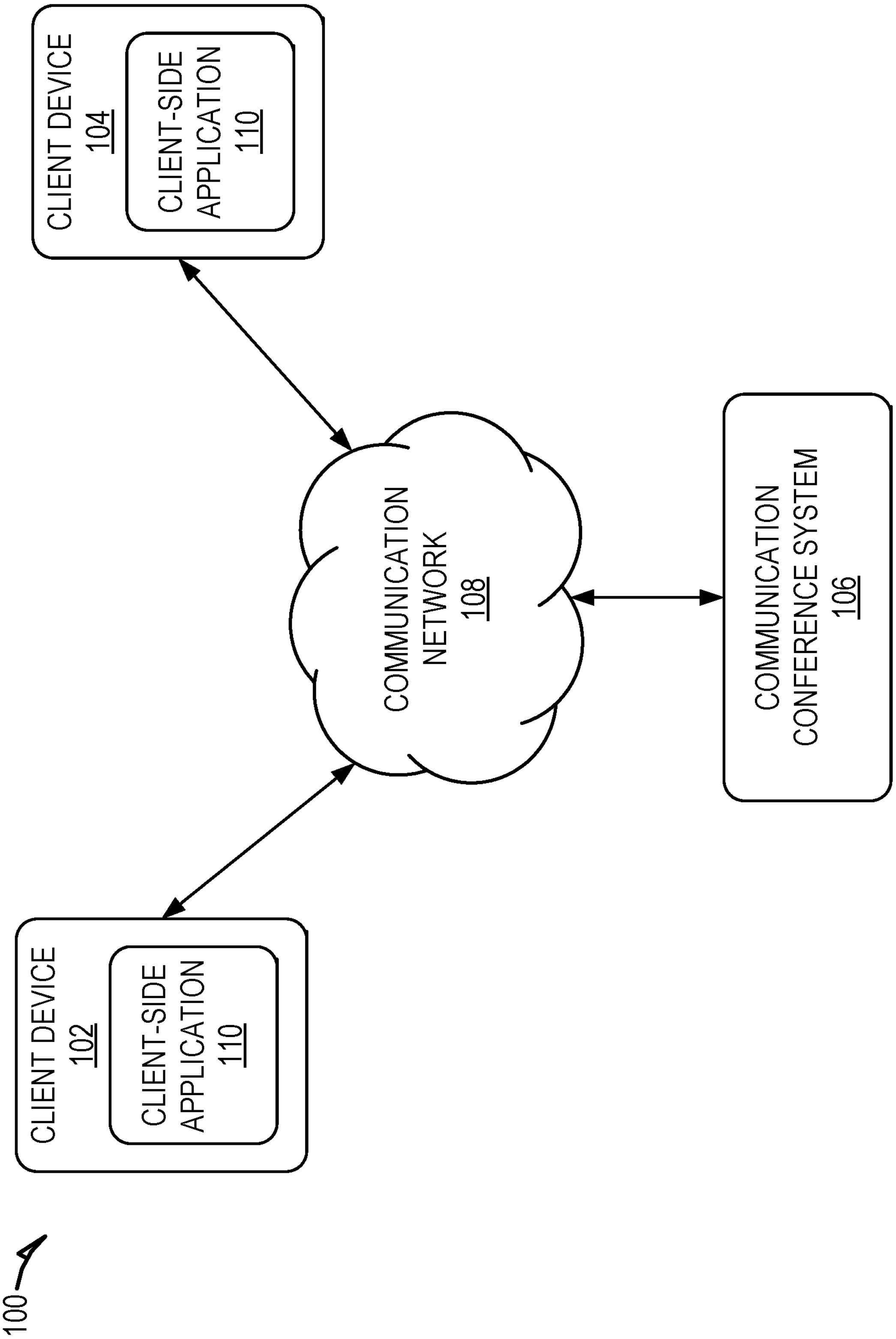
CLIENT DEVICE 104
CLIENT-SIDE APPLICATION 110
COMMUNICATION NETWORK 108
COMMUNICATION CONFERENCE SYSTEM 106
CLIENT DEVICE 102
CLIENT-SIDE APPLICATION 110
100

FIG. 1

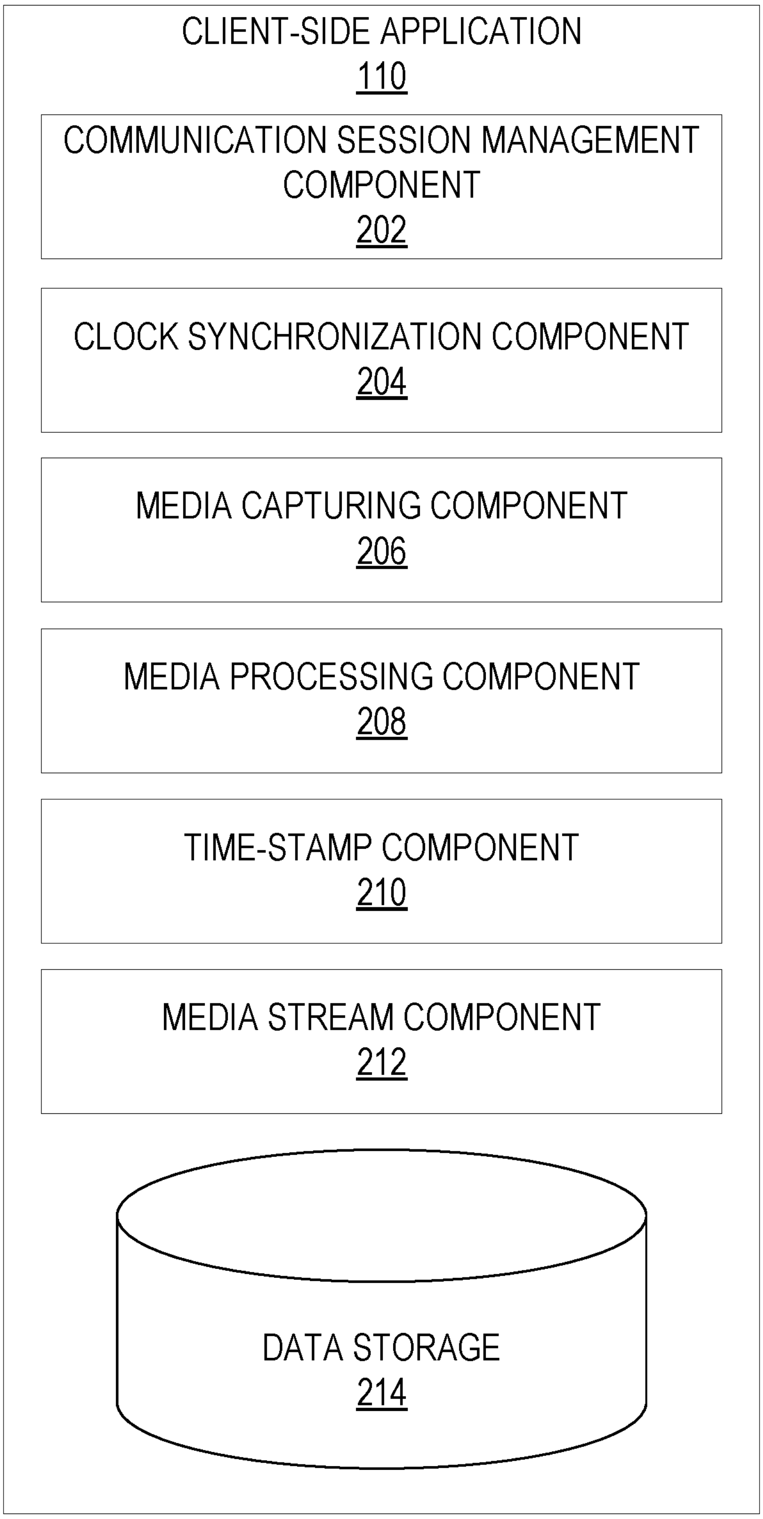
CLIENT-SIDE APPLICATION
110
COMMUNICATION SESSION MANAGEMENT COMPONENT
202
CLOCK SYNCHRONIZATION COMPONENT
204
MEDIA CAPTURING COMPONENT
206
MEDIA PROCESSING COMPONENT
208
TIME-STAMP COMPONENT
210
MEDIA STREAM COMPONENT
212
DATA STORAGE
214

FIG. 2

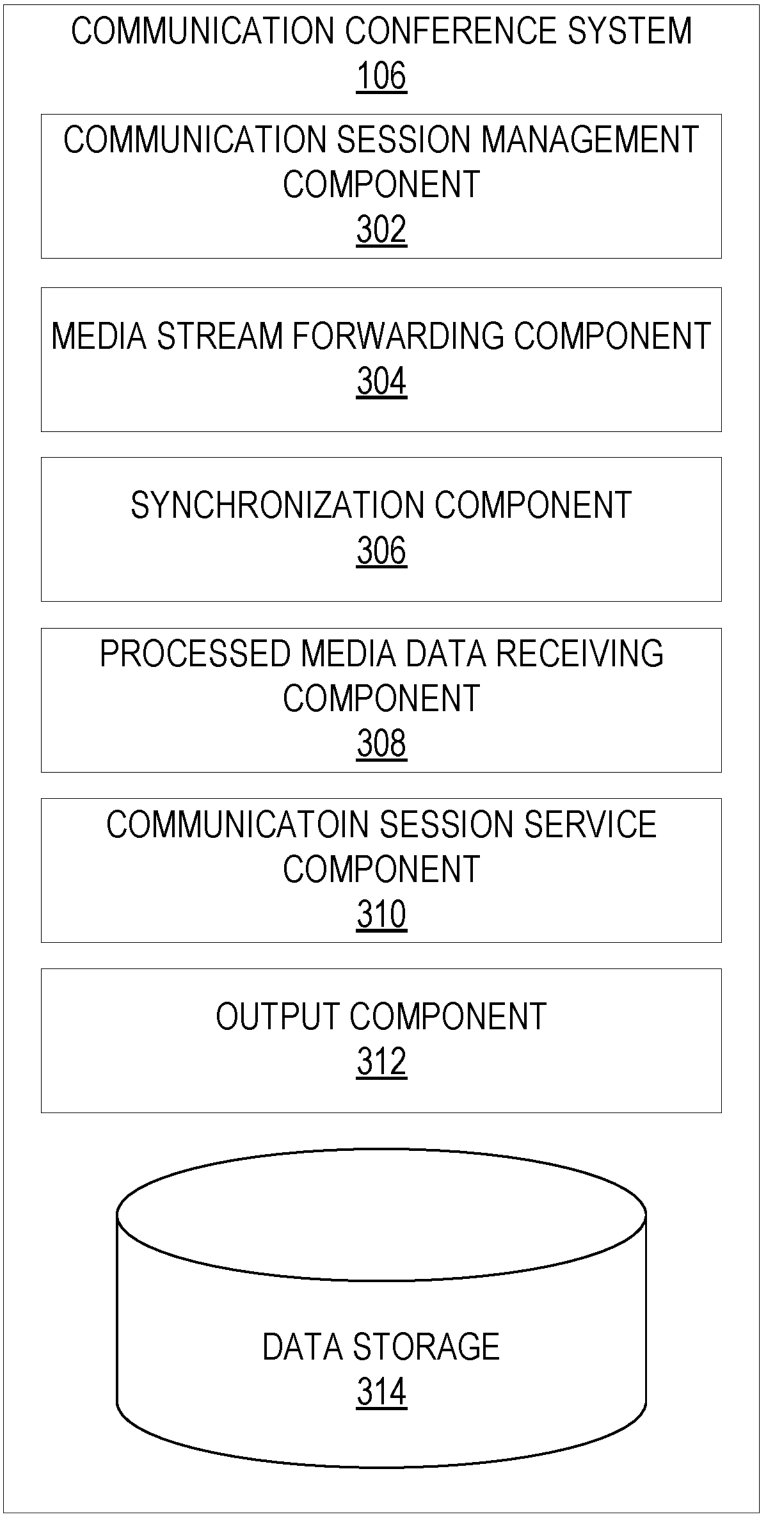
COMMUNICATION CONFERENCE SYSTEM
106
COMMUNICATION SESSION MANAGEMENT COMPONENT
302
MEDIA STREAM FORWARDING COMPONENT
304
SYNCHRONIZATION COMPONENT
306
PROCESSED MEDIA DATA RECEIVING COMPONENT
308
COMMUNICATOIN SESSION SERVICE COMPONENT
310
OUTPUT COMPONENT
312
DATA STORAGE
314

FIG. 3

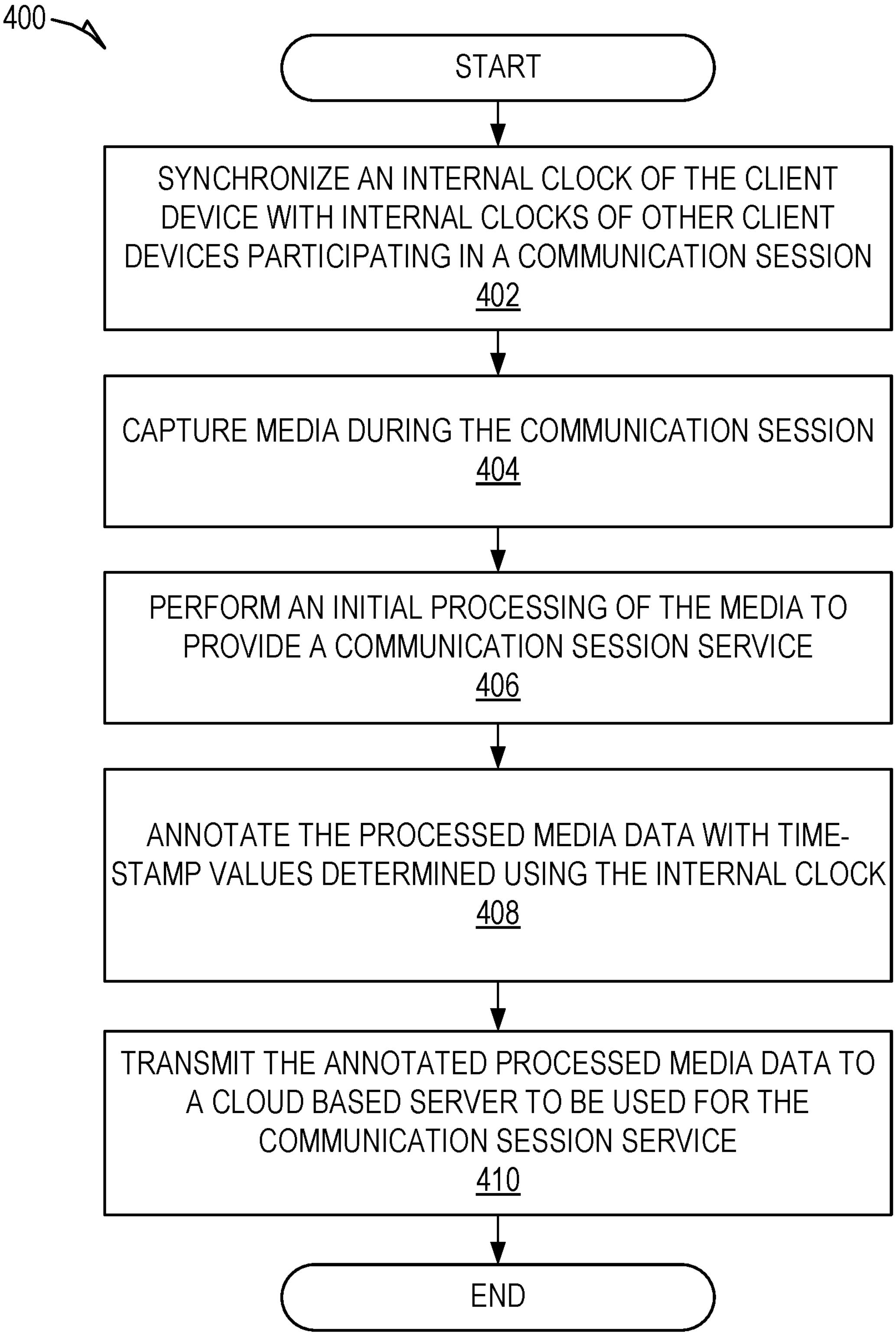
400
START
SYNCHRONIZE AN INTERNAL CLOCK OF THE CLIENT DEVICE WITH INTERNAL CLOCKS OF OTHER CLIENT DEVICES PARTICIPATING IN A COMMUNICATION SESSION
402
CAPTURE MEDIA DURING THE COMMUNICATION SESSION
404
PERFORM AN INITIAL PROCESSING OF THE MEDIA TO PROVIDE A COMMUNICATION SESSION SERVICE
406
ANNOTATE THE PROCESSED MEDIA DATA WITH TIME-STAMP VALUES DETERMINED USING THE INTERNAL CLOCK
408
TRANSMIT THE ANNOTATED PROCESSED MEDIA DATA TO A CLOUD BASED SERVER TO BE USED FOR THE COMMUNICATION SESSION SERVICE
410
END

FIG. 4

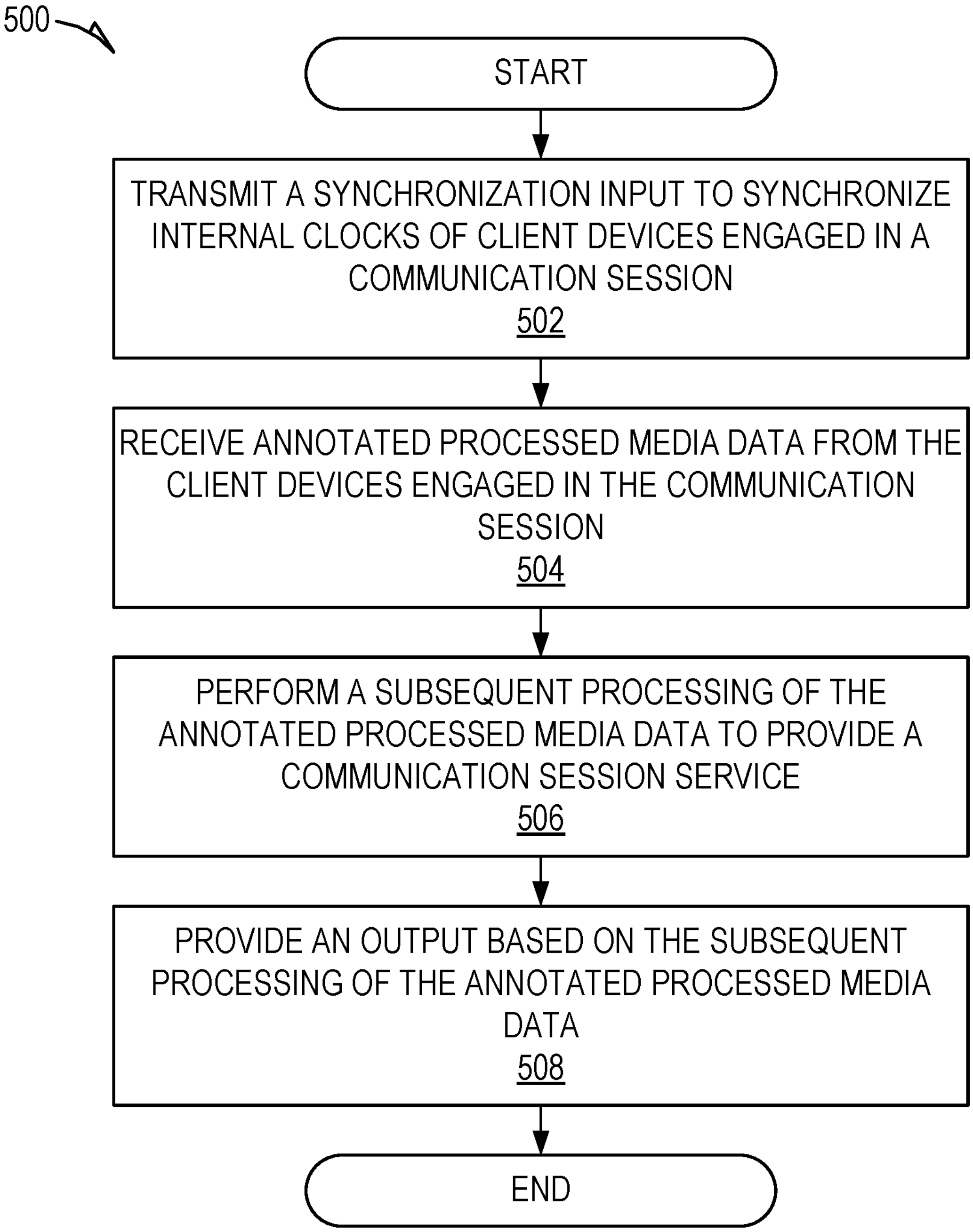
500
START
TRANSMIT A SYNCHRONIZATION INPUT TO SYNCHRONIZE INTERNAL CLOCKS OF CLIENT DEVICES ENGAGED IN A COMMUNICATION SESSION
502
RECEIVE ANNOTATED PROCESSED MEDIA DATA FROM THE CLIENT DEVICES ENGAGED IN THE COMMUNICATION SESSION
504
PERFORM A SUBSEQUENT PROCESSING OF THE ANNOTATED PROCESSED MEDIA DATA TO PROVIDE A COMMUNICATION SESSION SERVICE
506
PROVIDE AN OUTPUT BASED ON THE SUBSEQUENT PROCESSING OF THE ANNOTATED PROCESSED MEDIA DATA
508
END

FIG. 5

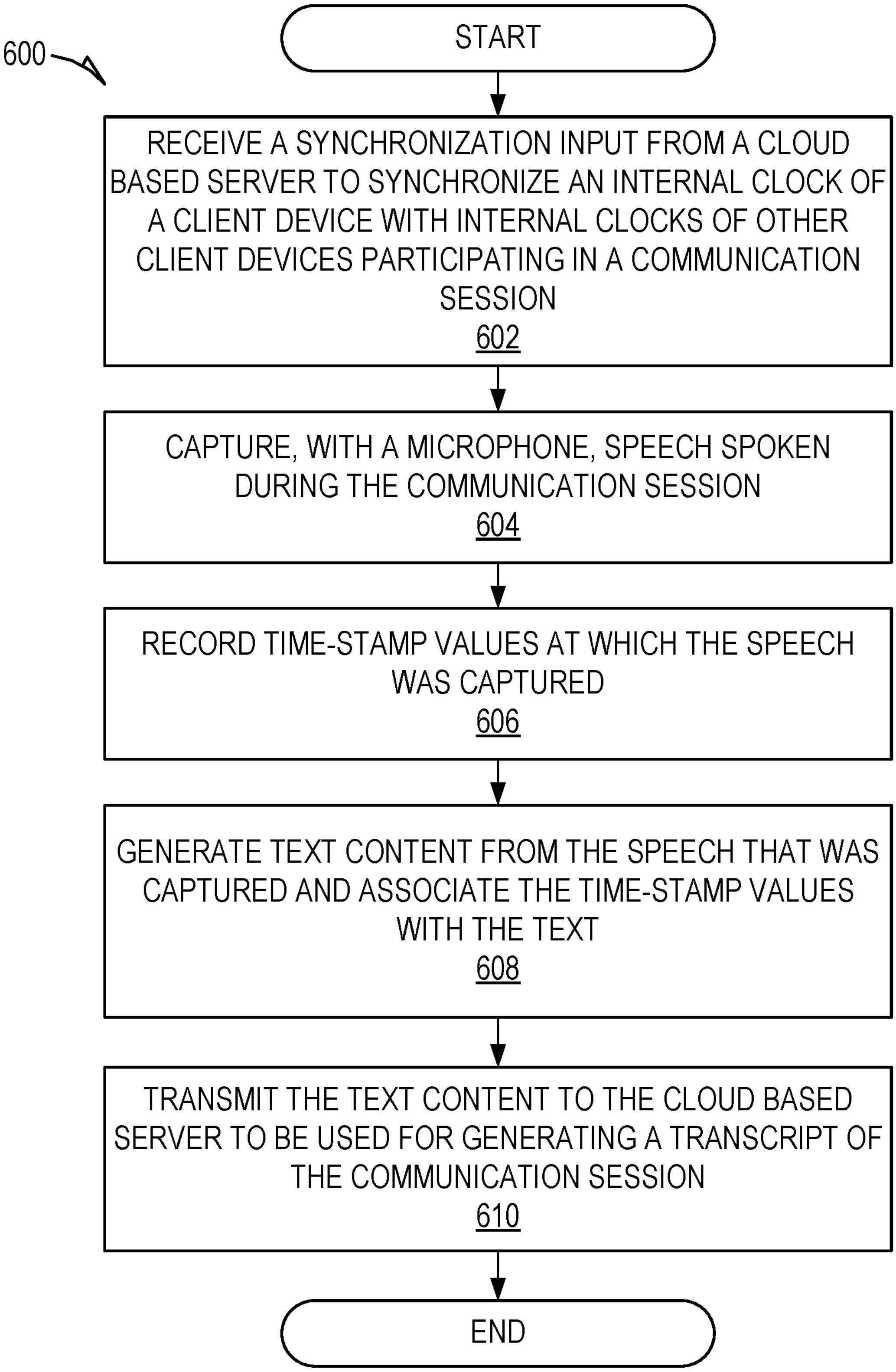
600
START
RECEIVE A SYNCHRONIZATION INPUT FROM A CLOUD BASED SERVER TO SYNCHRONIZE AN INTERNAL CLOCK OF A CLIENT DEVICE WITH INTERNAL CLOCKS OF OTHER CLIENT DEVICES PARTICIPATING IN A COMMUNICATION SESSION
602
CAPTURE, WITH A MICROPHONE, SPEECH SPOKEN DURING THE COMMUNICATION SESSION
604
RECORD TIME-STAMP VALUES AT WHICH THE SPEECH WAS CAPTURED
606
GENERATE TEXT CONTENT FROM THE SPEECH THAT WAS CAPTURED AND ASSOCIATE THE TIME-STAMP VALUES WITH THE TEXT
608
TRANSMIT THE TEXT CONTENT TO THE CLOUD BASED SERVER TO BE USED FOR GENERATING A TRANSCRIPT OF THE COMMUNICATION SESSION
610
END

FIG. 6A

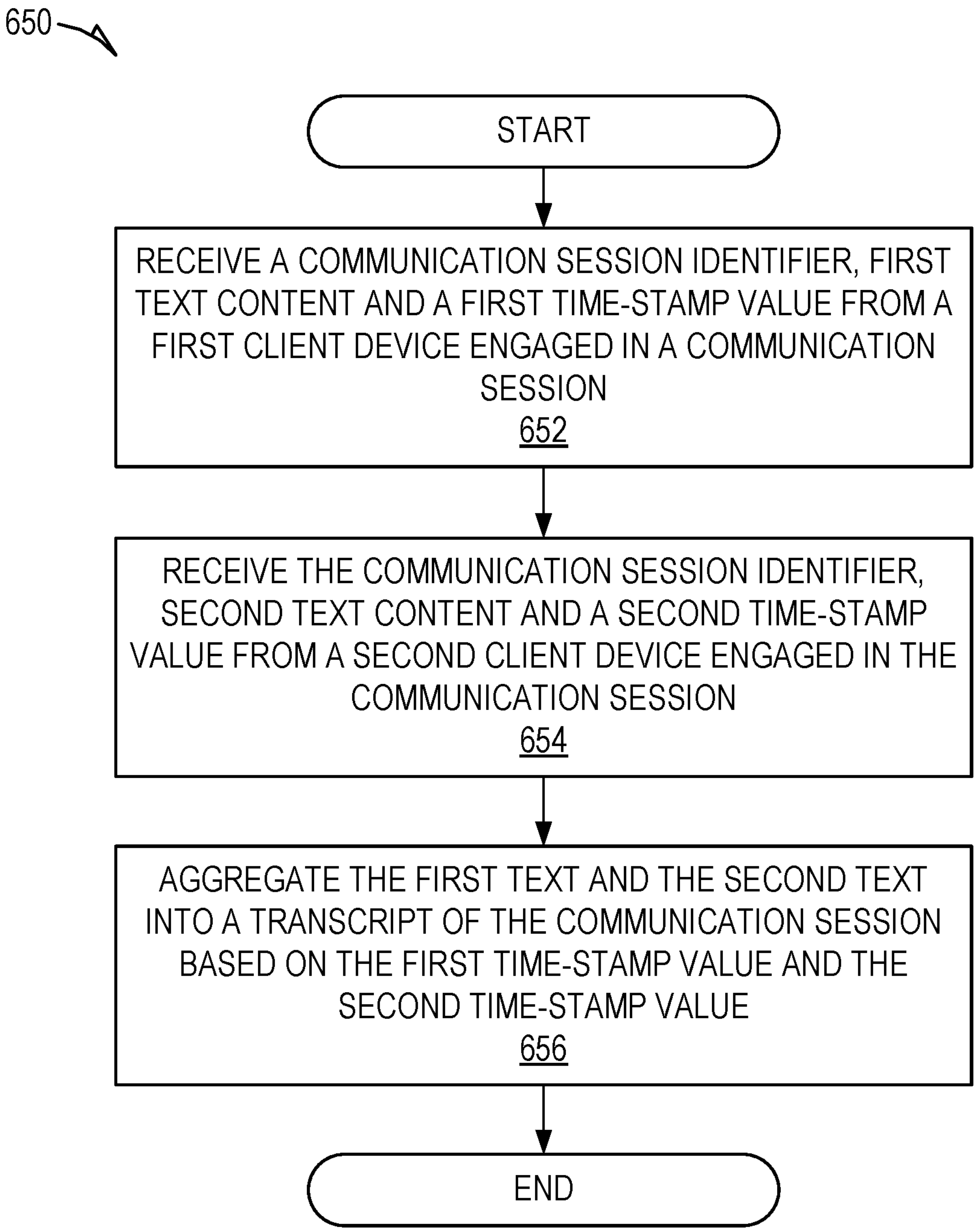
650
START
RECEIVE A COMMUNICATION SESSION IDENTIFIER, FIRST TEXT CONTENT AND A FIRST TIME-STAMP VALUE FROM A FIRST CLIENT DEVICE ENGAGED IN A COMMUNICATION SESSION
652
RECEIVE THE COMMUNICATION SESSION IDENTIFIER, SECOND TEXT CONTENT AND A SECOND TIME-STAMP VALUE FROM A SECOND CLIENT DEVICE ENGAGED IN THE COMMUNICATION SESSION
654
AGGREGATE THE FIRST TEXT AND THE SECOND TEXT INTO A TRANSCRIPT OF THE COMMUNICATION SESSION BASED ON THE FIRST TIME-STAMP VALUE AND THE SECOND TIME-STAMP VALUE
656
END

FIG. 6B

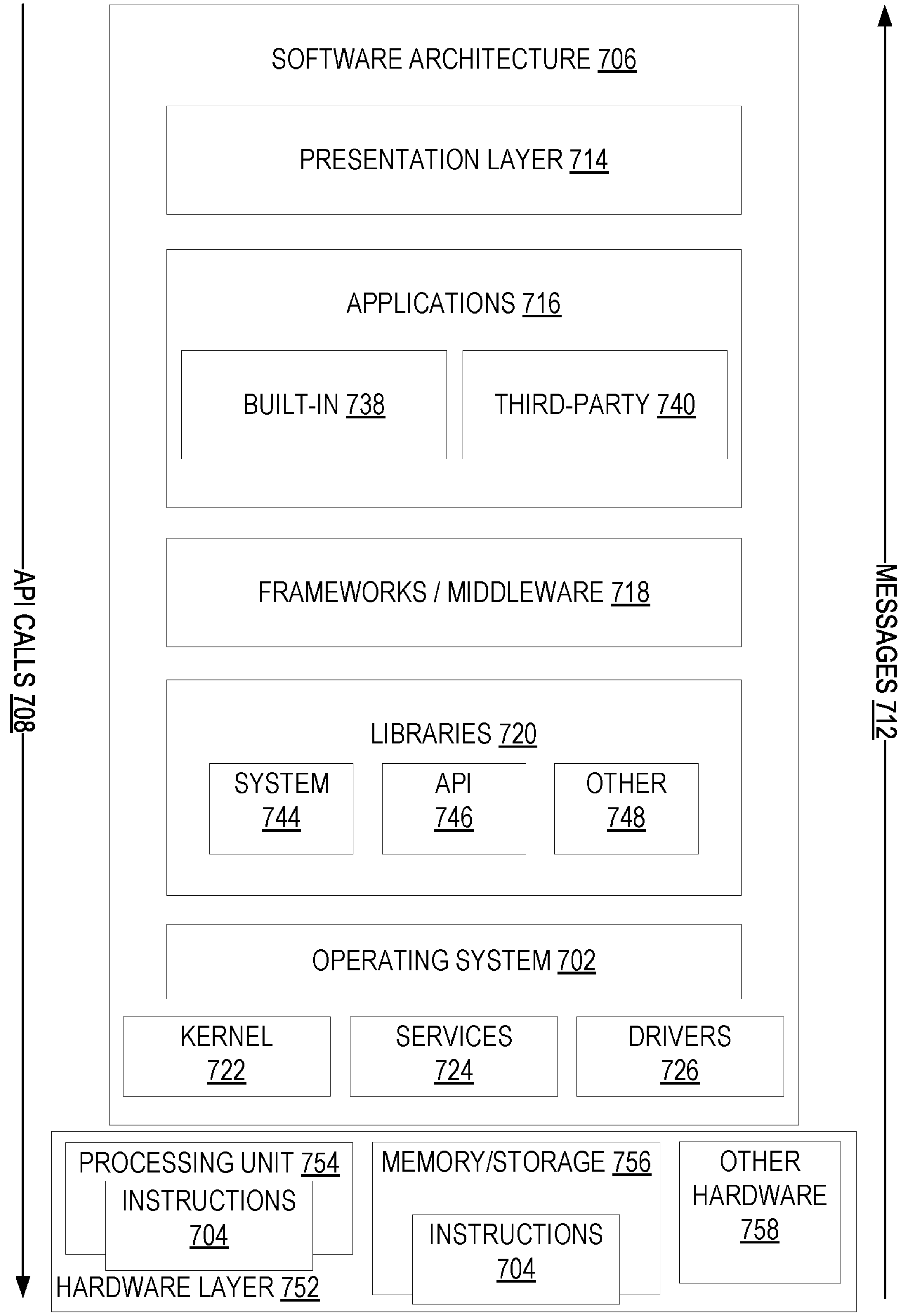
SOFTWARE ARCHITECTURE 706
PRESENTATION LAYER 714
APPLICATIONS 716
BUILT-IN 738
THIRD-PARTY 740
FRAMEWORKS / MIDDLEWARE 718
LIBRARIES 720
SYSTEM 744
API 746
OTHER 748
OPERATING SYSTEM 702
KERNEL 722
SERVICES 724
DRIVERS 726
PROCESSING UNIT 754
INSTRUCTIONS 704
MEMORY/STORAGE 756
INSTRUCTIONS 704
OTHER HARDWARE 758
HARDWARE LAYER 752
API CALLS 708
MESSAGES 712

FIG. 7

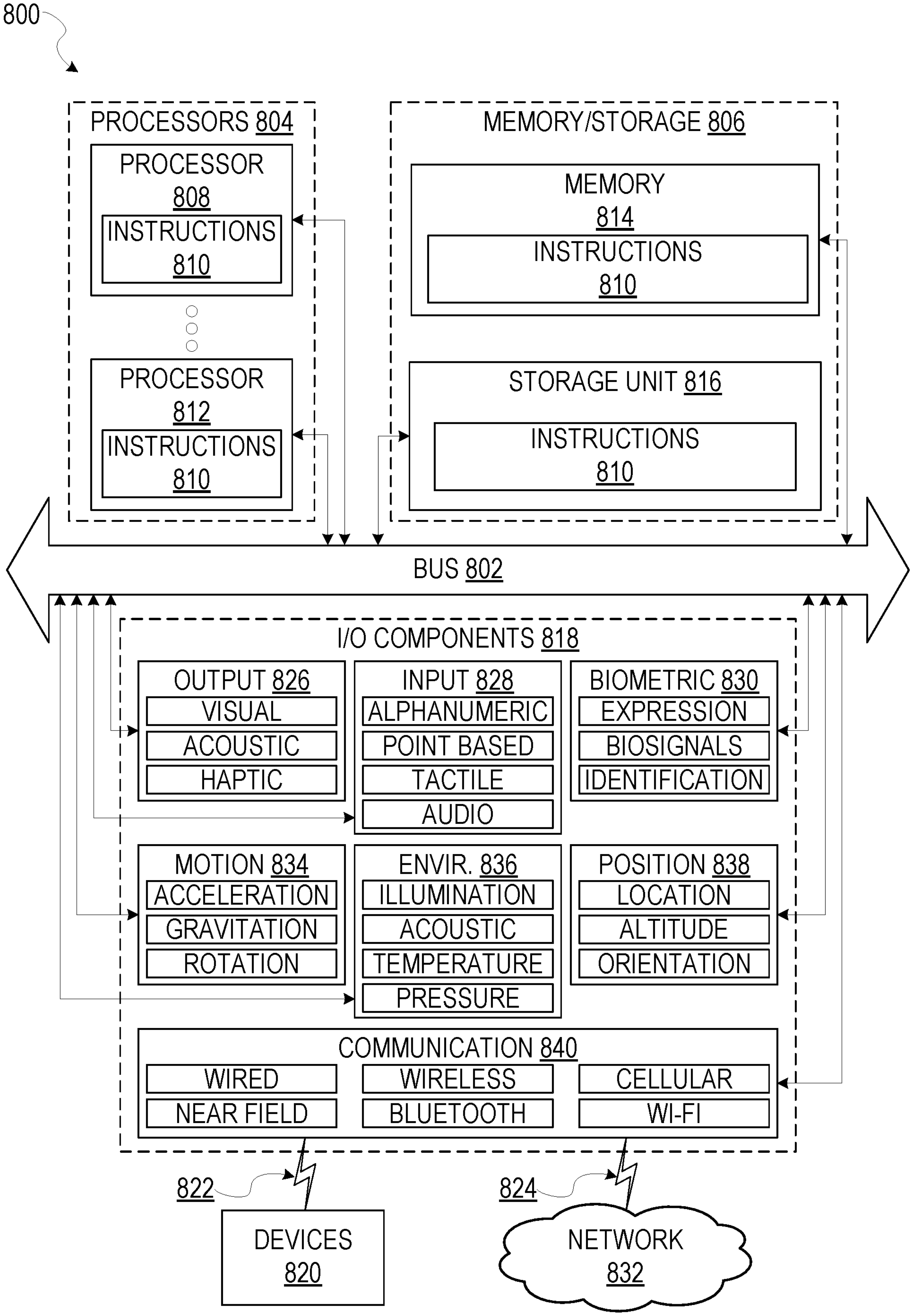
800
PROCESSORS 804
PROCESSOR 808
INSTRUCTIONS 810
PROCESSOR 812
INSTRUCTIONS 810
MEMORY/STORAGE 806
MEMORY 814
INSTRUCTIONS 810
STORAGE UNIT 816
INSTRUCTIONS 810
BUS 802
I/O COMPONENTS 818
OUTPUT 826
VISUAL
ACOUSTIC
HAPTIC
INPUT 828
ALPHANUMERIC
POINT BASED
TACTILE
AUDIO
BIOMETRIC 830
EXPRESSION
BIOSIGNALS
IDENTIFICATION
MOTION 834
ACCELERATION
GRAVITATION
ROTATION
ENVIR. 836
ILLUMINATION
ACOUSTIC
TEMPERATURE
PRESSURE
POSITION 838
LOCATION
ALTITUDE
ORIENTATION
COMMUNICATION 840
WIRED
WIRELESS
CELLULAR
NEAR FIELD
BLUETOOTH
WI-FI
822
824
DEVICES 820
NETWORK 832

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2022/029087

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L65/10
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L G06F G10L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2019/341050 A1 (DIAMANT ADI [IL] ET AL) 7 November 2019 (2019-11-07)	1,3-5,7,8,10-12,14,15
Y	paragraphs [0064] - [0082], [0115] - [0116], [0128] - [0129], [0139], [0153] - [0159]; figures 1,22 -----	2,6,9,13
Y	US 2020/349230 A1 (YOSHIOKA TAKUYA [US] ET AL) 5 November 2020 (2020-11-05) paragraphs [0068] - [0069]; figure 9 -----	2,9
Y	US 2019/108492 A1 (NELSON STEVEN [US] ET AL) 11 April 2019 (2019-04-11) paragraphs [0208] - [0334] -----	6,13
A	US 2014/082096 A1 (SCHERPA JOSEF A [US] ET AL) 20 March 2014 (2014-03-20) paragraphs [0045] - [0080] -----	5,12

□ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 August 2022	31/08/2022

Name and mailing address of the ISA/
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Valero, Monica

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2022/029087

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2019341050 A1	07-11-2019	CN 112075075 A	11-12-2020
		EP 3788780 A1	10-03-2021
		US 2019341050 A1	07-11-2019
		US 2021210097 A1	08-07-2021
		WO 2019212920 A1	07-11-2019
US 2020349230 A1	05-11-2020	AU 2020265509 A1	07-10-2021
		CA 3132837 A1	05-11-2020
		CN 113874936 A	31-12-2021
		EP 3963574 A1	09-03-2022
		IL 287494 A	01-12-2021
		JP 2022532313 A	14-07-2022
		KR 20220002326 A	06-01-2022
		SG 11202109373T A	29-09-2021
		US 2020349230 A1	05-11-2020
		WO 2020222925 A1	05-11-2020
US 2019108492 A1	11-04-2019	US 2019108492 A1	11-04-2019
		US 2021272071 A1	02-09-2021
US 2014082096 A1	20-03-2014	US 2014082096 A1	20-03-2014
		US 2014082106 A1	20-03-2014
		US 2016197860 A1	07-07-2016
		US 2016197861 A1	07-07-2016
		US 2018295077 A1	11-10-2018