



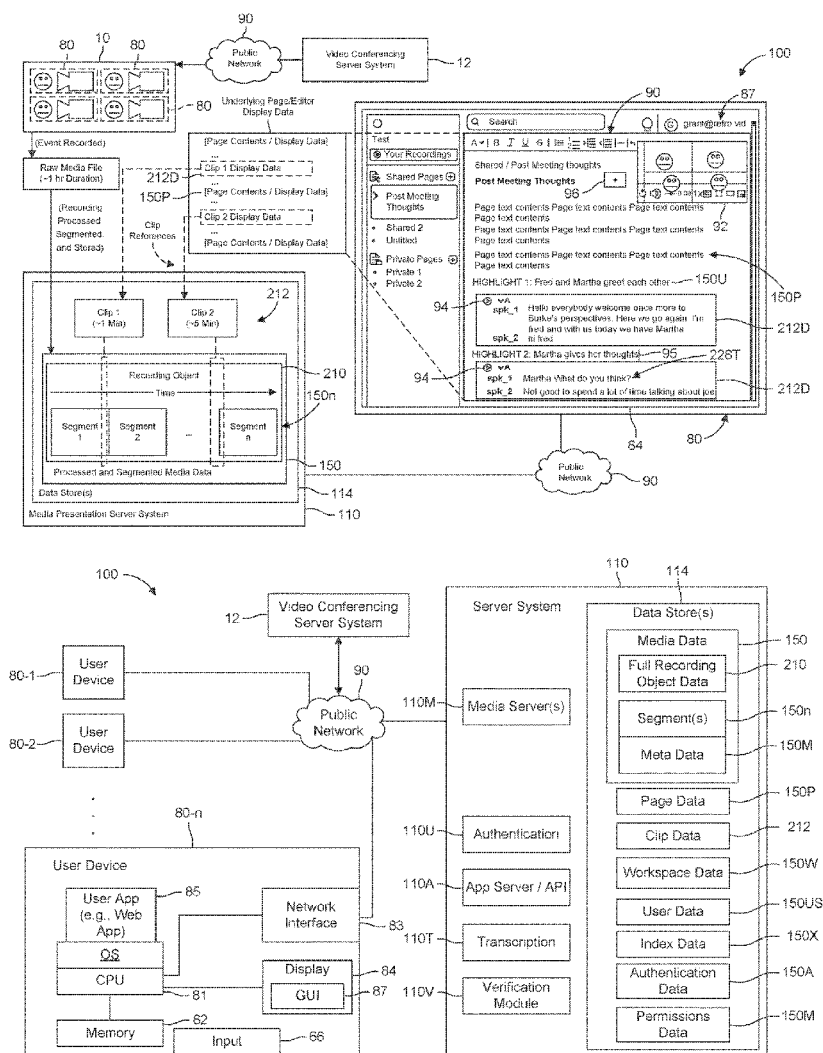
US 20230154497A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2023/0154497 A1**
Malan et al. (43) **Pub. Date: May 18, 2023**(54) **SYSTEM AND METHOD FOR ACCESS CONTROL, GROUP OWNERSHIP, AND REDACTION OF RECORDINGS OF EVENTS****Related U.S. Application Data**

(60) Provisional application No. 63/280,837, filed on Nov. 18, 2021, provisional application No. 63/280,830, filed on Nov. 18, 2021.

Publication Classification(51) **Int. Cl.**
G11B 27/00 (2006.01)
(52) **U.S. Cl.**
CPC **G11B 27/002** (2013.01)(57) **ABSTRACT**

A media presentation system includes a page editor enabling embedding into pages of clips referencing portions of a full time-indexed media recording. The pages can be shared with other users with different permissions levels. When a page is loaded, text-layers (e.g., transcript text) of the media data referenced by any embedded clips are displayed, and a playback token is provided, granting access only to the referenced portions of the recording. For full recordings, a group ownership scheme dictates that any recording can have one or many owners that have full rights in accessing and modifying the recording, including redaction.

(21) Appl. No.: **18/056,978**(22) Filed: **Nov. 18, 2022**

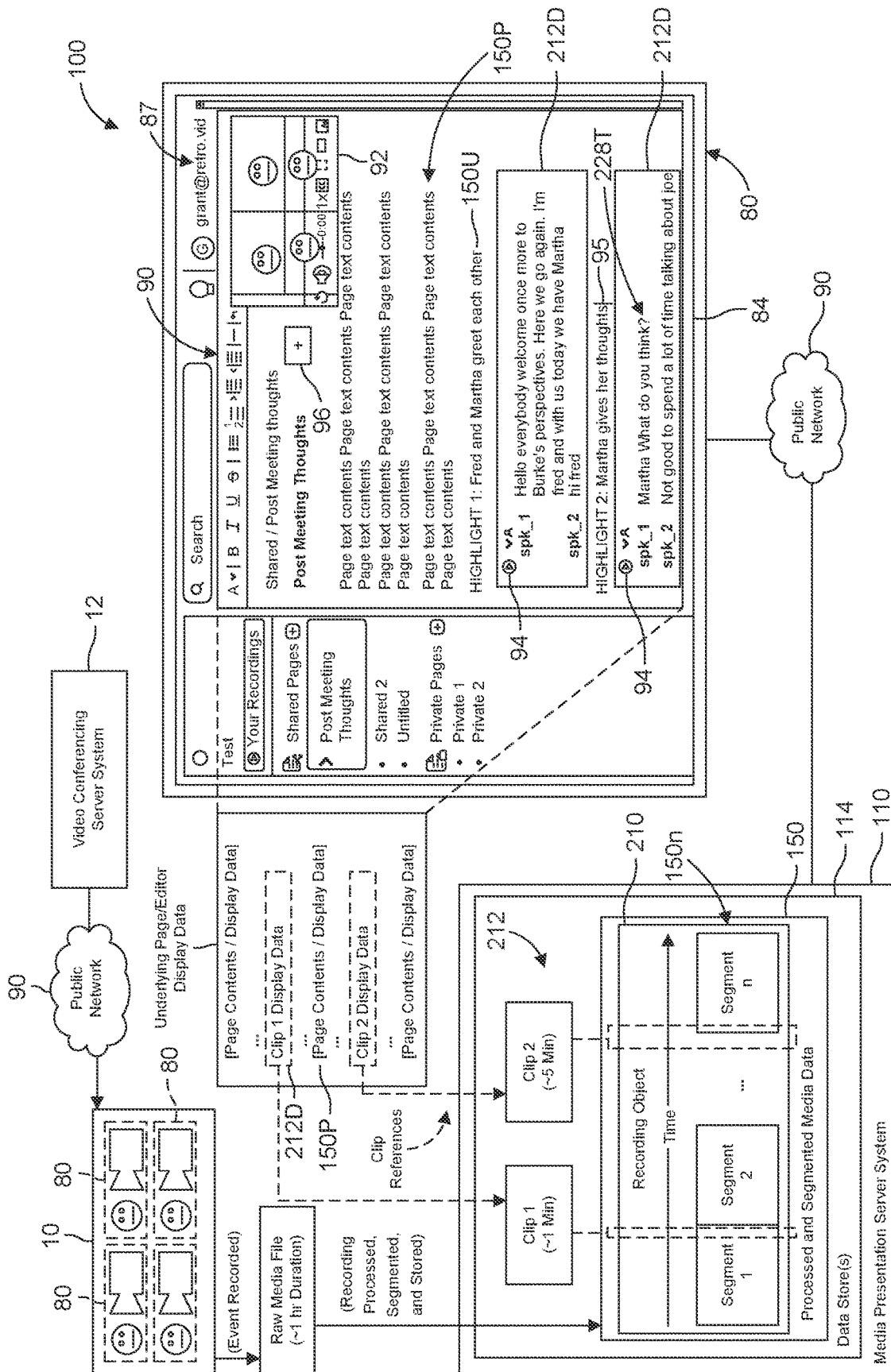


FIG. 1A

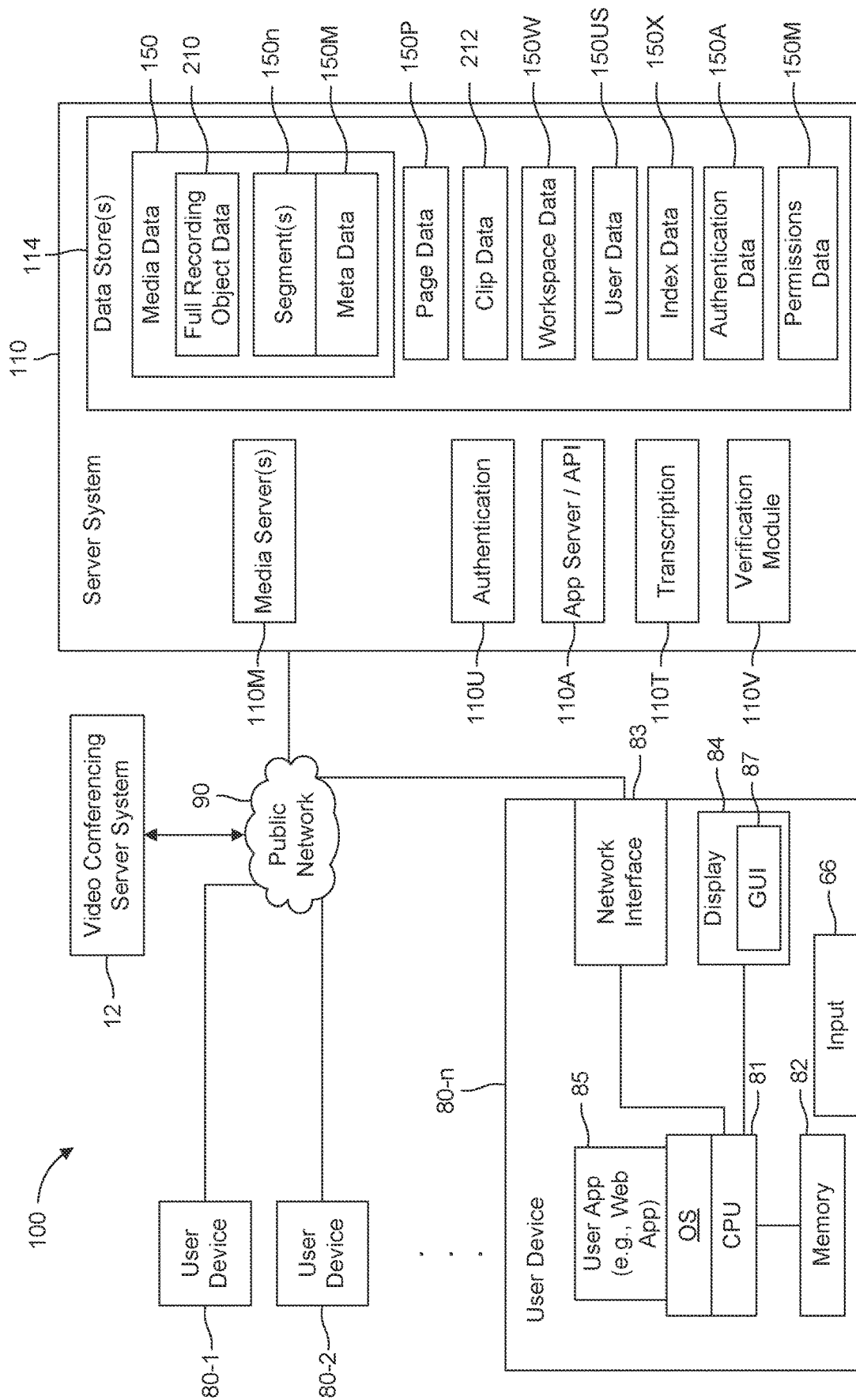


FIG. 1B

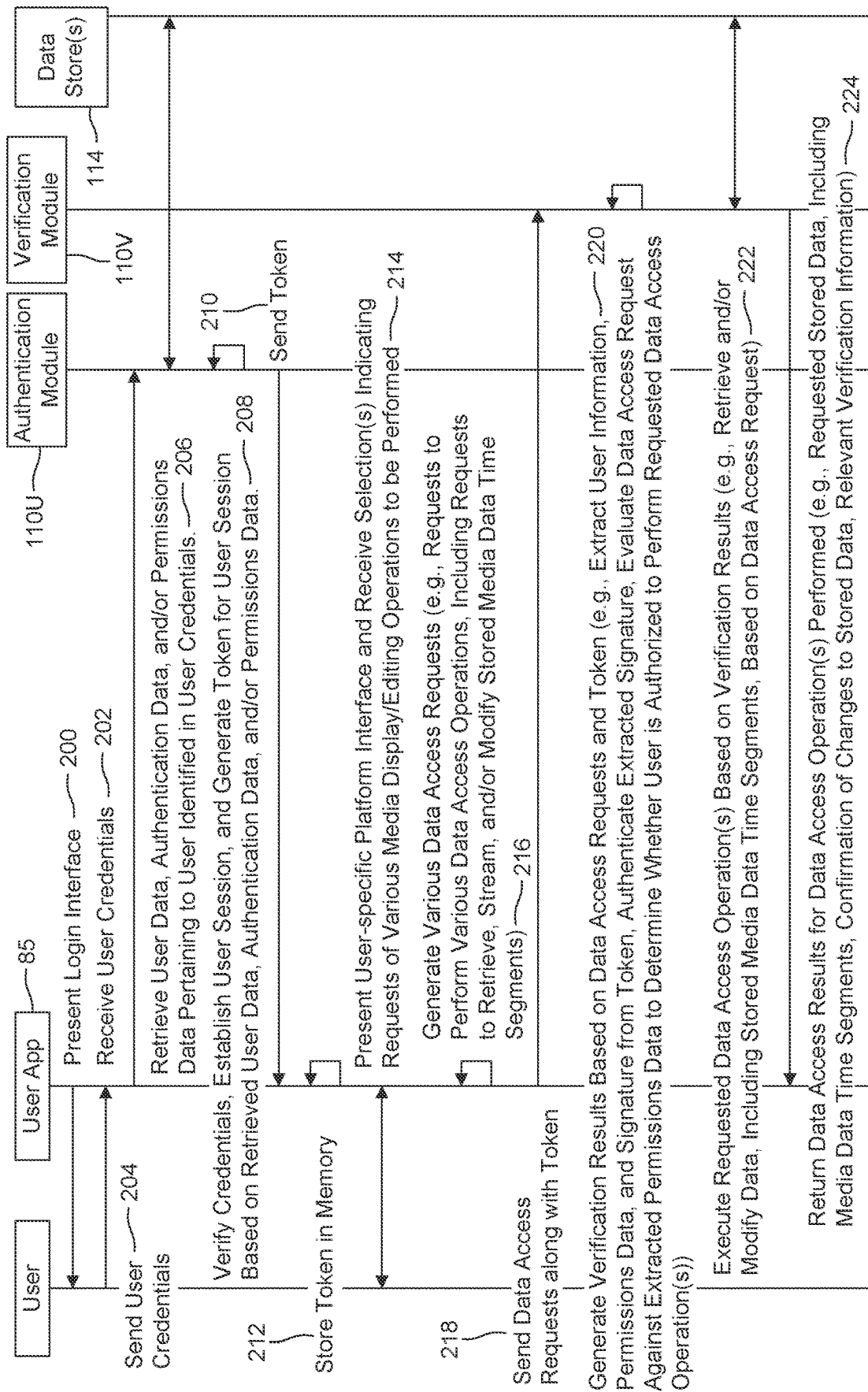


FIG. 2

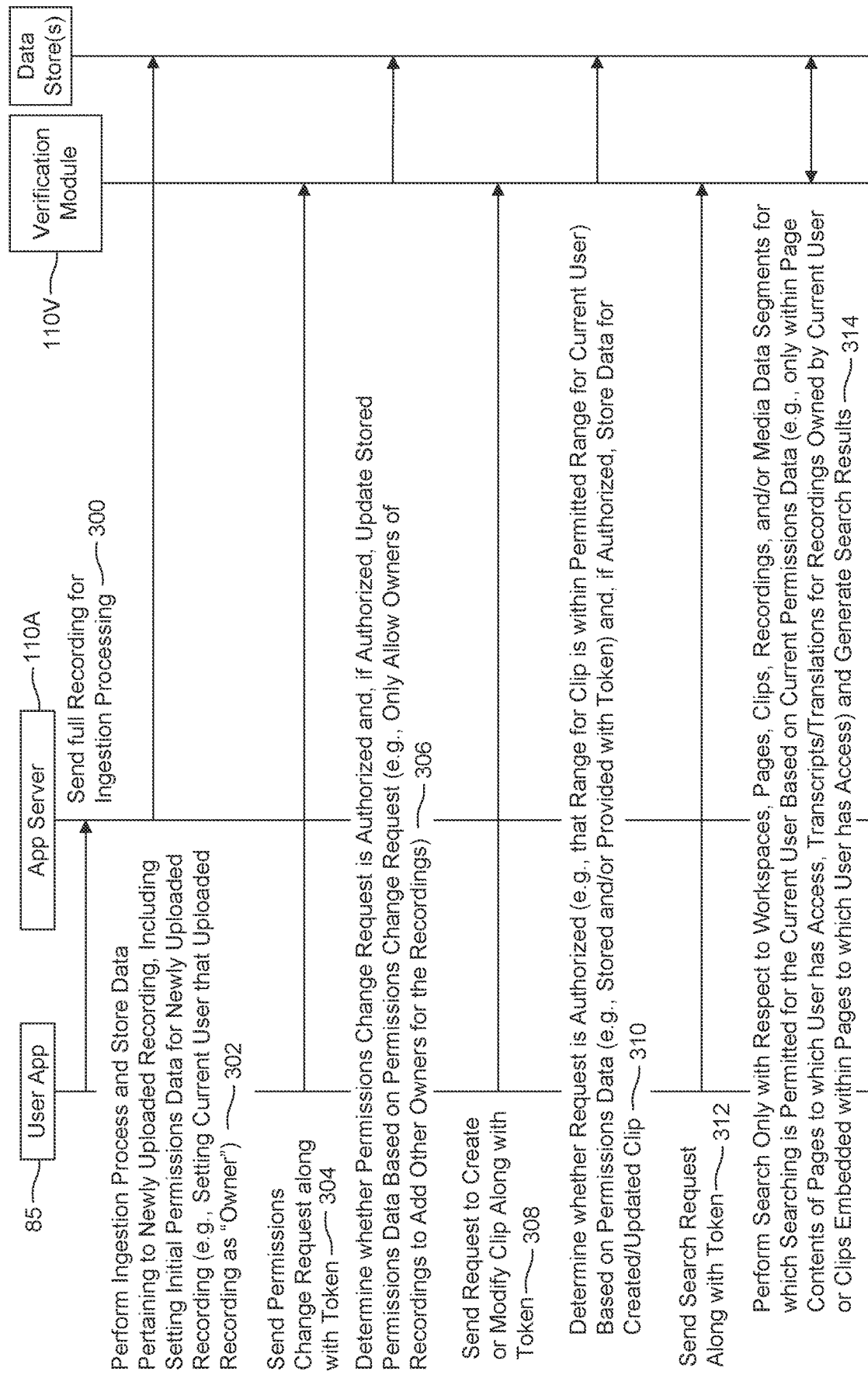
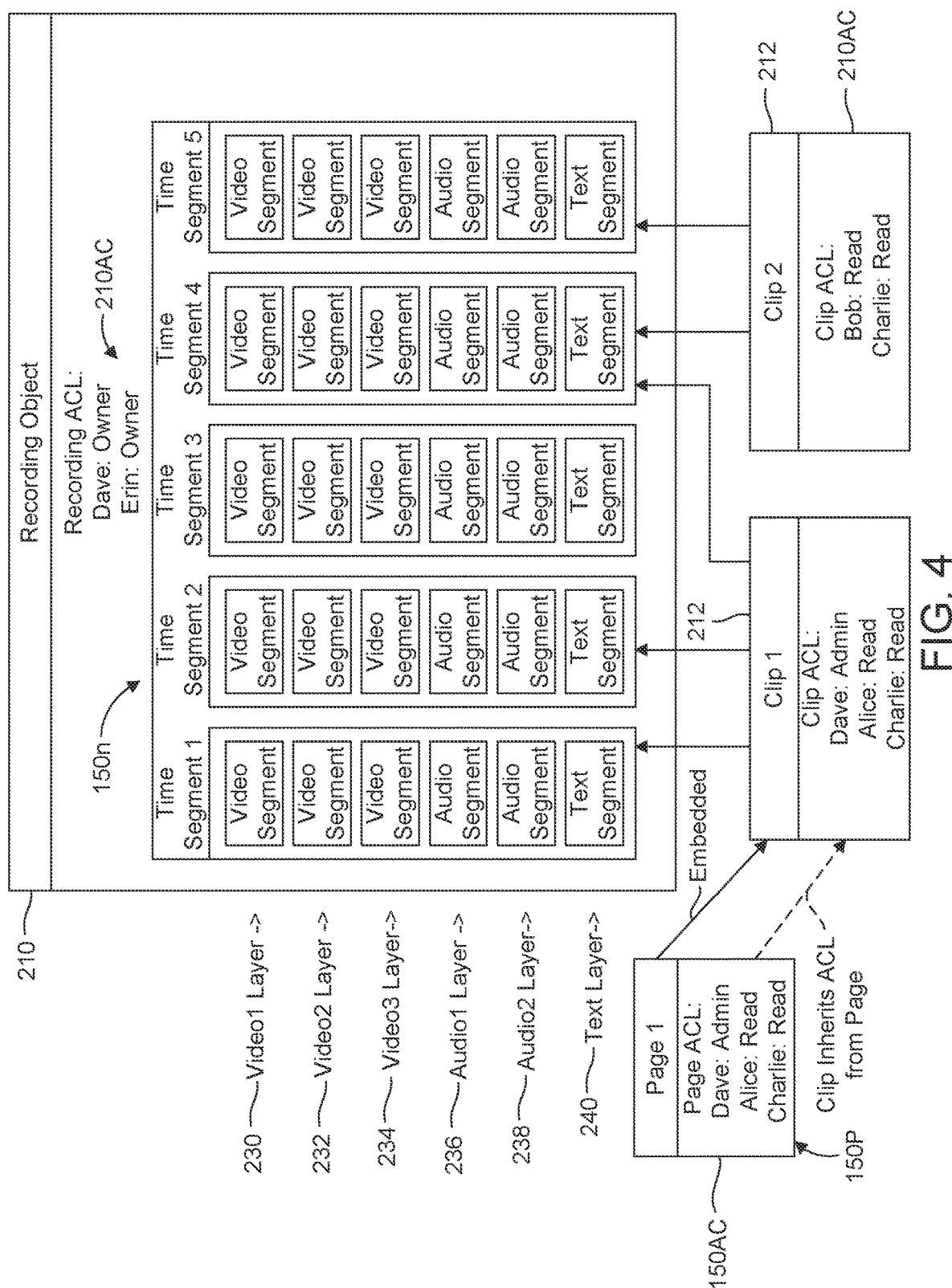


FIG. 3



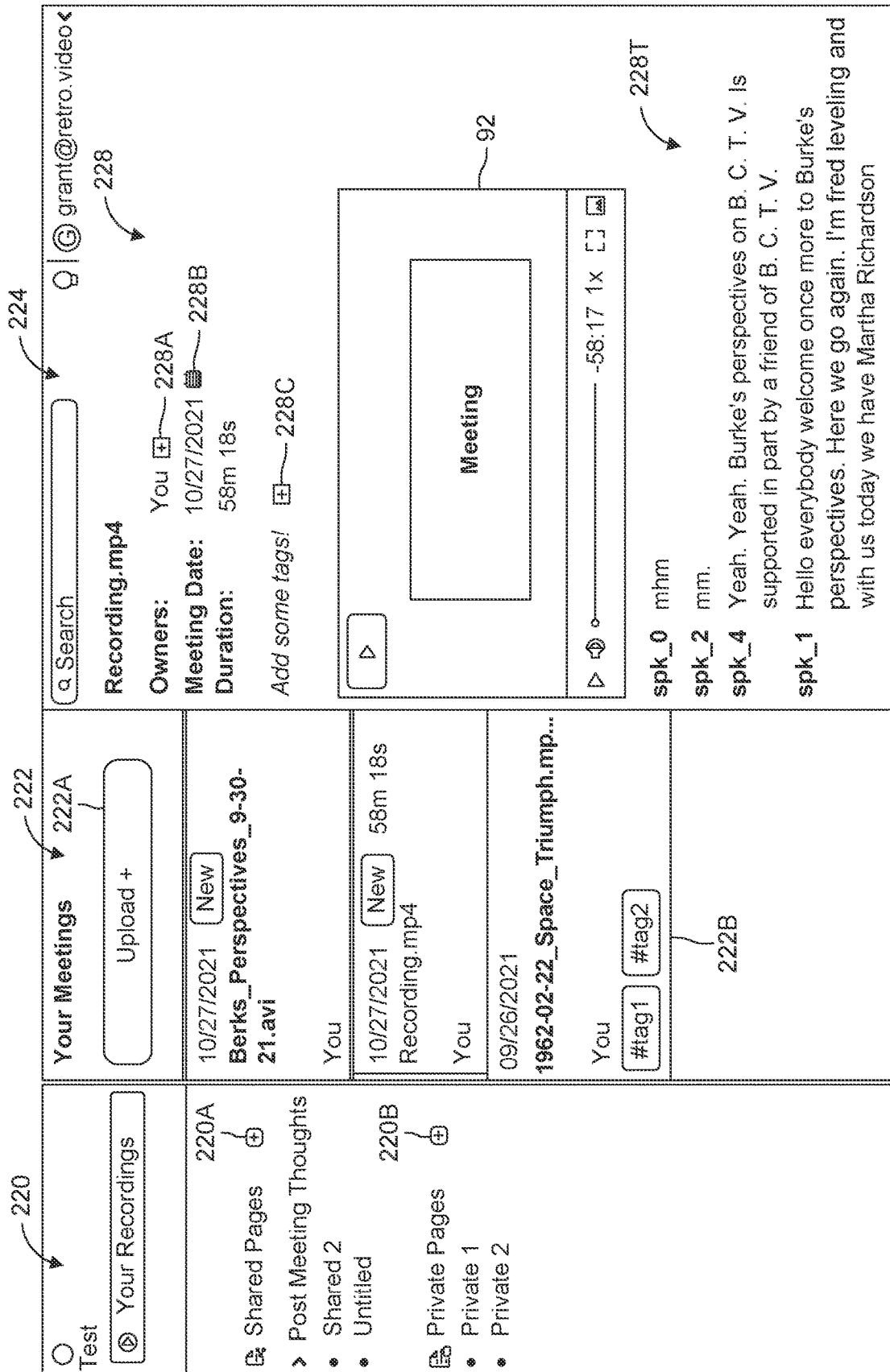


FIG. 5A

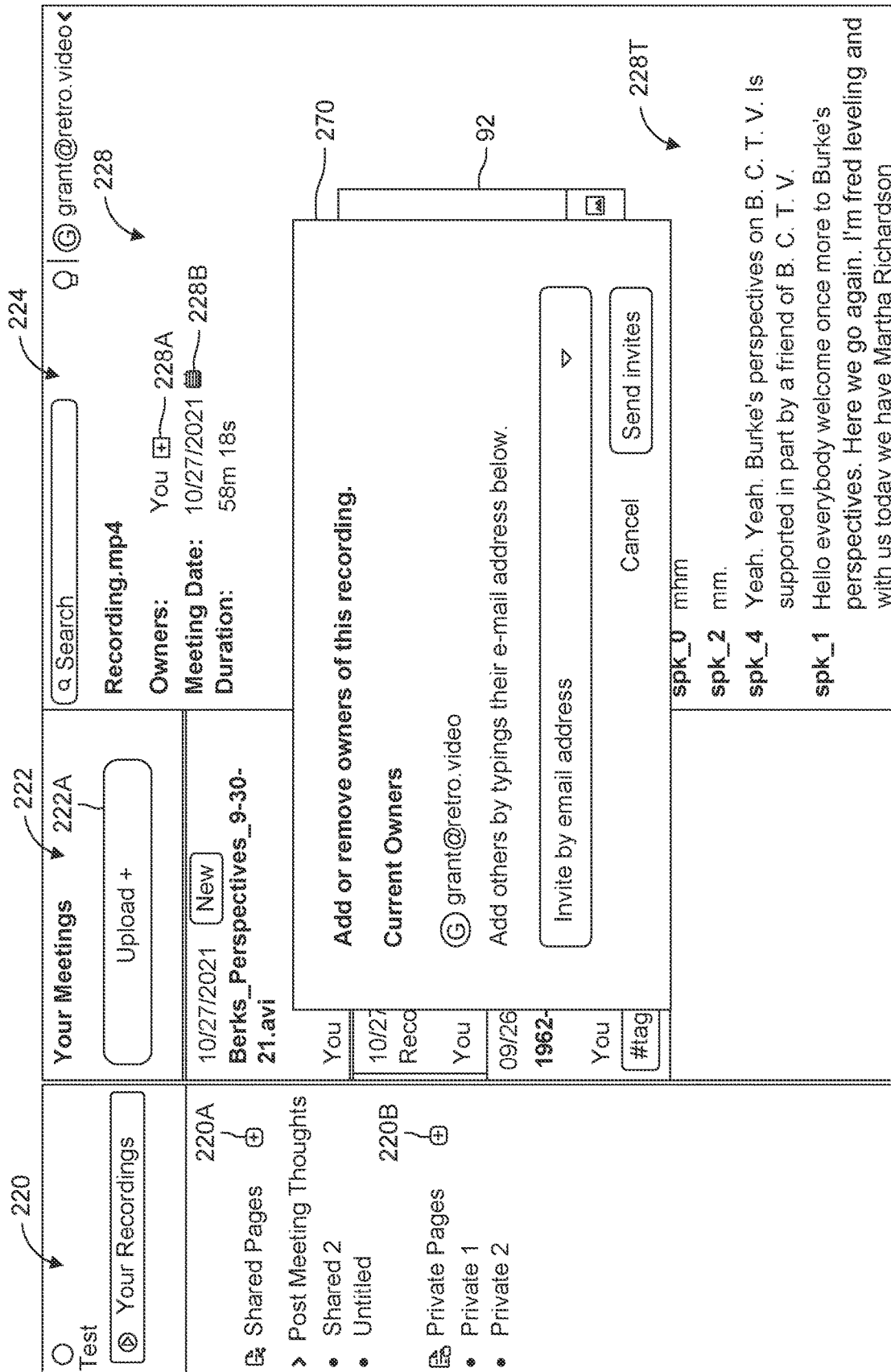


FIG. 5B

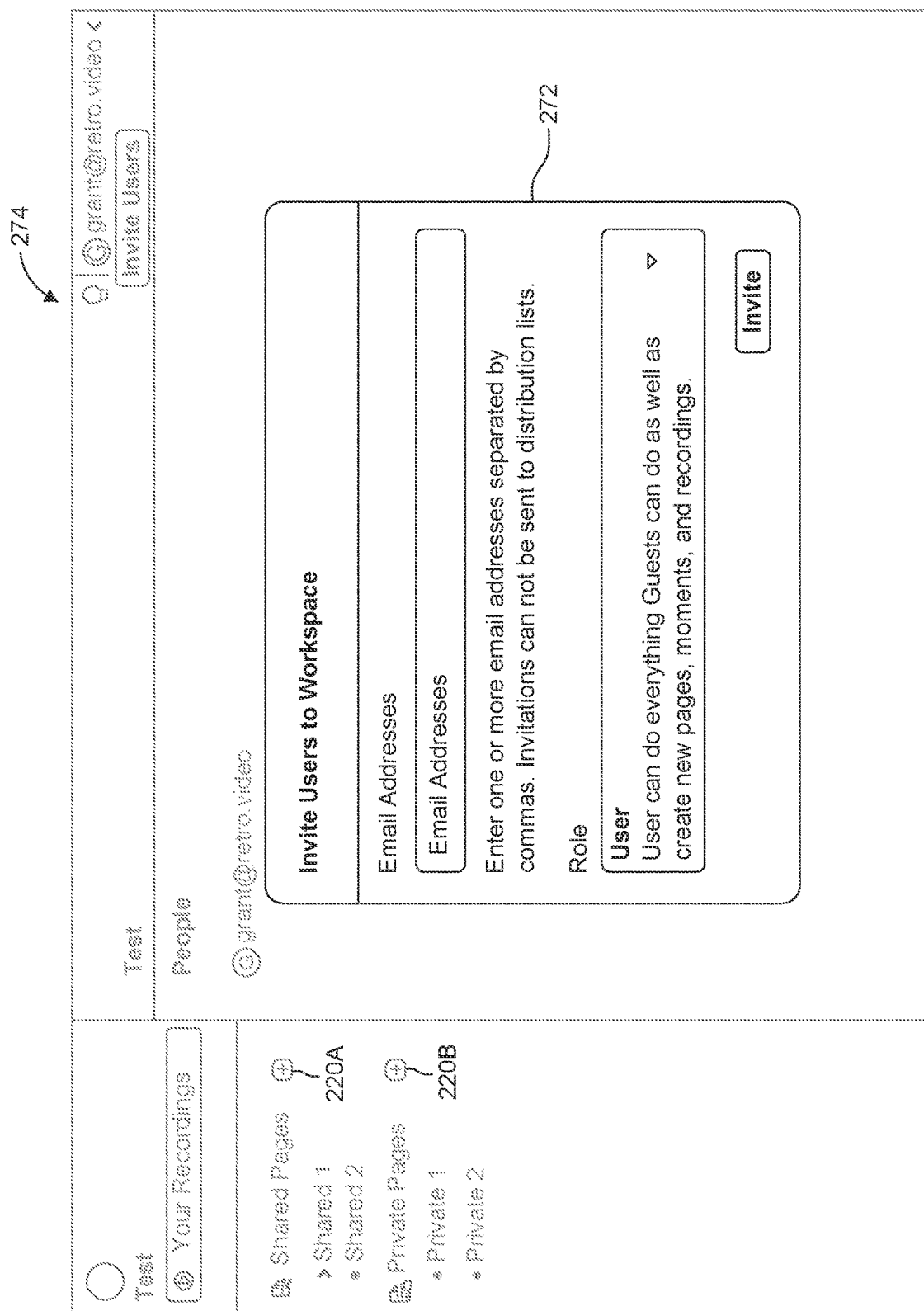


FIG. 6A

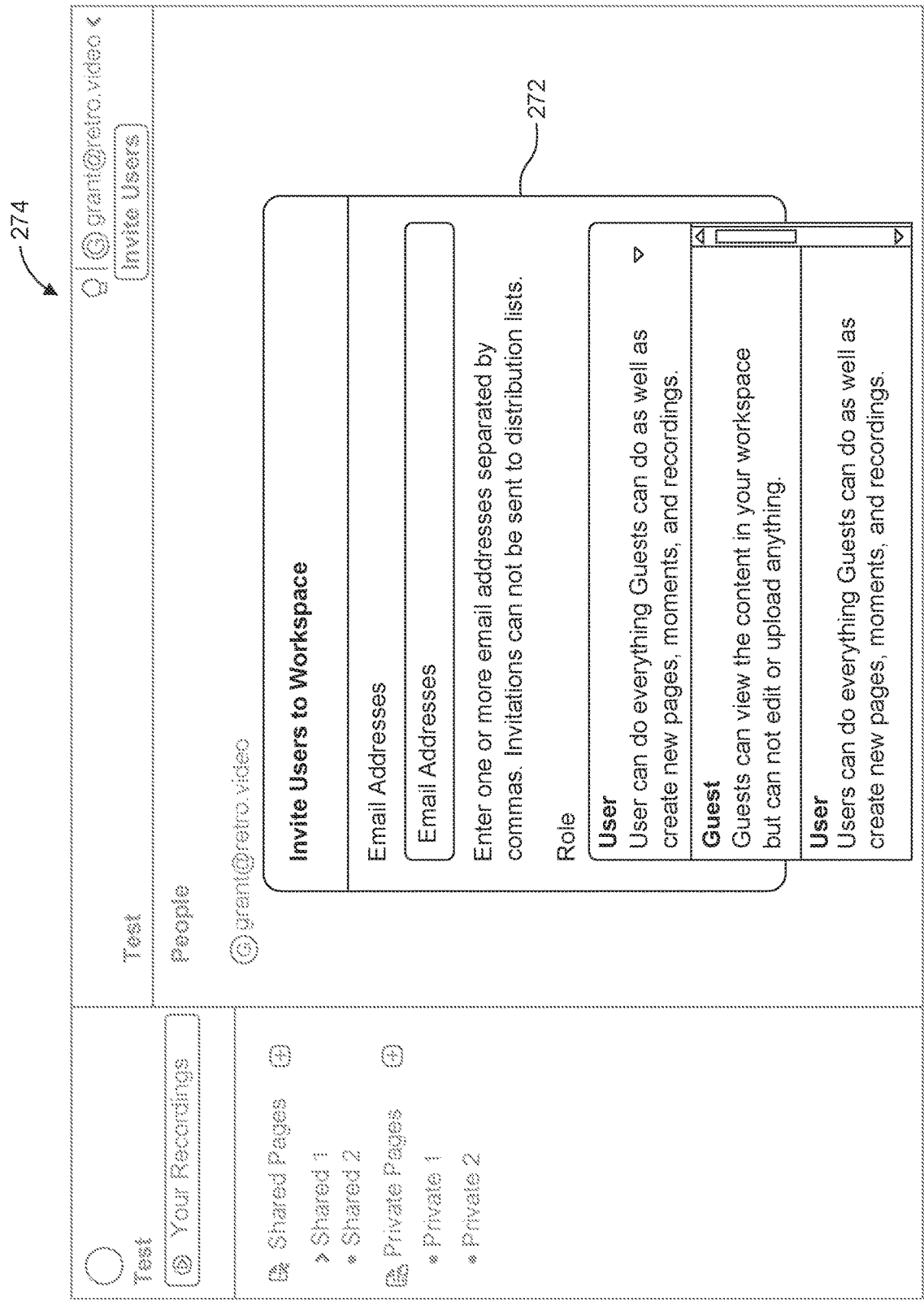
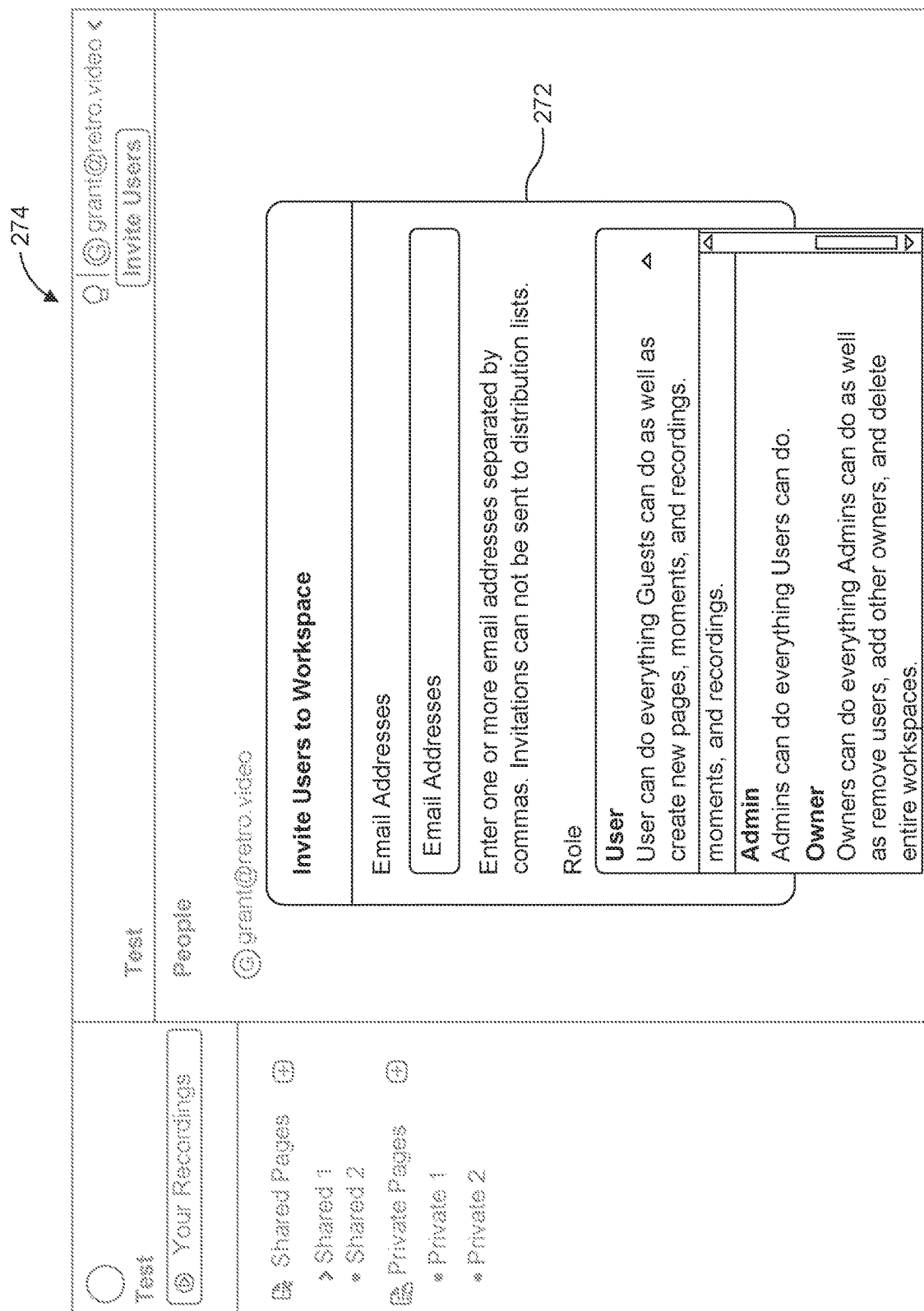


FIG. 6B



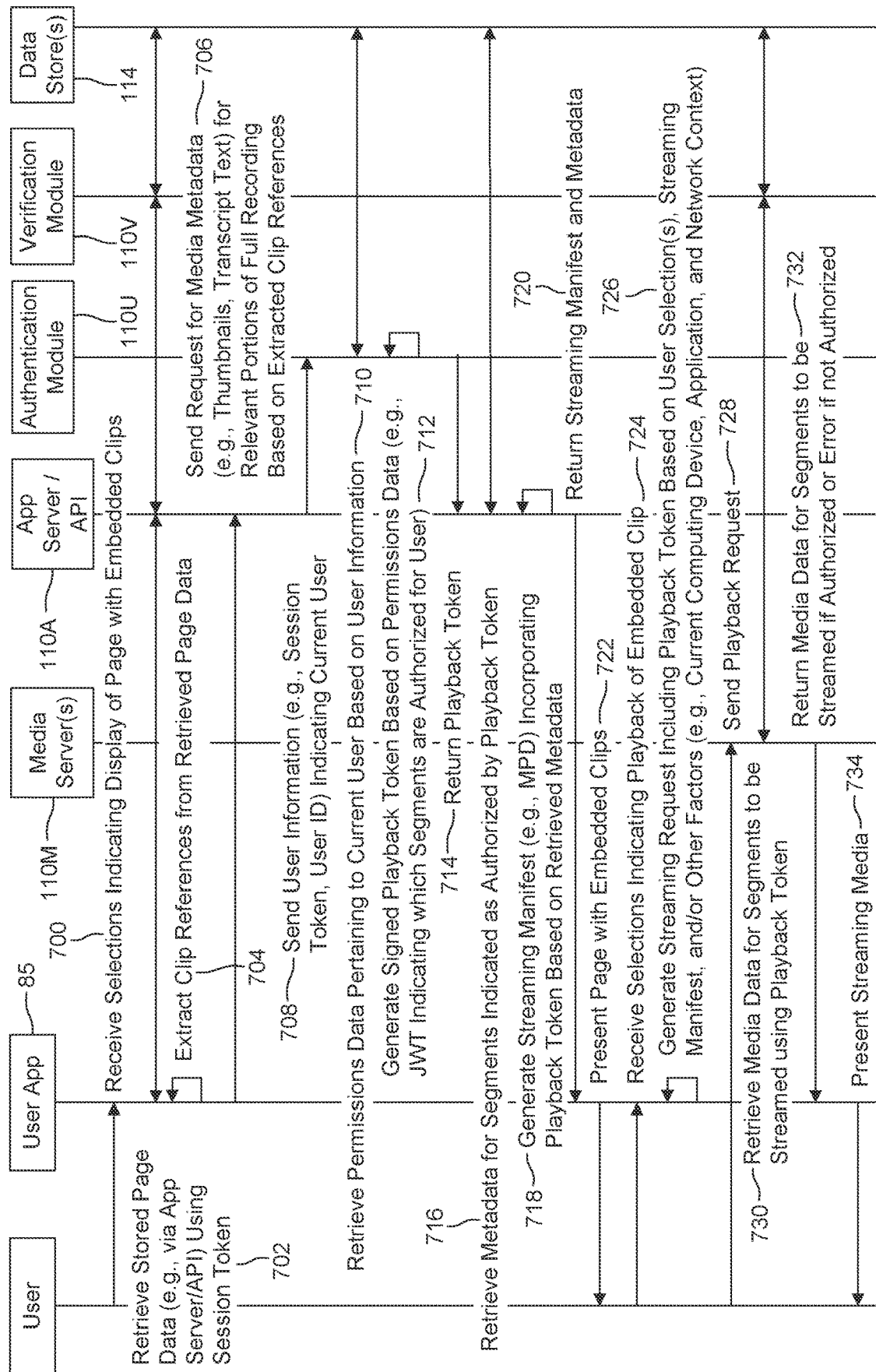


FIG. 7

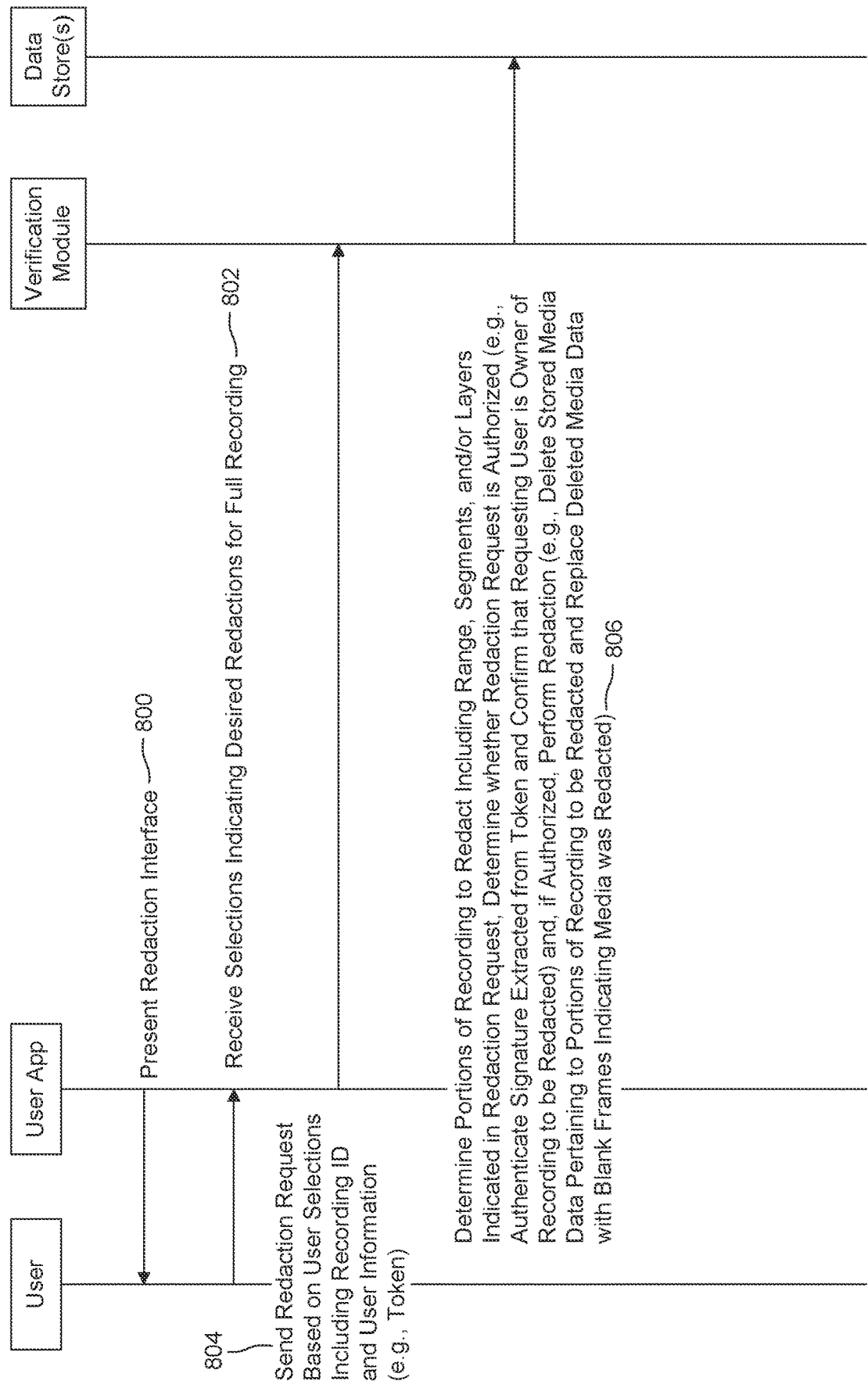
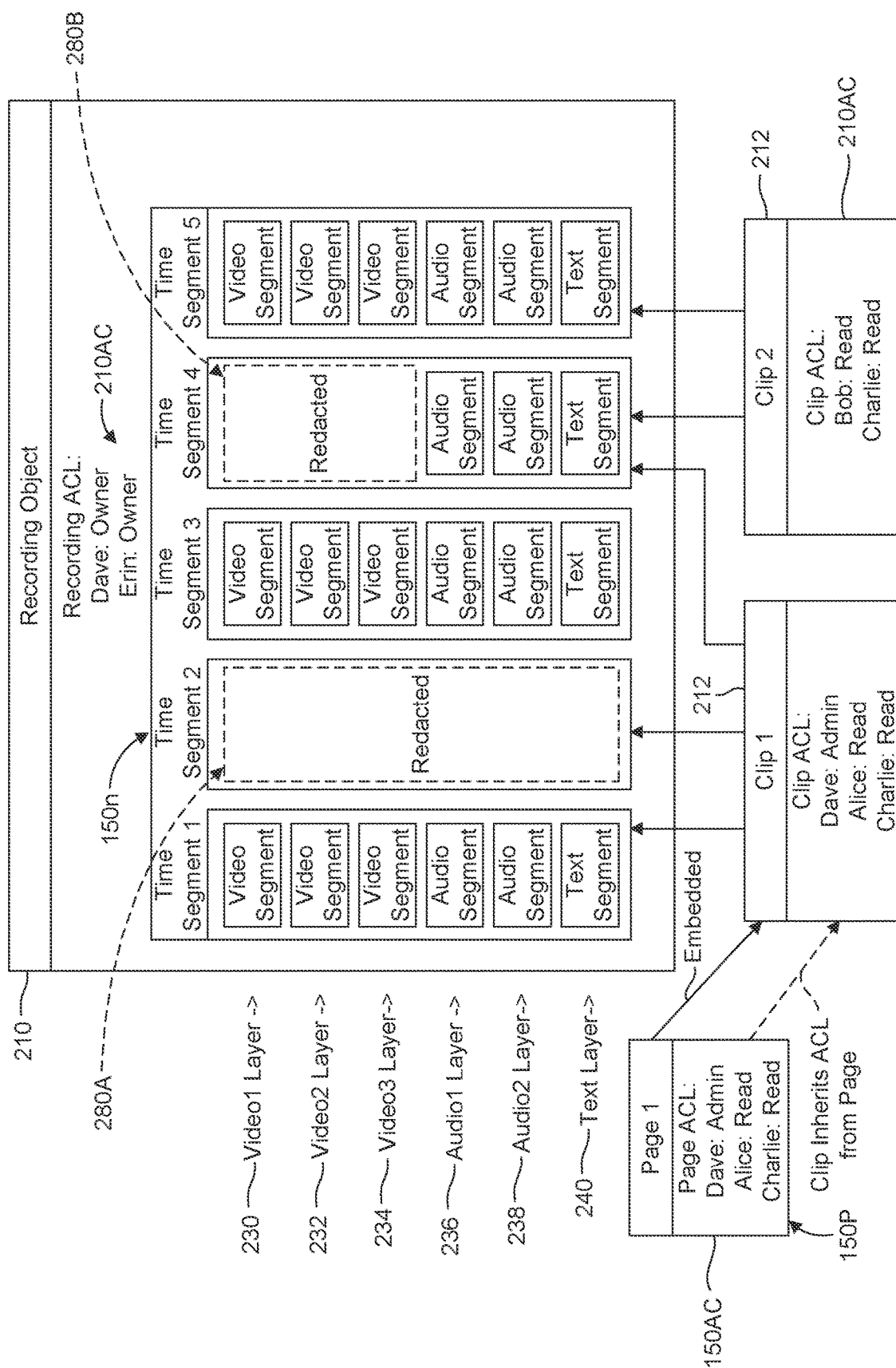


FIG. 8



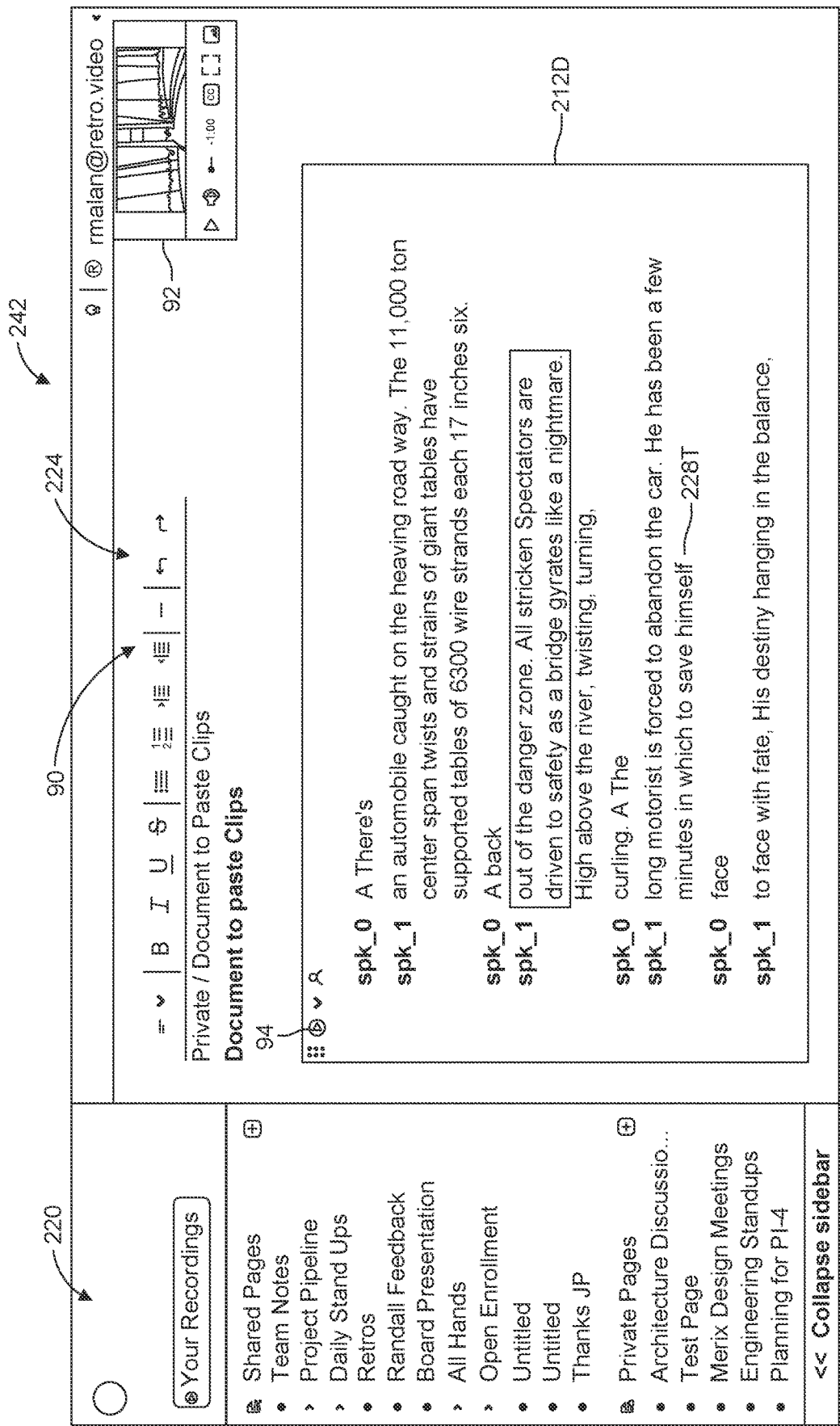


FIG. 10A

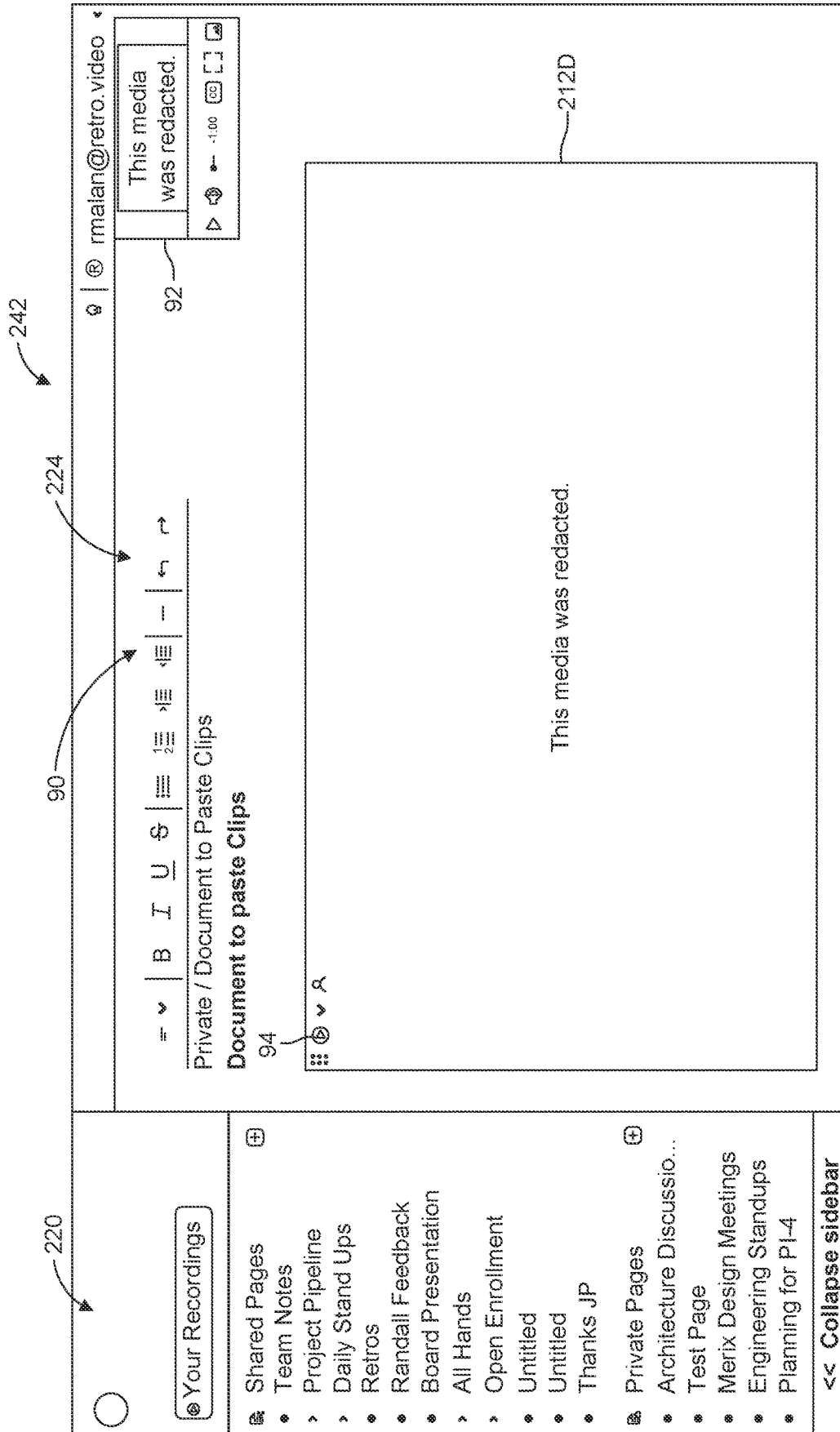


FIG. 10B

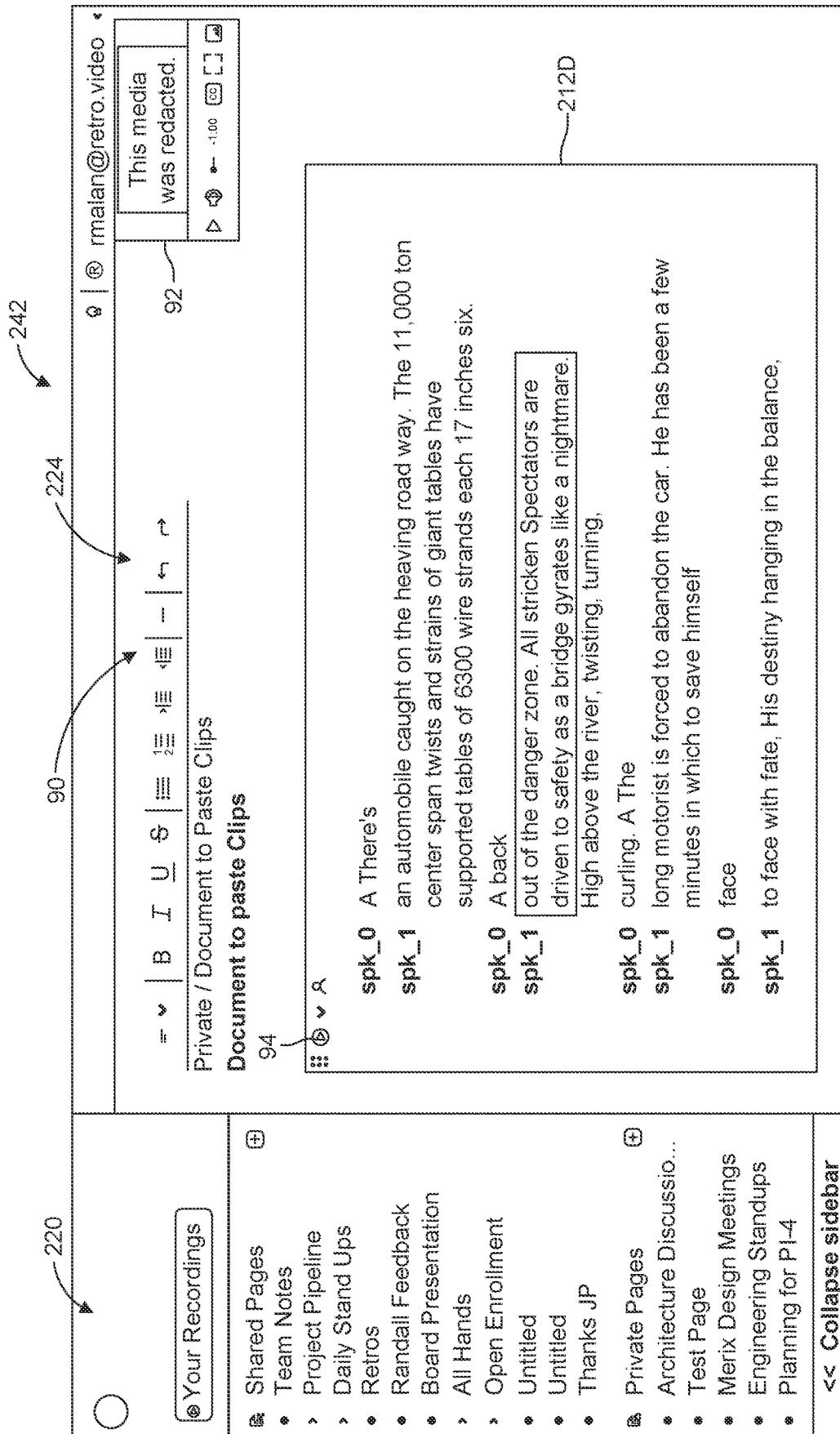


FIG. 10C

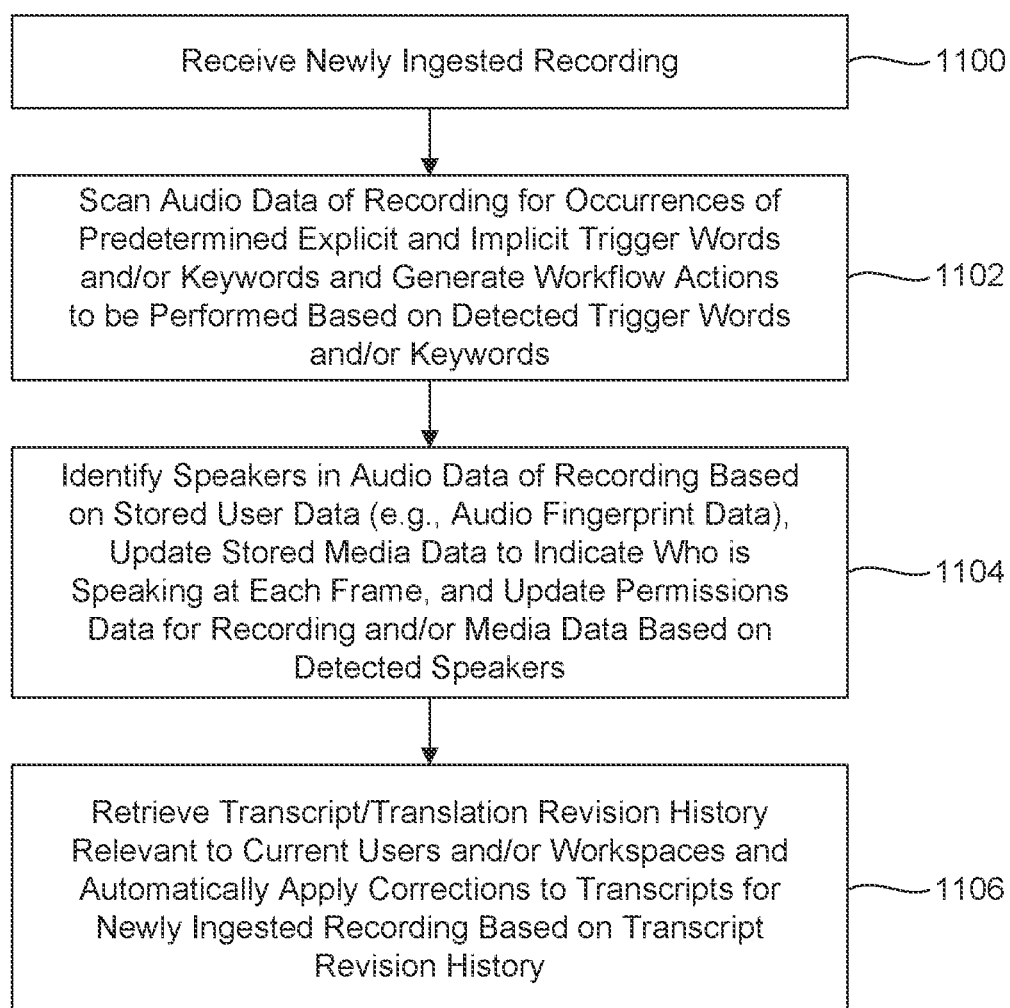


FIG. 11

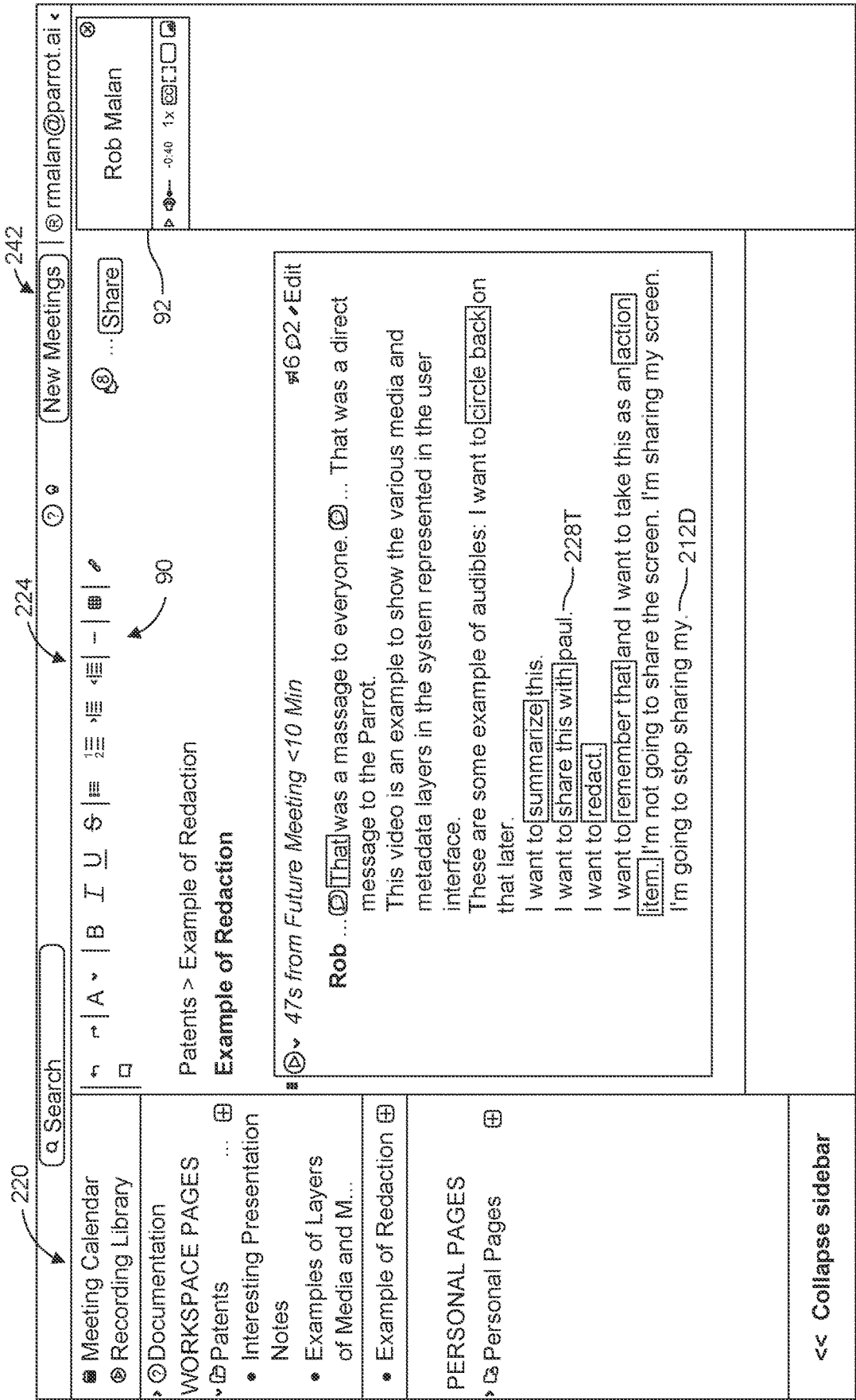


FIG. 12A

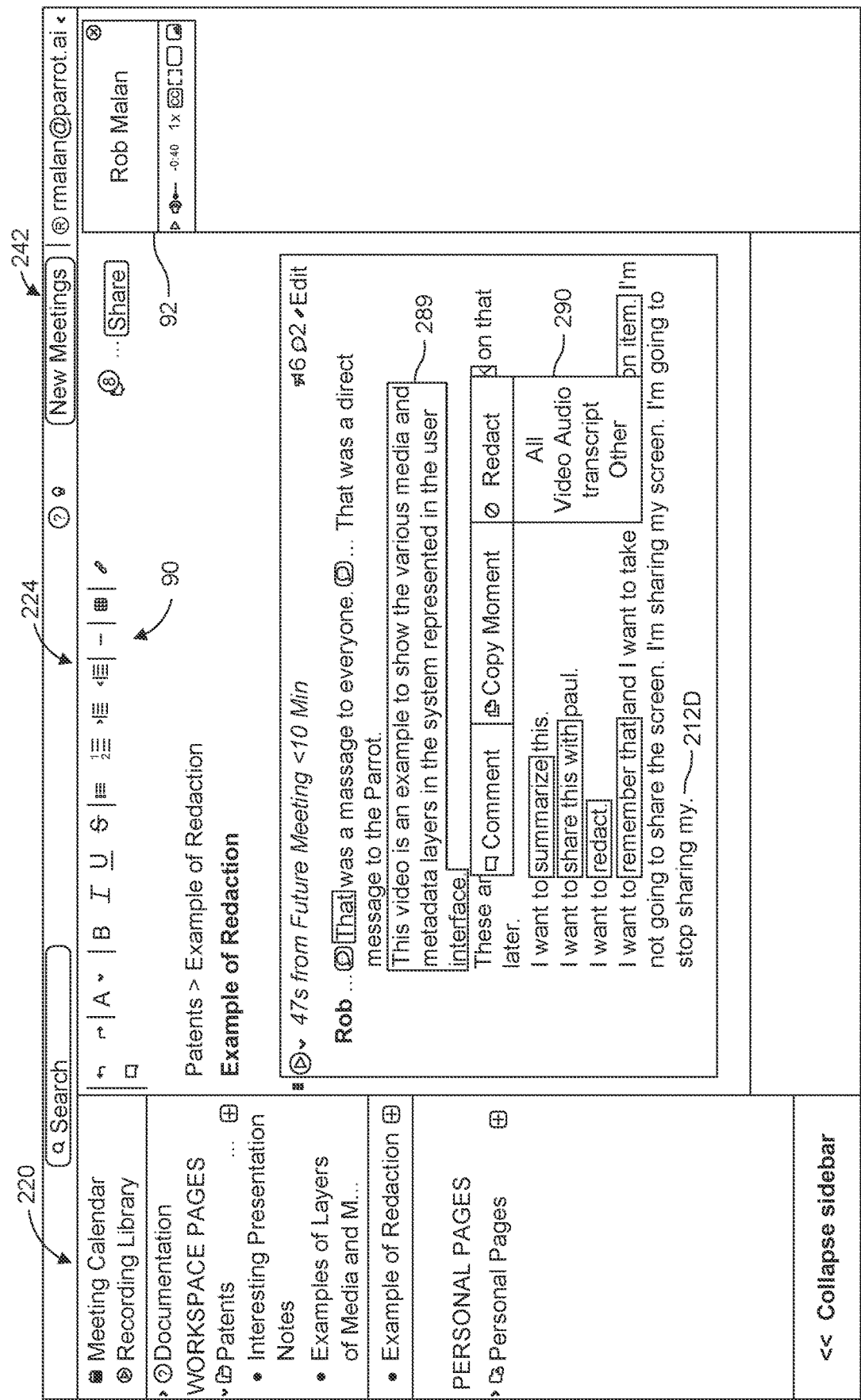


FIG. 12B

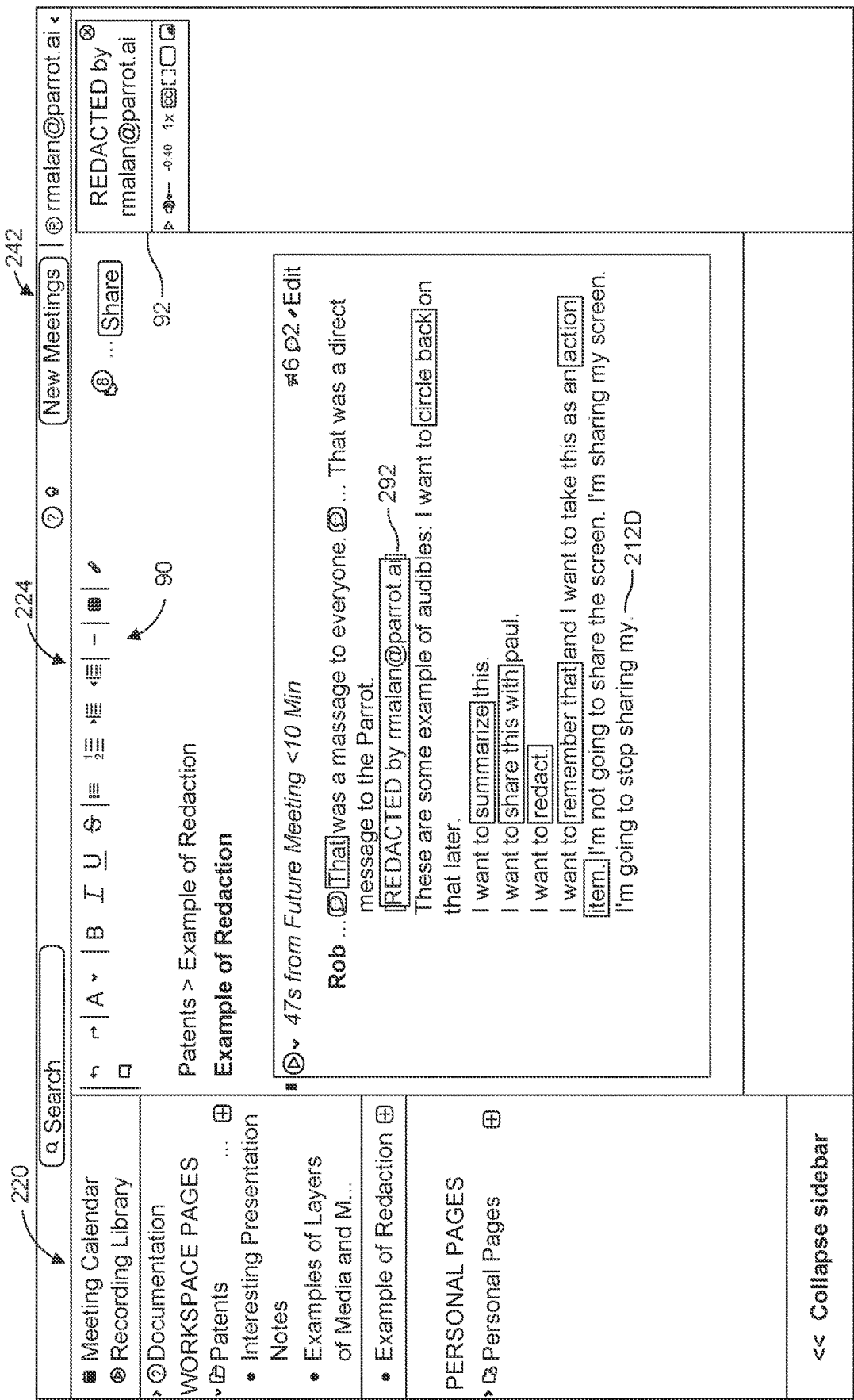


FIG. 12C

SYSTEM AND METHOD FOR ACCESS CONTROL, GROUP OWNERSHIP, AND REDACTION OF RECORDINGS OF EVENTS

RELATED APPLICATIONS

[0001] This application claims the benefit under 35 USC 119(e) of U.S. Provisional Application No. 63/280,830, filed on Nov. 18, 2021, and 63/280,837, filed on Nov. 18, 2021, both of which are incorporated herein by reference in its entirety.

[0002] This application is related to U.S. patent application Ser. No. _____, filed on an even date herewith, entitled “System and method for documenting recorded events,” and sharing inventors herewith, which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

[0003] Video conferencing uses audio, video, and video and static media streaming to allow users who are located in different places to communicate with each other in real time and hold on-line meetings in a variety of contexts, including business, government, education, and personal relationships, to name a few examples. In a typical implementation, audio and/or video capture devices (e.g., microphones and cameras connected to or built into user devices such as desktop computers, laptop computers, smart phones, tablets, mobile phones, and/or telephones) capture audio containing speech of users or groups of users at each location and video visually depicting the users or groups of users and the user devices distribute static images and/or video that is being presented by and for the users. The audio and video data from each location is possibly combined and streamed to other participants of the meeting and can even be recorded and stored (e.g., as a media file) that can later be accessed directly or streamed, for example, to non-participants of the meeting seeking to find out what was discussed or participants of the meeting seeking to engage with the contents of the meeting after the fact.

[0004] At the same time, productivity client and cloud-based platforms such as word processing, presentation, publication, and note-taking programs exist for inputting, editing, formatting, and outputting text and still images. These are increasingly implemented in an online or hybrid online/desktop context (e.g., as a web application presented in a web browser, or as a desktop application or mobile app connected to a cloud-based platform), allowing for sharing and collaboration of the same document and files between multiple users. Notable examples include Microsoft Word and its related productivity programs included in the Microsoft Office 365 productivity suite developed by Microsoft Corporation and Google Docs and its related productivity programs included in the G Suite or Google Drive platforms developed by Alphabet Inc. Similarly, hypertext publication platforms such as wikis present, typically in a web browser, text and still images while also allowing collaboration between users in inputting, editing, formatting, and outputting the published content, often using a simplified markup language in combination with hypertext markup language (HTML).

SUMMARY OF THE INVENTION

[0005] Existing productivity and hypertext publication platforms do not effectively deal with audio and video and

other time-indexed media. In general, time-indexed media often combines audio, video, graphic, and/or text information with a durational or temporal dimension such that the media is created, presented, and experienced over a period of time. Common examples include audio and video media, but can also incorporate written textual and graphic information (e.g., slides or screenshares displayed simultaneously with a spoken presentation).

[0006] For example, all major client and cloud services that provide document editing as a feature almost universally handle time-indexed media using pointers or links to monolithic media objects such as files. Often, these media objects can only be referenced in a page or document as a thumbnail screenshot from the original recording or an opaque hyperlink to a media object (e.g., mp4) or location within the object's file. These services do not treat contents of time-indexed media, including other information such as the speakers, words, and transcripts, as part of the page or document but rather as opaque attributes of the recording that can only be accessed by an external media player that is launched when a user clicks on the thumbnail or link.

[0007] Sharing and distributing media in an uncontrolled manner presents challenges involving security and access control if it were applied in other contexts. In such a system, clips that reference portions of the same underlying stored media recording can be shared among different groups of people that do not necessarily coincide or even overlap. This presents a challenge when the underlying media needs to be accessed, for example, for streaming playback, as a user may have permission to access one portion but not other portions of the same recording. In one example, a recording of a board meeting for a company might encompass different discussions on topics with varying levels of sensitivity. Some of the discussions might be able to be distributed more widely to employees of the company or even individuals outside the company, while other discussions occurring in the same meeting (e.g., concerning compensation, sensitive policy matters) might need to remain confidential and possibly limited to only those participants who were present.

[0008] Moreover, currently, events such as in-person or virtual meetings might not be recorded due to potential issues surrounding participants' comfort with being recorded. Similarly, for events that are recorded, participants' awareness of being recorded could hinder the quantity or quality of their contributions during meetings. More generally, it is desirable for participants to be comfortable with more widespread adoption of recording events and to have control over their own contributions to a recorded event, such as video layers of portions of a recording that visually depict the participants or audio/transcription layers of portions of a recording that depict the participant's voice, to name a few examples.

[0009] The presently disclosed media presentation system addresses these issues by providing efficient, fine-grained access control for an underlying base of stored time-indexed media data for recordings, including a group ownership rights and management scheme along with redaction functionality for certain users of the stored recordings.

[0010] A page editor enables embedding of clips referencing portions of a full recording into pages, which can be shared with a plurality of other users with a variety of different permissions levels (e.g., view, edit). Generally, when a page is loaded, text-layers (e.g., transcript text) of the media data referenced by any embedded clips are retrieved

from the underlying stored media data along with a playback descriptor or manifest including a playback token that, in general, grants access only to the referenced portion of the recording by describing ranges of media data the user is allowed to access. When the page is displayed, the transcript text is presented as part of the page along with interface elements (e.g., a play button) that allow activation of a media player that streams only the portion of the full recording referenced by the embedded clip and authorized by the playback token, which is used to request the streaming media.

[0011] Moreover, any users granted access to a portion of the recording via an existing clip embedded within a page shared with them by another user can, in turn, share that same portion with other users by embedding a new clip based on the existing clip into one of their pages that they then share with others. When embedding the new clip, the user can narrow the scope (e.g., the extent of the full recording referenced by the clip) of the new clip with respect to the existing clip but is prevented from expanding the scope beyond what was shared with them originally. Thus, the present system allows hierarchical sharing of media.

[0012] In this way, the presently disclosed media presentation system provides fine-grained access control with respect to shared portions of the underlying stored media data of recordings.

[0013] At the level of full recordings, access control and permissions are further based on a group ownership scheme, in which any recording can have one or many owners that have rights in accessing and modifying the recording. In one example, any owners of a recording can add other owners for the recording (e.g., by selecting other users to add on a user interface provided by the system) but are prevented from removing any current owners. In another example, any owners of a recording can add and remove other owners. The group ownership scheme might specify various levels of ownership, the levels associated with different combinations of access privileges such as viewing, redacting, adding/removing owners, to list a few examples. In embodiments, the media presentation system may initially set owners of a newly added recording based on which user uploaded or imported the new recording, based on analysis of the new recording, and/or based on integration with systems or services that originally hosted the event depicted in the recording and/or generated the raw media data for the recording. Moreover, in different configurations, owners of recordings can correspond to different functional roles potentially played by users with respect to events and recordings of events, including users who added the recordings (as previously mentioned), users who are owners of the audio and/or video capture devices used to generate the recording, users who are known to have been present at and/or contributors to the events, users who are depicted in video or audio layers of the recordings, and/or owners of particular objects and/or spaces visually depicted in the recording, to name a few examples. In one embodiment, the system determines the different functional roles using the metadata for the recordings and/or information extracted from calendars and/or event invitations via an API or other integration with a calendar system.

[0014] The media presentation system also allows redaction of a recording by certain users based on different possible configurations. In general, redaction of a recording prevents certain information within the stored media data of

the recording from being accessed by users of the system. In a typical example, a recording can be redacted by removing the information from the underlying stored media data and serving the redacted media data in response to access requests. Preferably, the system allows owners (and only owners) of the recordings to redact the recordings. In another example, any part of any recording can be redacted by any of its owners at any time, or alternatively, a user can redact only portions of the recording having certain predetermined associations with the user (e.g., parts of a recording where the user was speaking or visually depicted, parts of a recording where an object or space known to be owned by the user is visually depicted, an individual stream captured by the user's audio and/or video capture device). When recordings are redacted, the media presentation system modifies or deletes the stored media data of the recordings such that the redactions are reflected in any clips referencing the portion of the recording that was redacted. Redactions can include deleting any layer (e.g., video, audio, transcript text, translation text, presentation slides, metadata, user-specified and/or automatically generated tags, user information, and/or user-specified notes, comments, and/or action items) and/or replacing the deleted layer(s) with blank frames, user-specified images, video data, audio data, and/or text indicating that the portion of the recording was redacted. In one embodiment, redaction is permanent and includes destruction of all relevant artifacts of the media data for the redacted portion of the recording from the storage medium storing the media data. The media presentation system may provide an option to undo redactions within a predetermined period of time and/or delay full deletion and/or destruction of the stored data until after a predetermined age-out period (e.g., 30 days). In one embodiment, the media presentation system prompts the user performing the redaction to type a keyword such as "redact" into a text input box, in response to which the system starts a countdown timer of a predetermined duration (e.g., 10 seconds). The countdown timer is reset in response to the system detecting that the text entered into the text input box no longer matches the keyword. On the other hand, in response to the countdown timer expiring or reaching the end of the countdown, the system performs the requested redaction irreversibly.

[0015] In general, according to one aspect, the invention features a system and method for controlling access to recordings of events. Time-based media data for a recording of an event is stored in a data store along with permissions data indicating access permissions associated with different portions of the recording. Access to requested portions of the stored media data is then controlled based on the stored permissions data associated with the requested portions of the recording.

[0016] In embodiments, pages with embedded clip objects are generated and presented, and access to requested portions of the stored media data is controlled based on permissions data associated with pages with embedded clip objects that reference the requested portions of the stored media data. Here, the access to the requested portions of the stored media can be controlled by preventing users from expanding the scope of new and/or modified clip objects beyond that of previously existing clip objects shared with and/or accessible by the users.

[0017] In another example, access to requested portions of the stored media data is controlled by generating a playback token for each user indicating portions of the recording that

are allowed for that user based on the stored permissions data and granting or denying requests from users for portions of the stored media data based on validation of playback tokens included with the requests.

[0018] Tag information indicating portions of the recordings to tag and users associated with the indicated portions of the recordings is received by the system, in which case the system updates the stored media data corresponding to the indicated portions of the recordings to include tags indicating the users associated with the indicated portions of the recordings based on the tag information. A tagging interface for receiving selections indicating the portions of the recordings to tag and the users associated with the selected portions of the recordings can be provided, with the tag information being generated based on the received selections. Access to the stored media data is then controlled based on the tags.

[0019] In general, according to another aspect, the invention features a system and method for group ownership of recordings referenced in a media presentation system. Time-based media data for recordings of events is stored in a data store along with ownership information indicating one or more owners for each of the recordings. Access to the stored media data for the recordings is controlled based on the ownership information for the recordings, and changes to the ownership information for the recordings is restricted based on current ownership information for the recordings and predetermined group ownership rules.

[0020] In general, according to another aspect, the invention features a system and method for redaction of recorded media in a media presentation system. Time-based media data for recordings of events is stored in a data store. A redaction interface for receiving selections indicating redactions to the stored media data is then provided, and the stored media data is redacted in response to the received selections indicating the redactions. In response to requests for the stored media data, the redacted media data is provided.

[0021] In embodiments, redaction of the stored media data is allowed or restricted based on whether current users are indicated as owners of the recordings.

[0022] Redacting the stored media data in response to the received selections indicating the redactions comprises receiving selections of portions of the recording to be redacted, and redacting the stored media data in response to the received selections comprises redacting stored media data only for the selected portions of the recording to be redacted. Here, the received selections can include selections of particular video, audio, text, and/or metadata layers for the portions of the recording to be redacted, in which case the redaction is performed only with respect to the particular video, audio, text, and/or metadata layers indicated for redaction.

[0023] The redaction might be performed by deleting the stored media data pertaining to the indicated portions of the recording and/or replacing the deleted media data with blank frames and/or text indicating that the media was redacted, a user who made the redaction, and/or time information for the redaction, by updating stored index data such that the redactions are reflected in search results, and/or by destroying or clearing all artifacts of the media data for the redacted portion of the recording from the data store. An option can be provided to undo redactions within a predetermined period of time and/or delay full deletion and/or destruction of the stored data until after a predetermined age-out period.

[0024] In general, according to another aspect, the invention features a system and method for group ownership of recordings referenced in a media presentation system. Time-based media data for recordings of events is stored in a data store along with ownership information indicating one or more owners for each of the recordings. Access to the stored media data for the recordings is then controlled based on the ownership information for the recordings. Moreover, changes to the ownership information for the recordings are restricted based on current ownership information for the recordings and predetermined group ownership rules.

[0025] The above and other features of the invention including various novel details of construction and combinations of parts, and other advantages, will now be more particularly described with reference to the accompanying drawings and pointed out in the claims. It will be understood that the particular method and device embodying the invention are shown by way of illustration and not as a limitation of the invention. The principles and features of this invention may be employed in various and numerous embodiments without departing from the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] In the accompanying drawings, reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale; emphasis has instead been placed upon illustrating the principles of the invention. Of the drawings:

[0027] FIG. 1A is schematic diagram of an exemplary media presentation system according to one embodiment of the present invention;

[0028] FIG. 1B is a schematic diagram of the media presentation system showing components of the system in additional detail;

[0029] FIG. 2 is a sequence diagram illustrating an exemplary access control process performed by the media presentation system;

[0030] FIG. 3 is a sequence diagram illustrating examples of how the media presentation system sets and uses permissions data for access control;

[0031] FIG. 4 is a schematic diagram showing exemplary processed and segmented media data, page data, and clip data stored in data store(s) of a server system of the media presentation system;

[0032] FIG. 5A shows an exemplary recording screen of a graphical user interface (GUI) rendered on a display of a user device of the media presentation system;

[0033] FIG. 5B is an illustration of an exemplary add owners window of the GUI overlaid on the recording screen of FIG. 5A;

[0034] FIG. 6A shows an exemplary invite users window of the GUI;

[0035] FIG. 6B shows the invite users window of FIG. 6A, showing a permissions selector of the invite users window expanded to reveal different permissions level options;

[0036] FIG. 6C shows the invite users window of FIG. 6B, showing the expanded permissions selector scrolled down to reveal additional permissions level options;

[0037] FIG. 7 is a sequence diagram illustrating an exemplary access control process for streaming the processed and segmented media data;

[0038] FIG. 8 is a sequence diagram illustrating an exemplary redaction process for the processed and segmented media data;

[0039] FIG. 9 is an illustration of the processed and segmented media data of FIG. 4 after redaction;

[0040] FIGS. 10A-10C show exemplary page editor screens of the GUI, each showing how different segments of the media data of FIG. 9 would be presented;

[0041] FIG. 11 is a flow diagram illustrating an exemplary automation process of the media presentation system; and

[0042] FIGS. 12A-12C show exemplary page editor screens of the GUI, each showing how different segments of the media data of FIG. 9 would be presented.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0043] The invention now will be described more fully hereinafter with reference to the accompanying drawings, in which illustrative embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

[0044] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items. Also, all conjunctions used are to be understood in the most inclusive sense possible. Thus, the word “or” should be understood as having the definition of a logical “or” rather than that of a logical “exclusive or” unless the context clearly necessitates otherwise. Further, the singular forms and the articles “a”, “an” and “the” are intended to include the plural forms as well, unless expressly stated otherwise. It will be further understood that the terms: includes, comprises, including and/or comprising, when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. Further, it will be understood that when an element, including component or subsystem, is referred to and/or shown as being connected or coupled to another element, it can be directly connected or coupled to the other element or intervening elements may be present.

[0045] Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. It will be further understood that terms, such as those defined in commonly used dictionaries, should be interpreted as having a meaning that is consistent with their meaning in the context of the relevant art and will not be interpreted in an idealized or overly formal sense unless expressly so defined herein.

[0046] In general, the present invention relates to a video conferencing, productivity, and media presentation system for presenting, editing, and sharing time-indexed media such as audio and/or video recordings of events such as meetings, presentations, conferences, or lectures, which occur in a variety of contexts, including business, government, education, and in personal relationships, to name a few examples. In examples, the media presentation system provides a hypertext publication platform and/or productivity program enabling collaboration by a plurality of users in viewing, inputting, editing, formatting, and outputting user-authored content such as text and still images along with the shared time-indexed media. More particularly, the present inven-

tion concerns a system and method for access control, group ownership, and redaction of recorded media in a media presentation system.

[0047] FIG. 1A is schematic diagram of an exemplary video conferencing, productivity and media presentation system 100 according to one embodiment of the present invention.

[0048] In one example, the video conference meeting 10 is hosted by a video conferencing server system 12. As is the case with many presently-available platforms such as Google Meet offered by Alphabet Inc, Zoom offered by Zoom Video Communications, Inc, and Microsoft Teams offered by Microsoft Corporation, the video conferencing server system 12 receives real-time audio and/or video and presentations from the user devices 80 of each of the meeting participants and distributes the audio/video and/or presentations to the user devices of the other participants. The audio/video and/or presentations are displayed on the user devices, often in windows or full screen presentations in which the participants are shown in panes, with other panes being dedicated to shared presentations, often in a screen or presentation sharing arrangement.

[0049] Also provided is a productivity and media presentation server system 110. It receives and stores time-indexed media 150 in data store(s) 114. In a common use-case, this time-indexed media is the audio/video/presentations associated with recorded events such as video conference meetings hosted by the video conferencing server system 12. This media presentation system itself is capable of serving documents and streaming the stored time-indexed media to the user devices 80, which present the documents and streaming time-indexed media to users of the user devices via graphical user interfaces 87 rendered on displays 84 of the user devices 80.

[0050] Typically, the time-indexed media 150 is a recording of an event such as a virtual meeting or video conference 10 but can be any type of audio and/or video data and/or any type of digital media with a temporal dimension of any duration.

[0051] In the illustrated example, the event 10 is a virtual meeting with four different participants at four different locations conducted using video and/or audio capture devices (e.g., cameras and microphones connected to or included as internal components of user devices 80 such as desktop computers, laptop computers, smart phones, tablets, mobile phones, and/or telephones) deployed at each of the often different locations. The video and/or audio capture devices capture audio depicting speech of participants or groups of participants at each location and video visually depicting the users or groups of users. In addition to being served and distributed to be presented in real time to the different participants (and/or possibly other participants that are not depicted) on their respective user devices 80 by the video conferencing server system 12, a combined stream of the audio and video data or separate streams from each location/user device are also recorded as raw media files by the media presentation server system 110 or later uploaded to the system 110. These media files of time-indexed data are then combined into documents displayed by page editors 90 that allow for the creation of associated user-authored content 150U such as plain text, formatted text, still images, tables, charts, bulleted lists, and/or other display elements.

[0052] The media presentation server system 110 ingests and processes the audio and/or video streams from each of

the users devices directly or indirectly via the video conferencing server system **12** and records or stores those streams, generally partitioning the meeting's media data **150** into a number of segments **150_n** (e.g., segmented media files) contained by a recording object **210** representing the full recording (e.g., the entire span of the originally ingested recording), and stores the segmented media data **150** in the data store(s) **114** along with clip data or clip objects **212** representing particular portions of the full recording. The clips **212** include recording references (e.g., start/stop times) delineating the extent of the clips with respect to the full recording object **210** and also specific layers of the recording object. In the current example, the clips **212** refer to the specific segments **150_n** of the full recording object **210** that the recording was chunked into.

[0053] In the illustrated example, the event was represented and displayed on the user devices **80** in realtime as part of the video conference **10**. The productivity and media presentation server system **100** also saves and serves a recording of the meeting. A recording object **210** representing this hour-long, for example, recording and containing the many segmented media files **150_n** for the recording is stored along with two user-defined clip objects **212**.

[0054] The first clip object "clip 1" represents a portion of the full recording **210** with a duration of approximately one minute and, accordingly, includes a recording reference defining the one-minute span with respect to the duration of the full recording. Similarly, the second clip object "clip 2" represents a portion of the full recording with a duration of approximately 5 minutes and, accordingly, includes a recording reference defining the five-minute span with respect to the duration of the full recording **210**. These respective clips are typically user defined references for the portions of the full recording that were of interest to the users.

[0055] In the arbitrary illustrated example, while the underlying stored media data corresponding to the portion of the recording represented by the first clip is entirely contained within one of the segmented media files, the underlying stored media data corresponding to the portion of the recording represented by the second clip spans across more than one of the segmented media files.

[0056] In general, the segmented media data **150** generated and maintained by the productivity and media presentation server system **110** is time-indexed, comprising a recording with a temporal or time-based dimension (e.g., corresponding to the duration of the recording and the duration of the recorded event) and media content for different points along the temporal dimension. In turn, the time-indexed media data has layers corresponding to the various different types of media content and metadata, such as video, audio, transcript text, translation text, presentation slides, meeting chats, screenshares, metadata, user-specified and/or automatically generated tags, user information (e.g., identifying current speakers and/or participants depicted visually), and/or user-specified notes, comments, and/or action items associated with different points along the temporal dimension. The layers can further include separate audio and video streams generated by each of the user devices **80** in the meeting. In general, the layers of the processed and segmented time-indexed media data stack or align with each other along the temporal dimension such that the media content provided on each of the different layers

have a common time-index with respect to the same points in time along the temporal dimension.

[0057] The time-indexed media data **150** stored by the productivity and media presentation system **100** preferably comprises several layers of different types of time-indexed content (e.g., video, audio, transcript text, translation text, presentation slides, metadata, user-specified and/or automatically generated tags, user information, and/or user-specified notes, comments, automations and/or action items) and/or of similar types (e.g., multiple different video or audio layers). In one example, multiple video layers of the media data are stored, each correspond to different encodings of essentially the same video stream. Similarly, multiple audio layers of the media data each correspond to different encodings of essentially the same audio stream. On the other hand, multiple layers of the media data can also each correspond to distinct content streams that are nevertheless indexed and synchronized by the temporal dimension such that the different layers for the different types of content depict the same recorded event, at the same points in time along the duration of the recording, but from different aspects.

[0058] For example, the time-indexed media data comprises multiple video or audio layers, each video layer corresponding to streams captured by different video and/or audio capture devices at different locations. Here in this example, one video layer provides media data captured by one video capture device at one location visually depicting one participant, while other video layers provide video content captured by other video capture devices at different locations visually depicting other participants. Still other video layers include video streams depicting a screenshare session that occurred during the recorded event.

[0059] The time-indexed media data also usually includes several audio layers corresponding to each of the different video layers providing audio data captured by audio capture devices at the respective locations and depicting the speech of the respective speakers that are often visually depicted in the video layers. Thus, the different video or audio layers are typically associated with particular individuals, and text and/or metadata layers then define an association between the different audio and/or video layers depicting different individuals with different users of the media presentation system.

[0060] In other cases, the video and audio of the several participants is a combined audio and video provided by the video conferencing system **12** in which the video of the separate participants is displayed in the different panes of each video frame.

[0061] These text and/or metadata layers often also are associated with different users depicted within the same audio and/or video layers by referencing different points of time along the temporal dimension for which the defined associations (e.g., tags) are applicable. The text and/or metadata layers also preferably include time-indexed information concerning user permissions, ownership, and/or access rights specified in permissions data stored by the system, including information associating users with various roles with respect to portions of the recording defined via time information specified for each association indicated in the layer of the media data. In one example, the stored permissions data establishes that users tagged via a text/metadata layer of the media data as having the role of "speaker" with respect to a recording or portions of a

recording (such as an individual that is depicted speaking at certain points in the audio and video layers or an individual that is considered a featured speaker for a portion of the recording in which other individuals also are depicted speaking, among other examples) should have edit and/or redaction rights for the portions within which they are tagged as a speaker.

[0062] Moreover, in addition to the layers discussed above, the time-indexed media data also typically includes layers for presentation content, including presentation slides showing different slides (e.g., of a PowerPoint slideshow or Slides from a G-Suite presentation) that were displayed during the recorded event at different points in time. Here, while one video layer visually depicts a presenter speaking, and one audio layer depicts the speech sounds from that presenter, a presentation slide or screenshare layer include time-indexed content for depicting the different slides (e.g., visually depicting the slides or portions of the slides via image data and/or providing actual text and/or formatting from the slides) or screenshare imagers or video along with timestamps specifying ranges of time for which the slides are applicable (e.g., corresponding to times when the slides were displayed during the event).

[0063] In any event, because the clips **212** include the recording references (e.g., start/stop times) delineating the extent of the clips with respect to the duration of the full recording **210**, and because the layers of the time-indexed media data stack or align with each other along the temporal dimension such that the content provided on each of the different layers are indicated with respect to the same points in time along the temporal dimension, any clips referencing a portion of the recording can potentially encompass all layers of the time-indexed media data within the time period specified by the clip **212** or a subset of the layers.

[0064] In addition to generally presenting streaming media content of the recordings, the user device **80**, via the graphical user interface **87** rendered on its display **84**, enables users to author content (e.g., static content that is not time-indexed), for example, using the page editor **90** (e.g., word processing web app, wiki platform) for inputting, editing, formatting, and outputting pages **150P** containing the user-authored content **150U** such as plain text, formatted text, still images, tables, charts, bulleted lists, and/or other display elements. The pages **150P** are viewed, created and/or edited by one or more users via the page editors **90** of one or more user devices, particularly via interface elements of the page editor **90** such as a text input box, a text formatting toolbar, and a cursor **95** indicating a current position for any incoming text input received by the user device such as via a keyboard.

[0065] Along with the user-authored content **150U**, the media presentation system enables users to embed clip data defining referenced portions of time-indexed content from an event (e.g., the recording and its associated time-indexed media data stored in the data store). In one embodiment, the media presentation system includes a user app **85** executing on the user devices **80**. This user app **85** renders the graphical user interface (GUI) **87** that includes the page editor **90** that enables the embedding of clip objects **212** representing the referenced portions of the time-indexed recording objects **210** into user-authorized multimedia documents **150P**.

[0066] In more detail, the embedded clip objects or clips **212** are displayed by the page editor **90** via clip display

elements **212D**, which reference content derived from the stored time-indexed media data (e.g., transcript text **228T**) pertaining to the referenced portion of the recording and a clip play button, among other examples. These clip display elements **212D** are rendered based on underlying page data for the displayed page, which includes the user-authored content itself (e.g., context-specific text entered by users) along with display data indicated via one or more markup languages (e.g., HTML and/or other wiki-related markup languages). Inserted into the underlying page data for the displayed page are clips **212** that are displayed or rendered as clip display elements **212D** for rendering the display elements of the embedded clips. The clip display data **212D** includes clip references, which are references to relevant clip data **212** and/or portions of the time-indexed media data **210** stored in the data store(s) **114** of the server system **110** (e.g., transcript text **228T** within the portion of the recording defined by the recording reference of the clip). In general, when initially loading a page to be displayed, the user device **80** first retrieves the page data **150P** for the page to be displayed and then retrieves relevant content derived from the time-indexed media data **210** based on any clip reference data of clip references extracted from the page data.

[0067] Clip display elements **212D** for embedded pages are generally formatted the same way as the user-authored content **150U** of the page **150P**, for example, having the same indentation level as any plain text around it and/or the same bullet and indentation level appropriate to its position.

[0068] Moreover, embedded clips **212** might have attributes (e.g., indicated in the clip data for the clip) that include which recording it came from, which speakers or participants were active in the clip, as well as other meta-information, all of which can be represented or hidden in the page editor **90** depending on the user's goals (e.g., based on user supplied or inferred display parameters).

[0069] The GUI **87** rendered on the display **84** user device **80** also includes a clip player **92**, which is a display element for streaming playback of the portions of the time-indexed media data referenced by the embedded clips **212**. In one example, the clip player **92** is first hidden and, in response to user selection of the clip play button **94** for an embedded clip, the clip player **92** is displayed overlaid on the page editor **90**, and the portion of the recording referenced by the selected embedded clip is streamed and presented.

[0070] More specifically, when the user app **85** loads a page, in addition to text-layers (e.g., transcript text) of the media data referenced by any embedded clips, the user app receives a playback descriptor or manifest including a playback token that, in general, grants access only to the referenced portion of the recording by describing ranges of media data the user is allowed to access. The user app stores the playback token and manifest in local memory of the user device and, in response to user selection of the clip play button for an embedded clip, uses the manifest to request the referenced portion of the recording and sends the playback token along with the request. The server system **110** determines whether the requested portion of the recording is authorized based on the playback token and, if so, streams the streaming media to the user device.

[0071] In general, the media presentation system allows the pages **150P** created by one user via the user app and page editor **90** to be shared with a plurality of other users with a variety of different permissions levels (e.g., view, edit). The page editor includes a share button **96**. In response to user

selection of the share button, the user app presents one or more additional interface elements (e.g., popup window with input elements) for receiving additional user selections indicating which users to share and/or which permissions to set for each of the indicated users. Any users granted access to a portion of the recording via an existing clip embedded within a page shared with them by another user (e.g., via the share button of the page editor presenting the page) can, in turn, share that same portion with other users by embedding a new clip based on the existing clip into one of their pages that they then share with others (e.g., via the share button of the page editor presenting the page). When embedding the new clip, the user can narrow the scope (e.g., the extent of the full recording referenced by the clip) of the new clip with respect to the existing clip, for example, by selecting only a portion of the transcript text of the embedded clip, copying the selected portion, and pasting the copied selection into the page editor for a page. However, when embedding a new clip from an existing clip, the user is prevented from expanding the scope beyond what was shared with them originally. For example, the inclusion of only the portion of transcript text pertaining to the embedded clip prevents selection of any portion outside of the displayed portion of the transcript. In one embodiment, an additional verification step is performed by the user app and/or the server system to confirm that any operation creating a new clip from an existing clip does not expand the scope of the new clip with respect to the existing clip.

[0072] In general, the media presentation system **100** also performs access control functionality at the level of full recordings. The access control and permissions for recordings are based on a group ownership scheme, in which any recording can have one or many owners that have full rights in accessing and modifying the recording. Any owners of a recording can add other owners for the recording (e.g., by selecting other users to add on the GUI) but are prevented from removing owners. In embodiments, the server system **110** initially sets owners of a newly added recording based on which user uploaded or imported the new recording, based on analysis of the new recording, and/or based on integration with a system or service that originally hosted the event depicted in the recording and/or generated the recording. Moreover, in different configurations, owners of recordings can correspond to different functional roles potentially played by users with respect to events and recordings of events, including users who added the recordings (as previously mentioned), users who were present at and/or contributed to the events, and/or users who are depicted in video or audio layers of the recordings, to name a few examples.

[0073] The media presentation system **100** allows redaction of portions of a recording, for example, based on permissions data and/or predetermined redaction control criteria (e.g., stored on the data store of the server system or in local memory of the user device). According to the permissions data and/or the redaction control criteria, the system allows owners (and only owners) of the recordings to redact the recordings, any owner of a recording can redact the recording, and any recording can be redacted by its owners at any time. In response to receiving a redaction request from the user app indicating a portion of the recording to be redacted, the server system modifies or deletes the media data for the indicated portion of the recording stored in the data store such that the redactions are reflected in any

clips referencing the portion of the recording that was redacted. Redactions can include deleting any layer (audio, video, text, or any combination thereof) and/or replacing the deleted layer(s) with blank frames and/or text indicating that the portion of the recording was redacted. In one embodiment, redaction is permanent. For example, in response to receiving a redaction request from the user app, the server system executes the redaction request by destroying or clearing all artifacts of the media data for the redacted portion of the recording from the data store.

[0074] FIG. 1B is a schematic diagram of the video conferencing and media presentation system **100** showing components of an exemplary user device **80-n**, the video conferencing system **12**, and productivity and media presentation server system **110** in additional detail and particularly how the system might be implemented in hardware.

[0075] In the illustrated example, a plurality of user devices **80** are connected to the video conferencing system **12** and productivity and media presentation server system **110** via the public network, such as the internet.

[0076] The media presentation server system **110** includes an app server **110A**, one or more media servers **110M**, usually an authentication module **110U**, a verification module **110V**, and one or more data stores **114**.

[0077] The productivity and media presentation server system **110** and its data store(s) **114** are typically implemented as a cloud system. In some cases, the server system **110** includes one or more dedicated servers having respective central processing units and associated memory. In other examples, they are virtual servers that are implemented on underlying hardware systems. The server system **110** may run on a proprietary or public cloud system, implemented on one of the popular cloud systems operated by vendors such as Alphabet Inc., Amazon, Inc. (AWS), or Microsoft Corporation, or any cloud data storage and compute platforms or data centers, in examples. The server system **110**, app server **110A**, and/or media server(s) **110M** can comprise or use various functions, modules, processes, services, engines, and/or subsystems. These various functions, modules, processes, services, engines, and/or subsystems, including the authentication module **110U** and verification module **110V**, and/or the app server and/or media server(s) themselves, are generally associated with separate tasks and can be discrete servers, or the separate tasks can be combined with other processes into a unified code base. They can be running on the same server or different servers, virtualized server system, or a distributed computing system. The server system **110** may also be implemented as a container-based system running containers, i.e., software units comprising a subject application packaged together with relevant libraries and dependencies, on clusters of physical and/or virtual machines (e.g., as a Kubernetes cluster or analogous implementation using any suitable containerization platform). Moreover, the user app **85**, app server **110A**, authentication module **110U**, verification module **110V**, transcription module **110T** and/or media server(s) **110M** can utilize or comprise various interacting functions, modules, processes, services, engines, and/or subsystems that are associated with discrete tasks, implemented as services or microservices running on different servers and/or a centralized server of the server system, and accessible by clients (e.g., user app executing on user devices, other services running on the server system).

[0078] The data store(s) **114** provide storage for the processed and segmented time-indexed media data **150** along with the clip data **212** for the clip objects, the page data **150P** for the different pages (e.g., including references to the clip data and segmented media data), workspace data **150W**, and/or user data **150US** used by the user app to present the different pages via the page editor and provide editing, collaboration, and sharing functionality for the different users. In addition, the data store(s) store authentication data **150A** for verifying user-supplied credentials and generating new login sessions for the users. The data store(s) also store permissions data **150M** for controlling access (e.g., reading and/or modifying) by users to pages, workspaces, and/or recordings (including media data). In one embodiment, the data store(s) are provided via a storage service accessed via a web interface, such as S3 provided by Amazon Web Services. In one example, newly ingested recordings, are stored as objects in an S3 bucket.

[0079] The app server **110A** provides an application programming interface (API) and handles requests from the user devices **80** (e.g., via the respective user apps **85** executing on those user devices) to retrieve and/or modify any of the page data **150P**, clip data **212**, workspace data **150W**, user data **150US**, and/or index data **150X**. The app server **110A** also generally handles ingestion processing of new recordings.

[0080] The media server(s) **110M** receive playback requests from the user apps **85** (along with possibly a playback token for authentication) and, in response, retrieve the time-indexed media data **150** for requested portions of full recordings (e.g., segments, portions of segments) from the data store(s) **114** and return the media data to the user device **80** (e.g., by generating playable media based on the retrieved media data and streaming the playable media to the user device). In one embodiment, the media server(s) **110M** and any data stores **114** storing the processed and segmented media data are implemented as a content delivery network (CDN), and the user app directs the playback requests to particular servers at particular addresses indicated in streaming manifests provided by the app server **110A**. In embodiments, the media server(s) user protocols, such as MPEG DASH or Apple HLS, are used to create playable pieces and stream them to the client.

[0081] In general, the authentication module **110U** retrieves the stored permissions data **150M** from the data store(s) **114** and generates signed cryptographic tokens identifying users and/or incorporating context-specific permissions data for the identified users. The tokens generated by the authentication module **110U** are sent to the user device **80**, which stores the tokens in local memory **82**. The tokens can include session tokens, which the user device includes with requests to the app server to retrieve and display page data **150P** and workspace data or modify data in the data store(s) such as permissions data, to list a few examples. The tokens can also include playback tokens, which the user device includes with playback requests to the media server(s) for streaming media data from the data store(s).

[0082] The verification module **110V** generally enforces access control with respect to incoming requests for any data stored in the data store(s), including page data **150P**, clip data **212**, and/or media data based on tokens provided with the requests and/or permissions data **150M** stored in the data store(s).

[0083] The user devices **80** are generally computing devices operated by users of the media presentation system **100**, and the system can accommodate many user devices **80** operated by different users at different times or simultaneously. The user device **80** will typically be a desktop computer, laptop computer, a mobile computing device such as a smartphone, tablet computer, phablet computer (i.e., a mobile device that is typically larger than a smart phone, but smaller than a tablet), smart watch, or specialized media presentation device to list a few examples. Each user device **80** includes a central processing unit **81**, memory **82**, a network interface **83** for connecting to the public network **90**, and a display **84**. Executing on the processor **81** is an operating system OS and a user app **85**, which generally receives user input (e.g., via input devices **66** such as a keyboard, mouse, and/or touchscreen, among other examples) indicating selections of pages to display via the page editor, changes to the pages, desired playback of recordings and/or clips, and new recordings to be ingested, to name a few examples. The user app **85** also receives from the server system **110** information such as page data **150P** including the clip data **212**, workspace data **150W**, user data **150US**, and/or index data **150X** for displaying the media data, page contents, the page editor **90**, and other interface elements on the display **84** via the graphical user interface **87**, which the user app **85** renders on the display **84**. In one example, the user app **85** executes within a software program executing on the processor **81** (via the operating system), such as a web browser, and renders specifically a browser user interface within a larger GUI **87** serving the user app **85**, web browser, and other applications and services executing on the processor **81** of the user device **80**. In another example, the user app **85** executes as a standalone software program executing on the processor **81** (via the operating system) and renders its own GUI **87** (e.g., in one or more windows generated by the standalone software application).

[0084] FIG. 2 is a sequence diagram illustrating an exemplary access control process performed by the video conferencing and media presentation system **100** using tokens.

[0085] In step **200**, the user app **85** presents a login interface to the user via the GUI **87**, and the user app receives user credentials (e.g., username, password) via the GUI in step **202**.

[0086] In step **204**, the user app **85** sends the user credentials to the authentication module **110U**, which retrieves from the data store(s) user data, authentication data, and/or permissions data **150M** pertaining to the user identified in the user credentials in step **206**. The authentication module **110U** then verifies the user credentials (e.g., based on the retrieved authentication data and/or user data), establishes a login session for the user (e.g., based on the retrieved user data and permissions data), and generates a session token in step **208**. The authentication module sends the session token to the user app in step **210**, which stores the session token in local memory of the user device in step **212**.

[0087] In step **214**, during normal operation during the login session, the user app **85** presents a user-specific platform interface (e.g., reflecting user's current workspace with the users' pages and recordings displayed) and receives selections indicating requests of various display and/or editing operations to be performed. In response, in step **216**, the user app generates various data access requests to perform various data access operations with respect to the

data stored by the data store(s) of the server system **110**, including retrieving, streaming, and/or modifying stored media segments. In step **218**, the user app sends the data access requests to the verification module **110V** along with the session token.

[0088] In step **220**, the verification module **110V** generates verification results based on the data access requests and the token. In one example, the verification module extracts user information, permissions data **150M**, and a signature from the session token, authenticates the extracted signature, and evaluates the data access request against the extracted permissions data **150M** to determine whether the user is authorized to perform the requested data access operation(s). In step **222**, the verification module **110V** executes the requested data access operation(s) based on the verification results by, for example, retrieving and/or modifying data stored in the data store(s) of the server system as requested in the data access request if the user is authorized or generating an error message if the user is not authorized. In step **224**, the verification module **110V** returns data access results for the data access operation(s) performed in step **222**, including, for example, requested data stored in the data store(s) of the server system or confirmation of changes to the stored data (if the user is authorized) or relevant verification information such as an error message (if the user is not authorized).

[0089] In this way the media presentation system provides basic access control functionality with respect to users' access to various types of data stored in the server system's data store(s) **114**.

[0090] FIG. **3** is a sequence diagram illustrating more detailed examples of how permissions data **150M** is set, changed, and used by the media presentation system **100** in controlling access to the stored data.

[0091] In general, throughout the following examples, the verification module **110V** processes and verifies the various incoming requests in the manner described with respect to the data access requests discussed in FIG. **2**. However, for the sake of clarity, only the details of the verification process relevant to the respective examples are explicitly described.

[0092] In one scenario, in step **300**, the user app **85** uploads a new recording of a video conference or any other meeting or event to the app server **110** for ingestion processing or the app server **110** loads the new recording according to its own rules. In response, in step **302**, the app server **110** stores data pertaining to the newly uploaded recording to the data store(s) **114**, which includes setting initial permissions data for the newly uploaded recording, namely setting the current user that uploaded the recording as having an "owner" status with respect to the recording. In addition, the app server **110** in some examples adds any participants to the meeting as additional owners.

[0093] In another scenario, in step **304**, the user app **85** sends a permissions change request along with a session token to the verification module **110V**, which, in response, determines whether the permissions change request is authorized, and, if authorized, updates the stored permissions data **150M** as requested in step **306**. In one example, the verification module **110V** sets a specified user as having an "owner" status with respect to a specified recording only in response to determining that the current user requesting the change in permission also has an "owner" status with respect to the specified recording. On the other hand, in another example, the verification module **110V** removes an "owner"

status from a specified user with respect to a specified recording only in response to determining that the specified user is the current user that is requesting the change. In this way, the media presentation system **100** ensures not only that only owners can add other owners to a recording but also that all owners of a recording have the same rights with respect to the recording.

[0094] In another scenario, in step **308**, the user app **85** sends a request to create or modify an embedded clip **212** along with a session token to the verification module **110V**, which, in response, determines whether the request is authorized based on the permissions data **110M** (e.g., stored and/or provided with the session token) and, if authorized, stores data for the created or updated clip. For example, before creating or updating a clip, the verification module **110V** determines that the scope of the new or updated clip is within the permitted range for the requesting user (e.g., based on the range of an existing clip from which the new clip is being created). Thus, in this way, the media presentation system **100** performs a back-end check to prevent newly created clips from expanding in scope with respect to parent clips from which they are spawned.

[0095] In another scenario, in step **312**, the app server **110A** sends a search request and token to the verification module **110V**, which, in step **314**, performs the search (e.g., by accessing the stored data, including index data) with respect to workspaces, pages, clips, recordings, and/or media data segments for which searching is permitted for the current user based on the permissions data. Thus, in this way, the media presentation system makes sure that only media data referenced by clips embedded in pages to which the user has access can be searched.

[0096] FIG. **4** is an illustration of an example of the processed and segmented media data that is stored in the data store(s) **114** upon ingestion, showing how permissions data is stored.

[0097] In general, the recording object **210** has permissions data **210AC** (e.g., an access control list or list of owners) and contains or is associated with five different media data segments 1-5 corresponding to successive portions of the original full recording. For each of the segments **150n**, there are multiple bitstream layers, including three video stream layers **230**, **232**, **234**, (Encoding A, Encoding B, Encoding C), two audio stream layers (Encoding 1, Encoding 1) **236**, **238**, and a text stream layer **240** such as a transcript of the event.

[0098] Additional time-indexed information is typically stored in the text stream layer **240**. This includes a transcription of the audio, translations, speakers (i.e., identifying the speaker who spoke the words in the transcription), and tags, comments, other annotations, and chat messages, among some examples. In some examples, this additional information is contained in separate time-indexed layers.

[0099] Within each of the segments, there are video, audio, and text segments corresponding to the respective layers **230**, **232**, **234**, **236**, **238**, **240**. In the illustrated example, the access control list for the recording object indicates that both user Dave and user Erin have an "owner" status with respect to the recording.

[0100] Referencing portions of the recording object are two clip objects **212**, each of which has an "embedded" relationship with a page object **150P** (although only one page object is shown in the illustrated example for the sake of clarity). Each of the clip objects inherits an access control

list **210AC** from the page object **150P** in which the respective clip is embedded. In the illustrated example, the first clip object has an access control list indicating that user Dave has “Admin” permissions, user Alice has “Read” permissions, and user Charlie has “Read” permissions with respect to the first clip. Accordingly, the recording object **210** included in the illustrated example has the same access control list, since the first clip inherits its access control list from the depicted page object **150P**. The second clip has an access control list indicating that both users Bob and Charlie have “Read” permissions with respect to the second clip.

[0101] Here, because user Dave is an owner of the recording object, the user can read, modify, redact, and share all segments of the recording object and add other users as owners of the recording, which is also true of user Erin. User Dave can also modify the contents of the page in which the first clip is embedded.

[0102] Because the first clip object references Segment **1**, Segment **2**, and Segment **4**, users Alice and Charlie can both view media data for these segments and the layers within each segment and share them with other users (e.g., by copying and pasting from the clips embedded in the page object). Similarly, because the second clip references Segment **4** and Segment **5**, users Bob and Charlie can both view media data for these segments and share them with other users. However, because none of them is indicated as owners of the recording **210**, none of them can redact these segments or any other segments of the recording. Also, because they each only have “read” permissions for the respective page objects, they can only view the pages and cannot edit any of its contents.

[0103] FIG. 5A is an illustration of an exemplary recording screen **228** of the GUI **87**.

[0104] In general, the GUI is rendered by the user app **85** and displayed on the display **84** of the user device **80** and includes a series of screens or views, which comprise graphical elements (such as icons, virtual buttons, menus, textual information) arranged within windows and/or panes. In response to detecting input from the user indicating interaction by the user with the graphical elements, the user app **85** receives input indicating selection of various options or functions represented by the graphical elements.

[0105] More particularly, the GUI comprises a home pane, a page navigation pane **220**, a recordings pane **222**, and a main display pane **224**, which is either a recording display pane or a page display pane.

[0106] The home pane includes a recordings button, upon selection of which the GUI shows the recordings pane.

[0107] The page navigation pane **220** includes a selectable page directory arranged in a hierarchical fashion allowing nested groups of pages to be expanded (e.g., revealed) or collapsed (e.g., hidden) in a shared pages section or a private pages section. The page navigation pane also includes add shared pages buttons **220A** and add private pages button **220B**, which, at the root level, are always displayed but at other levels of the hierarchy are only displayed when one of the pages indicated in the hierarchy is currently selected or hovered over (e.g., via a pointer of a mouse device).

[0108] The recordings pane comprises an upload button **222A** and an indication of the recordings **222B** (e.g., stored in the data store(s)) for which the current user is indicated as the owner. Upon selection of any of the recordings indicating in the recordings pane, a recording viewer is displayed in the main display pane.

[0109] In general, the recording viewer presents information about a recording, including textual information indicating the recording’s owners, meeting date, and duration, and transcript text **228T** for the recording. The recording viewer comprises an add owner button **228A**, a recording date selector **228B**, and an add tag button **228C**, selection of which allow the user to enter or change the respective information associated with the button/selector. The recording viewer also comprises a recording player **92**, with selectable playback buttons associated with playback of the media data for that recording, which is streamed by the media server(s) to the user device when the user selects the play button on the recording player, for example.

[0110] FIG. 5B is an illustration of an exemplary add owner window **270** of the GUI. The user app **85** displays the add owner window overlaid on the recording screen **228** in response to detecting selection of the add owners button **228A** of the recording screen. The add owner window includes a user selector, which is an input element for receiving the user’s selection of one or more other users to add as owners, and a send invites button. In response to detection of selection of the send invites button, the user app updates the permissions data **150M** stored in the data store(s) of the server system **110** to indicate that the users selected via the user selector are owners of the recording currently presented on the recording screen.

[0111] In general, FIGS. 6A-6C are illustrations of an exemplary invite users window. The user app **85** displays an invite users window **272** in response to detecting selection of an add users input element **220A**, **220B** on the manage workspaces screen **274** and/or a share button on the page editor. In the illustrated example, the invite users window is overlaid on a manage workspaces screen of the GUI. The invite users window includes a user selector similar to that described with respect to FIG. 5B and a permissions selector, which is an input element for receiving the user’s selection of a permissions level for the users selected via the user selector. In FIG. 6A, the invite users window has been displayed, with no selections indicated. FIG. 6B shows the invite users window of 6A with the permissions selector, which is a drop menu, expanded to show more permissions level options, including a guest level for view-only access and a user level for modify access. FIG. 6C shows the permissions selector scrolled down to reveal an admin level for modify access at an administrator level and owner access, adding the ability to remove users, add other owners, and delete entire workspaces. The invite users window also includes an invite button. In response to detection of selection of the invite button, the user app updates the permissions data **150M** stored in the data store(s) **114** of the server system **110** to indicate that the user(s) selected via the user selector should be added to the workspace currently presented in the manage workspaces screen or to the page currently presented in the page editor (not illustrated) at the permissions level(s) indicated via the permissions selector.

[0112] FIG. 7 is a sequence diagram illustrating an exemplary access control process for streaming media.

[0113] First, in step **700**, the user app **85** receives selections from the user indicating display of a page with embedded clips. For example, a user selects a name of a page indicated in the page navigation pane **220** of the GUI **87**, the selected page including embedded clips.

[0114] In step **702**, the user app **85** retrieves from the data store of the server system **110** the stored page data for the

page using a session token (e.g., via the app server and verification module). Here, the process may be similar to that described with respect to steps 216 through 224 of the process depicted in FIG. 2.

[0115] In step 704, the user app 85 extracts any clip references 212 (e.g., specially formatted clip reference strings and/or objects) from the retrieved page data 150P. Generally, the clip references identify clip objects stored in the data store(s) 114 of the server system, but they also preferably include any recording references, including recording IDs identifying the recording referenced by the clip and/or range data (e.g., start and stop times) delineating the referenced portion of the full recording. Based on these extracted clip references, the user app requests from the app server media metadata (e.g., thumbnail images, transcript text to be incorporated into clip display elements) for the portion of the full recording referenced by the extracted clips in step 706.

[0116] In step 708, the app server 110A sends user information identifying the user (e.g., a user ID, or session token) to the authentication module 110U.

[0117] In step 710, the authentication module 110U retrieves permissions data 150M pertaining to the user identified by the user information from the data store(s) 114 and, in step 712, generates a signed playback token based on the permissions data. The playback token generally indicates which particular segments of stored media data are authorized for the user to access and is used to prevent access to data outside of the permitted range. In one example, the playback token is a JSON web token (JWT). The authentication module 110U returns the playback token to the app server in step 714.

[0118] In step 716, the app server 110A retrieves the requested metadata for the segments indicated as authorized by the playback token. Based on this retrieved metadata, in step 718, the app server 110A generates a streaming manifest, which generally provides information about the media segments available to the user, including timing information, address information such as URLs, and characteristics of the media such as video resolution and/or bit rate information). In one example, the streaming manifest is specifically a media presentation description (MPD) according to the Dynamic Adaptive Streaming over HTTP (DASH or MPEG-DASH) protocol, and the JWT playback token is embedded in the MPD file. In another example, the streaming manifest and/or playback token might be provided in the form of signed URLs. In step 720, the app server 110A returns the streaming manifest and retrieved metadata to the user app.

[0119] In step 722, the user app 85 then presents the requested page to the user via the GUI 87 on the display of the user device 80 with the embedded clips based on the retrieved page data 150P and media metadata. For example, the user app renders the page based on the page data and retrieved media metadata such that the page contents (e.g., static web content such as text and images) are indicated within the page editor of the GUI according to previously entered text, formatting input, and other input. Inserted throughout the page contents in particular positions based on positions of the extracted page clip references 212 with respect to the rest of the page contents are clip display elements 212D, which include clip play buttons and expandable and collapsible text display elements incorporating the retrieved media metadata (e.g., transcript text) for the

respective portions of the full recording referenced by the embedded clips. The user app also stores the streaming manifest, including the playback token, in local memory of the user device 80.

[0120] In step 724, the user app 85 receives selections such as the user selecting a play button 94 associated with a clip display element 212D indicating playback of an embedded clip 212. For example, the user app might detect via the GUI selection of a clip play button of a clip display element, in response to which the user app displays a clip player for presenting streaming media data for the portion of the full recording referenced by the embedded clip for which the clip play button was selected. In response, in step 726, based on the user selections, the streaming manifest, and/or other factors (e.g., current computing device, application, or network context), the user app generates a streaming request that incorporates the playback token that was received from the user app and stored in local memory of the user device. The user app sends the streaming request to one of the one or more media servers in step 728.

[0121] In step 730, in response to the playback request, the media server(s) 110M retrieve media data for the segments to be streamed as requested in the playback request from the data store(s) 114 via the verification module 110V. Here, again, the process may be similar to that described with respect to steps 216 through 224 of the process depicted in FIG. 2, as the verification module 110V confirms based on the playback token that the requested segments are authorized for the requesting user and grants or denies access to the requested media data segments based on whether permissions data 150M of the playback token indicates that the requested segments are authorized. In step 732, the media server(s) 110M return the requested media data for the segments to be streamed to the user app 85, if the requested segments are authorized for the user. If the requested segments are not authorized, the media server(s) might return an error message to the user app instead.

[0122] Finally, in step 734, the user app 85 presents streaming media to the user via the GUI 87 and display of the user device, for example, via the clip player 92 activated by selection of the clip play button 94 associated with the clip display element 212D.

[0123] FIG. 8 is a sequence diagram illustrating an exemplary process for providing redaction of processed and segmented media data 150 stored in the data store(s) 114 of the server system 110.

[0124] First, in step 800, the user app 85 presents a redaction interface to the user via the GUI and display. In general, the redaction interface presents information about recordings stored in the data store(s) for which the current user is indicated to have an "owner" status according to the permissions data stored in the data store(s) and allows selection by the user of portions of the recording, including particular ranges and/or layers (e.g., video, audio, text) to redact or not redact.

[0125] In step 802, the user app 85 receives selections indicating desired redactions of a full recording 210. For example, the user app may receive start and stop time selections representing an extent along a time dimension of the time-indexed media data of the portion to be redacted and/or selections representing one or more layers to be redacted at each point along the extent defined by the start and stop time selections. The user app, in step 804, generates a redaction request based on the user selections and sends

the redaction request to the verification module **110V** in step **804**. In one example, the redaction request includes a recording ID, redaction range and layer data indicating the portions of the recording to be redacted, and a session token identifying the current user (and/or providing permissions data for the current user).

[**0126**] In step **806**, the verification module **110V** determines the portions of the recording to redact based on the redaction request and determines whether the redaction request is authorized. Here, the verification process may be similar to that described with respect to steps **216** through **224** of the process depicted in FIG. **2**. If the redaction request is authorized (e.g., based on a session token provided with the redaction request and/or stored permissions data), the verification module performs the redaction by, for example, deleting the stored media data pertaining to the indicated portions of the recording in the redaction request, replacing deleted media data with blank frames in the case of video and silence or a predetermined sound in the case of audio and/or text indicating that the media was redacted, permanently destroying or clearing from the data store(s) all relevant artifacts of the media data including the corresponding transcript for the redacted portion of the recording from the storage medium storing the media data, and/or updating index data based on the redaction such that redacted media is not reflected in search results.

[**0127**] In different examples, at step **806**, controlling access to the recording for redaction or any other modification can be based on ownership (e.g., whether a user is set as an owner of a recording) as well as other attributes indicated in the stored permissions data **150M**, user data, and/or media data. For example, the verification module **110V** allows redaction of any part of a full recording for any user set as an owner of the recording, in some implementations. In another example, the verification module allows redaction of only portions of the recording for an owner or user (e.g., only parts of the meeting where the owner or user was talking or presenting). Additionally, for recordings that comprise multiple streams recorded by different audio and/or video capture devices, the verification module **110V** allows redaction for an owner or user of an individual stream associated with the owner or user (e.g., the stream captured by the owner or user's audio and/or video capture device). The media presentation system **100** also provides an option to undo redactions within a predetermined period of time and/or delay full deletion and/or destruction of the stored data until after a predetermined age-out period (e.g., 30 days), according to some implementations.

[**0128**] FIG. **9** is an illustration of an example of the processed and segmented media data depicted in FIG. **4** after being redacted. As before, the recording object **210** incorporates five segments **150n** of media data, each with various video, audio, text segments at respective layers **230**, **232**, **234**, **236**, **238**, **240**. Often the text layer **240** is a transcript derived from the audio layers **236**, **238**. In other example, slides or other images and/or screenshares for video of a presentation are provided in other layers or as part of the text layer. Clip objects **212** representing clips embedded within pages refer to the various segments. Now, however, the second segment has had all of its layers redacted **280A**, and the fourth segment has had only its video layers redacted but not its audio or text layers **280B**. In one example, instead of the redacted video layers including frames of image data, the redacted video layers now include image data that are

displayed as blank frames including text indicating that the media was redacted, instead of the audio layers including audio data from the recording, the redacted audio layers include no sound or previously specified audio data including speech stating that the media has been redacted, and instead of the text layers including transcript text from the recording, the redacted text layers include only text indicating that the media was redacted. In the illustrated embodiment, portions spanning entire segments of the segmented media data are redacted. However, in other embodiments (not illustrated), portions within the segments themselves (e.g., not the entire segments) can be redacted.

[**0129**] In general, FIGS. **10A-10C** are illustrations of exemplary page editor screens of the GUI, showing how media data for clips embedded in pages is presented for different segments of the stored redacted media shown in FIG. **9**. Here, the GUI includes the page editor **90** with a clip display element **212D** for an embedded clip **212**, the clip display element including transcript text **228T**. Additionally, the clip player **92** is overlaid on the page editor screen and presents streaming media playback for the embedded clip. More particularly, FIG. **10A** shows how the first, unredacted segment from the example of FIG. **9** would be presented. As the segment is unredacted, both the transcript text **228T** and the video frames in the player **92** reflect the original media data for the recording. On the other hand, FIG. **10B** shows how the second segment from FIG. **9**, with all of its layers redacted, would be presented. Here, the clip display element **212D** shows text indicating that the media was redacted instead of the transcript text, and the clip player shows blank video frames with text indicating that the media was redacted instead of the original video frames of the recording. FIG. **10C** shows how the fourth segment from FIG. **9**, with only its video layers redacted, would be presented. Now, the transcript text is displayed, but the clip player **92** shows the blank video frames.

[**0130**] FIG. **11** is a flow diagram illustrating an exemplary automation process for newly ingested recordings.

[**0131**] First, in step **1100**, the media presentation system **100** receives a newly ingested recording.

[**0132**] In step **1102**, the media presentation system (e.g., the app server **110A** of the server system **110**) scans audio data of the recording for occurrences of any predetermined explicit or implicit trigger words and/or keywords and generates workflow actions to be performed based on the detected trigger words and/or keywords. In one example, the system detects an explicit trigger word for creating an action item (e.g., a user stating a predetermined phrase "create action item") and generates an item for an action item or to-do list associated with one or more users. In another example, the system detects an implicit trigger phrase for sharing the recording or portion of the recording to (e.g., a user stating naturally in context "invite Alice" or "get input from Bob on this") and generates a prompt to share the recording or portion of the recording (e.g., a predetermined period of time before and after occurrence of the trigger, only a discrete portion in which the current speaker who stated the trigger is speaking) with another user. In another example, the system detects occurrences of predetermined keywords and generates tags to assign to the media metadata at the points where the keywords occurred based on the detected keywords.

[**0133**] In step **1104**, the system identifies speakers depicted in the audio data of the recording based on stored

user data, which may include audio fingerprint data configured and stored for each user, and updates stored media data for the recording to indicate the detected user as the speaker at each frame of media data where the user's voice is detected. The system also updates the stored permissions data for the recording and/or media data based on the detected speakers by, for example, setting detected speakers as owners of the recording and/or giving detected speakers read, modify, and/or redact permissions to segments of the recording where they are detected. Speakers can be identified 'manually' by the owner of the media data **210**. In a preferred embodiment, speaker identification is performed by speech recognition process executed by the transcription module **110T** when the media data is chunked into the segments for storage.

[0134] In step **1106**, the system retrieves a transcript/translation revision history relevant to current users and/or workspaces and automatically applies corrections to transcripts for newly ingested recordings based on the transcript revision history. Here, for example, the media presentation system can use machine learning and/or artificial intelligence to learn specialized terms used by particular groups of users or within particular workspaces and calibrate its automatically generated transcriptions to account for these specialized terms that might not otherwise be detected during speech recognition.

[0135] FIGS. **12A-12C** are illustrations of exemplary page editor screens **242** of the GUI **87**, showing how portions of the recording of the recording object **210** are redacted and deleted via the interface.

[0136] Here, the GUI includes the page editor **90** with a clip display element **212D** for an embedded clip **212**, the clip display element including transcript text **228T**. Additionally, the clip player **92** is overlaid on the page editor screen and presents streaming media playback for the embedded clip.

[0137] As shown in FIG. **12B**, the user highlights **289** the text that they desire to redact and then invoke the tool tip **290** by selecting the right mouse button for example. The tool tip provides the redact function. In this example, the user is redacting the highlighted portion of the clip.

[0138] The tool tip **290** further enables the user to redact all or specific layers. For example, the GUI enables the user to choose to redact only the "video," only the "audio," only the "transcript" or still "other" layers individually, in addition to redacting "all" layers. This allows the user to selectively redact layers if they want. For example, a user might want to just redact their camera view if their cat walked past the camera but would be happy keeping other aspects of the data (audio, presentation, etc).

[0139] FIG. **12C** shows the effect of the redaction; the previously highlighted text is replaced with a REDACTED message **292**. The player **92** also provides a redacted message rather than the original video from the event.

[0140] While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

What is claimed is:

1. A method for controlling access to recordings of events, the method comprising:

storing in a data store time-based media data for a recording of an event and permissions data indicating access permissions associated with different portions of the recording; and

controlling access to requested portions of the stored media data based on the stored permissions data associated with the requested portions of the recording.

2. The method of claim 1, further comprising presenting pages with embedded clip objects and controlling access to requested portions of the stored media data based on permissions data associated with pages with embedded clip objects that reference the requested portions of the stored media data.

3. The method of claim 2, further comprising controlling access to the requested portions of the stored media by preventing users from expanding a scope of new and/or modified clip objects beyond that of previously existing clip objects shared with and/or accessible by the users.

4. The method of claim 1, further comprising controlling access to requested portions of the stored media data by generating a playback token for each user indicating portions of the recording that are allowed for that user based on the stored permissions data and granting or denying requests from users for portions of the stored media data based on validation of playback tokens included with the requests.

5. The method of claim 1, further comprising receiving tag information indicating portions of the recordings to tag and users associated with the indicated portions of the recordings and updating the stored media data corresponding to the indicated portions of the recordings to include tags indicating the users associated with the indicated portions of the recordings based on the tag information.

6. The method of claim 5, further comprising providing a tagging interface for receiving selections indicating the portions of the recordings to tag and the users associated with the selected portions of the recordings and generating the tag information based on the received selections.

7. The method of claim 5, further comprising controlling access to the stored media data based on the tags of the stored media data.

8. A system comprising user devices enabling users to access the stored media data and a server system for storing the time-indexed content into the data store for implementing the method of claim 1.

9. A method for group ownership of recordings referenced in a media presentation system, the method comprising:

storing in a data store time-based media data for recordings of events and ownership information indicating one or more owners for each of the recordings;

controlling access to the stored media data for the recordings and generation of references to portions of the recordings based on the ownership information for the recordings; and

restricting changes to the ownership information for the recordings based on current ownership information for the recordings and predetermined group ownership rules.

10. A system comprising user devices enabling users to access the stored media data and a server system for storing the time-indexed content into the data store for implementing the method of claim 9.

11. A method for redaction of recorded media in a media presentation system, the method comprising:

storing in a data store time-based media data for recordings of events;
 providing a redaction interface for receiving selections indicating redactions to the stored media data;
 redacting the stored media data in response to the received selections indicating the redactions; and
 restricting access to the redacted media data in response to requests for the stored media data.

12. The method of claim **11**, further comprising controlling access to the stored media data by allowing or restricting redaction of the stored media data for the recordings based on whether current users are indicated as owners of the recordings.

13. The method of claim **11**, wherein redacting the stored media data in response to the received selections indicating the redactions comprises receiving selections of portions of the recording to be redacted, and redacting the stored media data in response to the received selections comprises redacting stored media data only for the selected portions of the recording to be redacted.

14. The method of claim **13**, wherein the received selections include selections of particular video, audio, text, and/or metadata layers for the portions of the recording to be redacted, and redacting the stored media data in response to the received selections comprises redacting only the particular video, audio, text, screenshare and/or slide layers indicated for redaction.

15. The method of claim **11**, further comprising redacting the stored media data by deleting the stored media data pertaining to the indicated portions of the recording and/or replacing the deleted media data with blank frames and/or text indicating that the media was redacted, a user who made the redaction, and/or time information for the redaction.

16. The method of claim **11**, wherein redacting the stored media data comprises updating stored index data such that the redactions are reflected in search results.

17. The method of claim **11**, further comprising redacting the stored media data by destroying or clearing all artifacts of the media data for the redacted portion of the recording from the data store.

18. The method of claim **11**, further comprising identifying speakers in the stored media data by a speech recognition process and allowing speakers to redact portions of the media data in which they were detected as speakers.

19. A system comprising user devices rendering the redaction interface for users and a server system for managing the time-indexed content in the data store for implementing the method of claim **11**.

20. A method for group ownership of recordings referenced in a media presentation system, the method comprising:

storing in a data store time-based media data for recordings of events and ownership information indicating one or more owners for each of the recordings;

controlling access to the stored media data for the recordings based on the ownership information for the recordings; and

restricting changes to the ownership information for the recordings based on current ownership information for the recordings and predetermined group ownership rules.

21. A method for analyzing recordings of events, the method comprising:

storing in a data store time-based media data for a recording of an event;

analyzing the time-based media data for trigger words and/or keywords; and

generating workflow actions to be performed based on the detected trigger words and/or keywords.

22. The method of claim **21**, further comprising generating an item for an action item or to-do list for the workflow actions.

23. The method of claim **21**, further comprising generating tags to assign to the media metadata at points where the keywords occurred based on the detected keywords.

* * * * *