

TECHNIQUES FOR AUTOMATICALLY PERSONALIZING MEDIA VIEWING EXPERIENCES

CROSS-REFERENCE TO RELATED APPLICATIONS

5 **[0001]** This application claims benefit of United States patent application serial number 18/169,755 filed February 15, 2023, which is hereby incorporated herein by reference.

BACKGROUND

Field of the Various Embodiments

10 **[0002]** The various embodiments relate generally to computer science and to media streaming technology and, more specifically, to techniques for automatically personalizing media viewing experiences.

Description of the Related Art

15 **[0003]** In a process known as “video localization,” an original video source file associated with a given media title is modified to generate a localized video source file that conforms more to the individual preferences of a target audience. For example, English text that appears in an original video source file could be replaced with comparable French text to generate a localized video source file that conforms more to the preferences of a French-speaking target audience. In another example, a hockey scene in an original video source file could be replaced with a soccer scene to
20 reflect the sporting preferences of a target audience that lives in Brazil.

25 **[0004]** A typical media title can be streamed and played back via one or more presentations, where each presentation includes a video track, an audio track, and an optional timed text track that are derived, respectively, from a different video source file, a different audio source file, and a different optional timed text source file. Each track specifies either a single stream or multiple streams that usually correspond to different bitrates, where a given stream constitutes an encoded version of a source file.

30 **[0005]** To enable a client device to stream and playback different presentations of a media title, a media processing pipeline is used to independently generate a different video-specific package for each video source file associated with the media file. To generate a video-specific package for a given video source file, one or more

target presentation(s) are defined based on the preferences of target audiences and compatibility with the corresponding video source file. The audio source file(s) and any optional timed text source file(s) required to generate the target presentations(s) are identified. For each of the single video source file, the one or more identified
5 audio source file(s), and the zero or more identified timed text source file(s), the media processing pipeline generates a different set of one or more corresponding streams, referred to as a "stream set." The media processing pipeline then generates a video-specific package that includes the stream sets, stores the video-specific package in an original server device, and then deploys the video-specific package
10 from the origin server device to a content delivery network.

[0006] To playback the media title, an endpoint application executing on a client device requests a manifest file for the media title from cloud-based media services. In response to the manifest file request, a manifest application selects an edge server within the content delivery network that is proximate to the client device. The
15 manifest application then selects a video-specific package for the media title based on one or more characteristics associated with the user (e.g., a profile language). A profile language for a user specifies the language of user interfaces that the streaming media provider uses to interact with the user. Subsequently, the manifest application generates at least one recommended presentation based on the single
20 video stream set, the audio stream set(s), and the optional timed text stream set(s) included in the selected video-specific package. Each recommended presentation includes the same video track that specifies a subset of the video stream set and a different combination of an audio track that specifies a subset of an audio stream set and an optional timed text track that specifies a subset of a timed text stream set.
25 The manifest application generates and transmits a manifest file to the endpoint application, where the manifest file enables the endpoint application to determine the selected edge server and select the tracks in any recommended presentation. To playback each portion or "segment" of the media title, the endpoint application selects a corresponding segment from a stream in each selected track and requests the
30 selected segments from the selected edge server device. In some implementations, the endpoint application can switch between two or more recommended presentations during the playback of the media title.

- [0007]** One drawback of the above approach is that each audio source file and each timed text source file used to generate more than one video-specific package ends up being redundantly processed by the media processing pipeline. As a result, unnecessary amounts of storage resources, processing resources, and time are consumed in generating and deploying the different video-specific packages to the content delivery network. For example, with the above approach, if a Spanish audio source file were included in five different video-specific packages, then the Spanish audio source file would end up being encoded five different times at one or more different bitrates to generate five separate audio stream sets, and all five audio stream sets would include the same audio stream(s). The five audio stream sets also would be separately assessed for quality and separately deployed to the content delivery network. In such a scenario, storage resources, processing resources, and time are unnecessarily consumed in generating, processing, and deploying the audio streams associated with four of the five video-specific packages.
- [0008]** Another drawback of the above approach is that the number of different recommended presentations of a media title made available to each user can be unnecessarily limited. In that regard, the recommended presentations specified in a given manifest file are generated based on the subset of the stream sets associated with the media title that are actually included in one video-specific package. In other words, the recommended presentations are generated based on the single video stream set, the one or more audio stream sets, and the zero or more timed text stream sets that are included in the selected video-specific package. Accordingly, if the preferences of a given user differ from the preferences of the target audience(s) associated with the selected video-specific package, then the quality of the viewing experience for the given user is likely to be subpar. For example, suppose that a video-specific package for a media title included a Spanish video stream set, an English original audio stream set, and a Spanish dubbed audio stream set, but omitted a Polish dubbed audio stream set. The manifest file generated for a user associated with a profile language of Spanish and an audio preference of Polish would include one recommended presentation specifying both a Spanish localized video track and a Spanish dubbed audio track and another recommended presentation specifying both the Spanish localized video track and an English original audio track. Because the manifest file would not allow the user to access the Polish

dubbed audio stream set preferred by the user, the quality of the viewing experience for the user would most likely be reduced.

[0009] As the foregoing illustrates, what is needed in the art are more effective techniques for tailoring media viewing experiences based on user preferences.

5 **SUMMARY**

[0010] One embodiment sets forth a computer-implemented method for generating and distributing streams for playback. The method includes encoding a first video source file that is associated with a media title to generate a first set of video streams; encoding a second video source file that is associated with the media title to generate
10 a second set of video streams; generating first video metadata associated with the first set of video streams that includes a first set of values for a set of video characteristics; generating second video metadata associated with the second set of video streams that includes a second set of values for the set of video characteristics; generating a media package that includes the first set of video streams, the first video
15 metadata, the second set of video streams, and the second video metadata; and transmitting the media package to at least one server device.

[0011] At least one technical advantage of the disclosed techniques relative to the prior art is that, with the disclosed techniques, source files are not redundantly processed by the media processing pipeline when generating and deploying a single
20 inclusive media package that more comprehensively represents a media title to a content delivery network relative to conventional video-specific packages. Therefore, the amounts of storage resources, processing resources, and time consumed in generating and deploying a media title can be substantially reduced relative to prior art approaches. Another technical advantage of the disclosed techniques is that,
25 unlike prior art approaches, the tracks specified in a manifest file generated using a single inclusive media package can be selected on-the-fly from all stream sets derived from all source files associated with a corresponding media title based on the preferences of a given user. Therefore, the disclosed techniques are able to generate tracks for playback that more closely align with the preferences of users relative to
30 what can be achieved using prior art approaches where manifest files are generated based on a subset of streams included in a single video-specific package. These

technical advantages provide one or more technological advancements over prior art approaches.

BRIEF DESCRIPTION OF THE DRAWINGS

5 **[0012]** So that the manner in which the above recited features of the various embodiments can be understood in detail, a more particular description of the inventive concepts, briefly summarized above, may be had by reference to various embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical embodiments of the inventive concepts and are therefore not to be considered limiting of scope in any
10 way, and that there are other equally effective embodiments.

[0013] Figure 1 is a conceptual illustration of a system configured to implement one or more aspects of the various embodiments;

[0014] Figure 2 is a more detailed illustration of the inclusive media package of Figure 1, according to various embodiments;

15 **[0015]** Figure 3 is a more detailed illustration of the manifest file of Figure 1, according to various embodiments;

[0016] Figure 4 is a more detailed illustration of the alternate manifest file of Figure 1, according to various embodiments;

20 **[0017]** Figure 5 is a flow diagram of method steps for generating and deploying streams that are associated with a media title to a content delivery network, according to various embodiments;

[0018] Figure 6 is a flow diagram of method steps for generating a manifest file that itemizes streams that are associated with a media title, according to various embodiments; and

25 **[0019]** Figure 7 is a flow diagram of method steps for generating manifest files that enable video tracks to be switched during playback of a media title, according to various embodiments.

DETAILED DESCRIPTION

[0020] In the following description, numerous specific details are set forth to provide a more thorough understanding of the various embodiments. However, it will be apparent to one of skilled in the art that the inventive concepts may be practiced
5 without one or more of these specific details.

[0021] In video localization, an original video source file is modified to generate one or more localized video source files that conform more to corresponding target audiences. For example, lip reanimation techniques can be used to modify the lip movements of speakers in a movie produced in English to align with dubbing for the
10 movie in thirty languages to generate thirty localized versions of the movie. To enable client devices to stream and playback a media title, a typical media streaming service independently generates and distributes a different video-specific package each different video source file associated with the media title.

[0022] To generate a video-specific package for a given video source file that is
15 associated with a given media title, a subset of the audio source files associated with the media title, and a subset of the timed text source files associated with the media title are selected based on target audiences and compatibility with the video source file. A set of video streams is generated based on the video source file, a different set of audio streams is generated for each selected audio source file, and a different set
20 of timed text streams is generated for each selected timed text source file. The set of video streams, the set(s) of audio streams, and the set(s) of timed text streams are grouped into a video-specific package. The video-specific package is deployed to a content delivery network that provides streams to client devices on-demand.

[0023] To playback the media title, an endpoint application executing on a client
25 device requests a manifest file for the media title from cloud-based media services. The cloud-based media services then select a video-specific package for the media title based on one or more characteristics associated with the user (e.g., a profile language). The cloud-based media services generate one or more recommended presentations based on the selected video-specific package. Each recommended
30 presentation includes the same video track that specifies a subset of the video stream set and a different combination of an audio track that specifies a subset of an audio stream set and an optional timed text track that specifies a subset of a timed text

stream set. The cloud-based media services generate and transmit a manifest file to the endpoint application, where the manifest file enables the endpoint application to stream from the content delivery network and playback the media title in accordance with any of the recommended presentations.

- 5 **[0024]** One drawback of the above approach is that each audio source file and each timed text source file used to generate more than one video-specific package ends up being redundantly processed by the media processing pipeline. As a result, unnecessary amounts of storage resources, processing resources, and time are consumed in generating and deploying the different video-specific packages to the
- 10 CDN. Another drawback of the above approach is that the recommended presentations specified in a given manifest file are limited to combinations of a single video track and other tracks derived from the stream sets that are actually included in the selected single-video package. If the preferences of a user are more aligned with a stream set that is omitted from the selected single-video package, then the quality
- 15 of the viewing experience for the user is likely to be subpar.

- [0025]** For example, suppose that each of twenty video-specific packages was generated based on a different video source file, three audio source files selected from a total of twenty different audio source files, and nine timed text source files selected from a total of sixty different timed text source files. Unnecessary amounts
- 20 of storage resources, processing resources, and time would be consumed in generating and deploying forty redundant audio stream sets and one hundred and twenty redundant timed text stream sets. Furthermore, if the preferences of a user were more aligned with one of the seventeen different audio stream sets and/or one or the fifty-one different timed text stream sets that are omitted from the selected
- 25 single-video package, then the quality of the viewing experience for the user is likely to be degraded.

- [0026]** With the disclosed techniques, however, a media processing pipeline generates a single, inclusive media package for a media title based on all video source files, all audio source files, and all timed text source files associated with the
- 30 media title. The inclusive media package includes all video stream sets, all audio stream sets, and all timed text stream sets derived from all source files associated with the media title. The media processing pipeline also generates metadata that describes relevant characteristics of, distinguishes between, and defines compatibility

between the associated stream sets. The media processing pipeline deploys the inclusive media package to a content delivery network that provides streams to client devices on-demand. The media processing pipeline stores the metadata associated with the inclusive media package in cloud-based memory that is available to a
5 manifest customization application that is included in cloud-based media services.

[0027] Upon receiving a request for a manifest file from an endpoint application, the manifest customization application uses the metadata associated with the inclusive media package to select one of the video stream sets in the inclusive media package based on preferences of an associated user. The manifest customization
10 application generates one or more recommended presentations based on a video track derived from the selected video stream and preferences of the user. Each recommended presentation specifies the same video track and a different combination of an audio track and an optional timed text track. The manifest customization application generates a manifest file that describes each recommended
15 presentation, the video track, each audio track that is included in at least one recommended presentation, and each timed text track that is included in at least one recommended presentation and identifies one or more alternate video tracks that can be made available. Each alternate video track is associated with a different unselected video track. The manifest customization application transmits the
20 manifest file to the endpoint application.

[0028] The manifest file enables the endpoint application to playback any portion of the media title in accordance with any recommended presentation or any other combination of the tracks specified in the manifest file. Furthermore, the manifest file enables the endpoint application to select one of the alternate video tracks specified
25 in the manifest file based on preferences of the user. Prior to or while playing back the media title, the endpoint application can transmit a request for an alternate manifest file based on the selected alternate video track to the manifest customization application. In response to a request for an alternate manifest file based on a specified alternate video track, the manifest customization application generates an
30 alternate manifest file that describes one or more recommended presentations that each include the alternate video track, the alternate video track, each audio track that is included in at least one recommended presentation, and each timed text track that is included in at least one recommended presentation and identifies zero or more

other video tracks that can be made available. The manifest customization application transmits the alternate manifest file to the endpoint application. Upon receiving the alternate manifest file, the endpoint application can switch to the selected alternate video track while continuing to playback the media title.

- 5 **[0029]** At least one technical advantage of the disclosed techniques relative to the prior art is that, with the disclosed techniques, the media processing pipeline does not redundantly process source files when generating and deploying a single inclusive media package to a content delivery network. Therefore, the amounts of storage resources, processing resources, and time consumed in generating and deploying a
- 10 media title can be substantially reduced relative to generating and deploying multiple conventional video-specific packages. Another technical advantage of the disclosed techniques is that, unlike prior art approaches, the manifest customization application selects stream sets on-the-fly from all stream sets derived from all source files associated with a corresponding media title based on the preferences of a given user
- 15 to generate tracks for inclusion in a manifest file. Therefore, the disclosed techniques are able to generate tracks for playback that more closely align with the preferences of users relative to what can be achieved using prior art approaches where manifest files are generated based on a subset of streams included in a single video-specific package. These technical advantages provide one or more technological
- 20 advancements over prior art approaches.

System Overview

- [0030]** Figure 1 is a conceptual illustration of a system 100 configured to implement one or more aspects of the various embodiments. As shown, in some embodiments, the system 100 includes, without limitation, a compute instance 110(1),
- 25 a compute instance 110(2), a display device 192, an audio device 194, an input device 196, an origin server device 140, a content delivery network (CDN) edge server device 150, and cloud-based media services 160. For explanatory purposes, the compute instance 110(1), the compute instance 110(2), and a compute instance 110(3) included in the cloud-based media services 160 are also referred to herein
- 30 individually as “the compute instance 110” and collectively as “the compute instances 110.”

[0031] In some other embodiments, the system 100 can omit one or more of the compute instances 110, the origin server device 140, the CDN edge server device

150, the cloud-based media services 160, the input device 196, or any combination thereof. In the same or other embodiments, the system 100 can further include, without limitation, one or more other compute instances, any number of other display devices, any number of other audio devices, any number and/or types of other cloud-based services, any number of other CDN edge server devices, or any combination thereof.

[0032] The components of the system 100 can be distributed across any number of shared geographic locations and/or any number of different geographic locations and/or implemented in one or more cloud computing environments (*i.e.*, encapsulated shared resources, software, data, etc.) in any combination.

[0033] As shown, the compute instance 110(1) includes, without limitation, a processor 112(1) and a memory 116(1), the compute instance 110(2) includes, without limitation, a processor 112(2) and a memory 116(2), and the compute instance 110(3) includes, without limitation, a processor 112(3) and a memory 116(3). For explanatory purposes, the processor 112(1) - the processor 112(3) are also referred to herein individually as "the processor 112" and collectively as "the processors 112." The memory 116(1) - memory 116(4) are also referred to herein individually as "the memory 116" and collectively as "the memories 116." Each compute instance (including the compute instances 110) can be implemented in a cloud computing environment, implemented as part of any other distributed computing environment, or implemented in a stand-alone fashion.

[0034] The processor 112 can be any instruction execution system, apparatus, or device capable of executing instructions. For example, the processor 112 could comprise a central processing unit, a graphics processing unit, a controller, a micro-controller, a state machine, or any combination thereof. The memory 116 of the compute instance 110 stores content, such as software applications and data, for use by the processor 112 of the compute instance 110. The memory 116 can be one or more of a readily available memory, such as random-access memory, read only memory, floppy disk, hard disk, or any other form of digital storage, local or remote.

[0035] In some other embodiments, any number of compute instances can include any number of processors and any number of memories in any combination. In particular, any number of compute instances (including zero or more of the compute

instances 110) can provide a multiprocessing environment in any technically feasible fashion.

5 **[0036]** In some embodiments, a storage (not shown) may supplement or replace the memory 116 of the compute instance 110. The storage may include any number and type of external memories that are accessible to the processor 112 of the compute instance 110. For example, and without limitation, the storage can include a Secure Digital Card, an external Flash memory, a portable compact disc read-only memory, an optical storage device, a magnetic storage device, or any suitable combination of the foregoing.

10 **[0037]** As depicted in italics, in some embodiments, the compute instance 110(2) is a client device. Some examples of client devices include, without limitation, desktop computers, laptops, smartphones, smart televisions, game consoles, tablets, etc. The compute instance 110(2) can stream media titles via a network connection (not shown) to the CDN edge server device 150. The compute instance 110(0) can
15 playback the media title via the display device 192 and the audio device 194, and receive input from one or more associated user(s) in any technically feasible fashion. The audio device 194 can be any type of device (*e.g.*, a speaker) that can be configured to generate sound based on any amount and/or type of audio data in any technically feasible fashion. The input device 196 can be any type of device that can
20 be configured to receive user input in any technically feasible fashion. Some examples of input devices are a keyboard, a mouse, and a microphone that can be configured to receive user input in any technically feasible fashion.

25 **[0038]** In some other embodiments, the display device 192, the audio device 194, the input device 196, or any combination thereof can be replaced with any number and/or types of output devices, any number and/or types of input devices, any number and/or types of input/output devices, or any combination thereof. In the same or other embodiments, the compute instance 110(2), any number of other compute instances, the display device 192, the audio device 194, any number and/or types of other output devices, the input device 196, any number and/or types of input devices, any number
30 and/or types of input/output devices, or any combination thereof are integrated into a client device.

[0039] In some embodiments, the system 100 includes multiple client devices (including the compute instance 110(2)) that can each access a library of media titles provided by a media streaming service. The origin server device 140, any number of other origin server devices, a CDN (not shown) that includes the CDN edge server device 150, and the cloud-based media services 160 are included in a delivery infrastructure for the media streaming service

[0040] The origin server device 140 and zero or more other origin server devices (not shown) collectively store at least one copy of each of any number of streams for each media title in a library of media titles for streaming to client devices via one or more content delivery networks (not explicitly shown). As referred to herein, a “stream” constitutes an encoded version of a source file. In particular, a video stream constitutes an encoded version of a video source file, an audio stream constitutes an encoded version of an audio file, and a timed text stream constitutes an encoded version of a timed text file. Each video stream, audio stream, and timed text stream associated with a given media title includes, without limitation, a sequence of one or more discrete segments that correspond (in a playback timeline) to a sequence of one or more discrete segments of a baseline video source file for the media title. In some embodiments, the baseline video source file for the media title is an original video source file for the media title.

[0041] As used herein, a “video source file” for a media title includes any amount and/or types of primary video content for a media title. Some examples of types of video source files are original and localized. A localized video source file for a media title is a version of the original source file for the media title that has been modified to conform more to the individual preferences of an associated target audience. For example, text that appears in an original video source file can be replaced with comparable text in another language to generate a localized video source file. In another example, one or more scenes in an original video source file can be replaced with scenes that are more suitable for a different target audience to generate a localized video source file. For example, a hockey scene could be replaced with a soccer scene.

[0042] As used herein an “audio source file” for a media title includes any amount and/or types of primary audio content for the media title. Some examples of types of audio source files are original, dubbed, and description. An original audio source file

for a media title provides an original soundtrack for the media title. A dubbed audio source file for a media title provides a translated soundtrack for the media title. An audio description source file provides either the original soundtrack or a translated soundtrack for the media title and narration that verbally describes what is happening on screen (e.g., physical actions, facial expressions, costumes, settings, scene changes).

[0043] As used herein “timed text” refers to text content and associated playback timing data that enables the text content to be presented synchronously with other media content, such as video content and/or audio content. Some examples of types of timed text are closed captions, subtitles, and forced narratives. A subtitle source file provides a transcription or translation of the spoken dialogue of a media title. A closed caption source file provides a transcription or translation of the spoken dialogue of a media title and describes relevant non-verbal sounds as well as various audio cues (e.g., speaker identification). A forced narrative source file provides text overlays that clarifies communications or alternate languages meant to be understood by a viewer (e.g., a user). Forced narratives are also commonly referred to as “forced narrative subtitles” and forced “subtitles.”

[0044] Upon receiving a request for a stream segment from a client device (e.g., the compute instance 110(2)), the CDN edge server device 150 locates and transmits the nearest copy of the requested stream segment to the client device. More specifically, if the CDN edge server device 150 has a copy of the requested stream segment stored in the associated cache memory, then a “cache hit” occurs and the CDN edge server device 150 transmits the copy of the requested stream segment. If the requested stream segment is not stored in the associated cache memory, then, a “cache miss” occurs. When a cache miss occurs, the CDN edge server device 150 retrieves the requested stream segment from an intermediate cache memory included in the CDN, the origin server device 140, or another origin server device before transmitting The requested stream segment to the client device.

[0045] The cloud-based media services 160 includes, without limitation, any number and/or types of microservices, databases, and storage for activities and content associated with a streaming media service that are allocated to none of the origin server device(s), the CDN(s), and the client devices. Some examples of functionality that the cloud-based media services 160 can provide include, without

limitation, login and billing, logging, personalized media title recommendations, video transcoding, server and network connection health monitoring, and client-specific CDN guidance.

[0046] In general, each compute instance (including each compute instance 110) is configured to implement one or more software applications. For explanatory purposes only, each software application is described as residing in the memory 116 of a single compute instance and executing on the processor 112 of the same compute instance. However, in some embodiments, the functionality of each software application can be distributed across any number of other software applications that reside in the memories of any number of compute instances and execute on the processors of any number of compute instances in any combination. Further, the functionality of any number of software applications can be consolidated into a single software application.

[0047] In particular, the compute instance 110(1) is configured to encode media source files associated with a media title to generate streams and then deploy the streams to a CDN. The media source files include, without limitation, an original video source file, zero or more localized video files, an original audio source file, any number other audio source files, and zero or more timed text source files, and any number and/or types of other media source files (e.g., a trick play source file).

[0048] As described in greater detail previously herein, in a conventional approach to generating and deploying streams to a CDN, a different single-video package is independently generated for each video source file. To generate a video-specific package for a given video source file, one or more audio source file(s), and zero or more of any number of other types of media source files are selected based on compatibility with the video source file and target audience(s) associated with the video-specific package. The video source file and each selected source file are encoded to generate a different set of streams or "stream set" that includes at least one corresponding stream. The media processing pipeline then generates the video-specific package that includes each of the stream sets, stores the video-specific package in an original server device, and then deploys the video-specific package from the origin server device to a CDN.

[0049] To playback the media title, a conventional endpoint application executing on a client device requests a manifest file for the media title from conventional cloud-based media services. In response to the manifest file request, a conventional manifest application selects an edge server in the CDN and a video-specific package.

- 5 The conventional manifest application generates one or more recommended presentations based on the video-specific package. Each recommended presentation specifies the same video track and a different combination of one or more audio tracks and zero or more other tracks that are compatible with the video track, where each track includes a subset of the streams in a different stream set included in the
- 10 video-specific package. The manifest application generates and transmits a manifest file that enables the client device to determine the selected edge server, select a recommended presentation, and request stream segments for the tracks in the selected recommended presentation.

[0050] One drawback of the above approach is that each audio source file and

15 each timed text source file used to generate more than one video-specific package ends up being redundantly processed by the media processing pipeline. As a result, unnecessary amounts of storage resources, processing resources, and time are consumed in generating and deploying the different video-specific packages to the CDN. Another drawback of the above approach is that the recommended

20 presentations specified in a given manifest file are limited to combinations of a single video track and other tracks derived from the stream sets that are actually included in the selected single-video package. If the preferences of a user are more aligned with a stream set that is omitted from the selected single-video package, then the quality of the viewing experience for the user is likely to be subpar.

25 **Generating an Inclusive Media Package for a Media Title**

[0051] To address the above problems, the system 100 includes, without limitation, a media processing pipeline 120 that generates an inclusive media package 130 and a package metadata set 132 based on media source files 102 for a media title. The media source files 102 include, without limitation, any number and/or

30 types of media source files for the media title. For explanatory purposes, the media source files 102 as described herein include an original video source file, zero or more localized video source files, zero or more other video source files, an original audio source file, zero or more other audio source files, and zero or more timed text

source files, but do not include any other types of media source files (e.g., trick play source files). The inclusive media package 130 is also referred to herein as a “media package.”

[0052] Although not shown in Figure 1, the inclusive media package 130 includes, without limitation, a title metadata set that describes the media title and both a media metadata set and a stream set for each media source file associated with the media title. The media metadata sets describe relevant characteristics of, distinguish between, and define compatibility between the associated stream sets. Some examples of media metadata sets are video metadata sets, audio metadata sets, and timed text metadata sets. Each stream set includes one or more streams and a different stream metadata set for each of the streams. Some examples of stream sets are video stream sets, audio stream sets, and timed text stream sets. The stream metadata sets included in a given stream set describe relevant characteristics of and distinguish between the associated streams. An example of the inclusive media package 130 is described in greater detail below in conjunction with Figure 2. The package metadata set 132 includes the title metadata set, the media metadata sets, and the stream metadata sets.

[0053] As shown, in some embodiments, the media processing pipeline 120 resides in the memory 116(1) of the compute instance 110(1) and executes on the processor 112 of the compute instance 110. In some other embodiments, any number of portions (including all) of the functionality described herein with respect to the media processing pipeline 120 can be distributed across any number of compute instances in any technically feasible fashion.

[0054] As shown, the media processing pipeline 120 includes, without limitation, the inclusive media package 130, an ingest application 122, an encoding application 124, and a deployment application 126. The ingest application 122 initializes the inclusive media package 130 to an empty package. The ingest application 122 then generates a title metadata set based on a baseline video source file (not shown) and adds the title metadata set to the inclusive media package 130.

[0055] The baseline video source file defines the runtime of the media title and is used to establish the segments of the media title. In some embodiments, the ingest application 122 subsequently verifies that any other video source files included in the

media source files 102 conform to (e.g., have the same runtime as) the baseline video source file. In some embodiments, the ingest application 122 sets the baseline video source file to the original video source file by default.

- [0056]** The title metadata set can specify any amount and/or types of information that is relevant to playing back the media title. In some embodiments, the title metadata set specifies, without limitation, a title identifier (ID), a baseline video ID, a fallback video ID, a runtime, and segment definitions. The title ID, the baseline video ID, and the fallback video ID identify the media title, the baseline video source file, and a fallback video source file, respectively, in any technically feasible fashion.
- [0057]** The fallback video source file defines the source of a video stream set that is used to generate a manifest file when none of the video stream sets in the final version of the inclusive media package 130 is associated with one or more preferred languages for a user. As described in greater detail below, preferred language(s) for a user can be explicitly specified (e.g., via metadata) or inferred (e.g., determined based on a location). The fallback video source file and the fallback video ID are also referred to herein as a “default source file” and a “default video identifier,” respectively.

- [0058]** The segments individually correspond to different non-overlapping periods of time and collectively span the runtime in a contiguous fashion. The ingest application 122 can generate the segment definitions in any technically feasible fashion. In some embodiments the ingest application 122 partitions the baseline source video file into shots, where each shot is captured continuously from a single camera or virtual representation of a camera (e.g., in the case of computer animated videos). The ingest application 122 then selects sequences of one or more shots based on a target segment length and defines a segment for each selected sequence.

- [0059]** For each video source file included in the media source files 102, the ingest application 122 generates an associated video metadata set and adds the video metadata set to the inclusive media package 130. Each video metadata set specifies a different set of values for a set of video characteristics. In some embodiments, the set of video characteristics includes a language, a video type (e.g., original, localized),

and optionally any other characteristics of video content that are relevant to streaming the media title, such as an aspect ratio.

[0060] For each audio source file included in the media source files 102, the ingest application 122 generates an associated audio metadata set and adds the audio metadata set to the inclusive media package 130. Each audio metadata set specifies a different set of values for a set of audio characteristics. In some embodiments, the set of audio characteristics includes a language, an audio type, and zero or more other characteristics of audio content that are relevant to streaming the media title. Some examples of audio types are original, dubbed, and audio description.

[0061] For each timed text source file included in the media source files 102, the ingest application 122 generates an associated timed text metadata set and adds the timed text metadata set to the inclusive media package 130. Each timed text metadata set specifies a different set of values for a set of timed text characteristics. In some embodiments, the set of timed text characteristics includes a language, a timed text type, an optional timed text subtype, and zero or more other characteristics of audio content that are relevant to streaming the media title. Some examples of timed text types are subtitle, closed caption, and forced narrative. The optional timed text subtype is either full or partial. As described in greater detail below, the optional timed text subtype is used to avoid displaying duplicate text during playback of the media title.

[0062] A timed text subtype of full indicates that the corresponding timed text source file includes translations of text that is included in the video content of the media title. A timed text subtype of full therefore indicates that the associated timed text source file and the corresponding timed text streams are compatible with each video source file that is associated with a different language and the corresponding video streams. A timed text subtype of partial indicates that the corresponding timed text source file does not include translations of text that is included in the video content of the media title. A timed text subtype of partial therefore indicates that the associated timed text source file and the corresponding timed text streams are compatible with each video source file that is associated with the same language and the corresponding video streams.

[0063] The encoding application 124 generates a stream set for each of the media source files 102 and adds each of the stream sets to the inclusive media package 130. Within the inclusive media package 130, the stream set for a given media source file 102 is associated with the media metadata set for the media source file 5 102 in any technically feasible fashion (e.g., indexes, IDs, etc.).

[0064] Each stream set includes, without limitation, at least one stream and a different stream metadata set for each stream. As noted previously herein, a stream is an encoded version of a source file. As persons skilled in the art will recognize, the number of video streams included in a video stream set is typically substantially 10 larger than the number of streams included in any other type of stream set. (e.g., an audio stream set). The encoding application 124 can generate a stream in any technically feasible fashion that enables each segment of the stream to be decoded independently of the other segments of the stream. Ensuring that each stream segment can be independently decoded enables endpoint applications to switch 15 between streams at segment boundaries during playback of the media title.

[0065] In some embodiments, the encoding application 124 downscales a video source file to multiple different resolutions to generate lower-resolution video source files. The encoding application 124 then executes a shot-based encoder on the video source file and each of the lower-resolution video source files across different sets of 20 one or more encoding profiles to generate video streams having different combinations of resolutions and bitrates. The encoding application 124 propagates any amount of the metadata included in each media metadata set to each stream in the associated stream set.

[0066] The stream metadata sets included in a given stream set describe relevant 25 characteristics of, and distinguish between, the associated streams. For instance, in some embodiments, a video stream metadata set specifies a size in bytes, a bitrate, a resolution, an encoding profile, a frame rate, and a quality score for the associated stream.

[0067] The deployment application 126 adds any amount (including none) and/or 30 types of deployment-related metadata (not shown) to the inclusive media package 130. One example of deployment-related metadata is digital rights management (DRM) metadata. The deployment application 126 extracts the package metadata set

132 from the inclusive media package 130. As described in greater detail below, the package metadata set 132 is associated with the inclusive media package 130 upon which both a manifest file 174 and an alternate manifest file 178 are based. Notably, the manifest file 174 and the alternate manifest file 178 itemize different video tracks
5 derived from different video stream sets included in the inclusive media package 130.

[0068] In some embodiments, the package metadata set 132 includes, without limitation, the title metadata set, the video metadata sets, the stream metadata sets (included in the stream sets), and any deployment-related metadata. In some other
10 embodiments, the package metadata set 132 includes a subset of the title metadata set, subsets of the video metadata sets, subsets of the stream metadata sets, any amount of deployment-related metadata, or any combination thereof, that is relevant to generating manifest files as described in greater detail below.

[0069] As shown, the deployment application 126 stores the package metadata set 132 in a media services database 162 that is included in the cloud-based media
15 services 160. As also shown, the deployment application 126 transmits the inclusive media package 130 to the origin server device 140. The origin server device 140 stores the inclusive media package 130 in a memory associated with the origin server device 140. Any number and/or types of software applications executing on any number and/or types of server devices included in a CDN can transmit a request for
20 the inclusive media package 130 to the origin server device 140 at any point-in-time. In response to each request, a software application executing on the origin server device 140 transmits the inclusive media package 130 to the requesting software application. In this fashion, the deployment application 126 distributes streams for playback of the media title via the inclusive media package 130.

[0070] In particular, a software application executing on the CDN edge server device 150 transmits a request for the inclusive media package 130 to the origin
25 server device 140 either directly or indirectly via any number of intermediate server devices in the CDN. After receiving the inclusive media package 130, the software application executing on the CDN edge server device 150 stores the inclusive media package 130 in a cache memory associated with the CDN edge server device 150.
30 An example of the inclusive media package 130 is described in greater detail below in conjunction with Figure 2.

[0071] As shown, the cloud-based media services 160 includes, without limitation, the media services database 162 and the compute instance 110(3). Although not shown, the cloud-based media services 160 generates, maintains, and stores in the media services database 162 a CDN metadata set 134, a different client device metadata set for each client device, and a different user metadata set for each user associated with the streaming media service,

[0072] To generate and maintain the CDN metadata set 134, the cloud-based media services 160 monitors the health of the CDN edge server device 150 and any number of other CDN edge server devices in any number of CDNs, the compute instance 110(2), any number of other client devices, and associated network connections. The CDN metadata set 134 specifies any amount and/or types of characteristics of the CDN edge server devices and network connections. In some embodiments, the CDN metadata set 134 specifies, without limitation, a routing distance, throughout, and latency for any number of connections among CDN edge server devices and client devices.

[0073] Each client device metadata set specifies any number and/of types of characteristics of an associated client device. For instance, in some embodiments, each client device metadata set specifies, without limitation, a location and a display resolution of an associate client device. Each user metadata set specifies any number and/or types of characteristics and preferences of an associated user. In some embodiments, each user metadata set specifies, without limitation, a profile language, an audio preference, and a timed text preference. A profile language for a user specifies the language of user interfaces that the streaming media provider uses to interact with the user. An audio preference specifies a language and optionally an audio type. A timed text preference specifies a language and a timed text type.

[0074] Advantageously, the media processing pipeline 120 does not redundantly process any of the media source files 102 when generating and deploying the inclusive media package 130. Therefore, the amounts of storage resources, processing resources, and time consumed in generating and deploying the media title can be substantially reduced relative to prior art techniques that represent the media title using multiple video-specific packages. And, as described in greater detail below, the inclusive media package 130 and the package metadata set 132 allow a manifest customization application 170 to generate recommended presentations of the media

title and associated tracks for playback that more closely align with the preferences of users relative to what can be achieved using multiple video-specific packages. In that regard, the inclusive media package 130 includes at least one encoded version (*i.e.*, stream) of each of the media source files 102. Consequently, the manifest customization application 170 can generate the tracks for inclusion in a manifest file on-the-fly from all video stream sets, audio stream sets, and optional timed text stream sets derived from all of the media source files 102 based on the package metadata set 132 and any amount and/or type of other relevant metadata, such as the preferences of a given user.

10 **Generating Manifest Files Based on Inclusive Media Packages**

[0075] As shown, the manifest customization application 170 resides in the memory 116(23) of the compute instance 110(3) and executes on the processor 112(23) of the compute instance 110(3). The manifest customization application 170 generates manifest files that are individually tailored to different users in response to manifest requests received from client devices. Each manifest request identifies a user and a media title.

[0076] For explanatory purposes, the functionality of the manifest customization application 170 is described herein in the context of a sequence of events that are denoted by circles numbers 1 - 11. Also for explanatory purposes, the inclusive media package 130 as described herein includes video, audio, and timed text stream sets but does not include any other types of stream sets (*e.g.*, trick play stream sets).

[0077] As depicted with a circle numbered 1, the manifest customization application 170 receives a manifest request 172 from an endpoint application 180. The endpoint application 180 resides in the memory 116(3) of the compute instance 110(2) (a client device) and executes on the processor 112(3) of the compute instance 110(2). The manifest request 172 identifies the compute instance 110(2), a user, and the media title associated with the media source files 102.

[0078] In response to the manifest request 172, the manifest customization application 170 retrieves the package metadata set 132, the CDN metadata set 134, a client device metadata set 136, and a user metadata set 138 from the media services database 162 (depicted with the circle numbered 2). The client device

metadata set 136 and the user metadata set 138 are associated with the compute instance 110(2) and the user of the compute instance 110(2), respectively.

[0079] The manifest customization application 170 designates each of the CDN edge server device 150 and optionally one or more other CDN edge server devices as a “recommended” CDN edge server device based on the CDN metadata set 134 and the client device metadata set 136. Typically, the manifest customization application 170 designates at least one CDN edge server device that is proximate to the client device as a recommended server device.

[0080] The manifest customization application 170 selects one of the video stream sets that are included in the inclusive media package 130 based on the package metadata set 132, the user metadata set 138, and optionally the client device metadata set 136. In some embodiments, the manifest customization application 170 determines whether the language specified in any of the video metadata sets in the package metadata set 132 matches a preferred language and/or a preferred aspect ratio specified in the user metadata set 138 and/or the client device metadata set 136. If the manifest customization application 170 finds a matching video stream set and determines that rights to access the matching video stream set are not restricted for the location associated with the client device, then the manifest customization application 170 selects the matching video stream set. Otherwise, the manifest customization application 170 selects the video stream set associated with the default video identifier specified in the title metadata set included in the package metadata set 132.

[0081] If an audio preference is specified in the user metadata set 138, then the manifest customization application 170 determines whether one of the audio stream sets that are included in the inclusive media package 130 matches the audio preference based on the audio metadata sets in the package metadata set 132. If the manifest customization application 170 finds a matching audio stream set, then the manifest customization application 170 selects the matching audio stream set. Otherwise, the manifest customization application 170 executes any number and/or types of heuristics to select an audio stream set based on the user metadata set 138, the package metadata set 132, optionally the client device metadata set 136, and optionally the selected video stream set.

[0082] If a timed text preference is specified in the user metadata set 138, then the manifest customization application 170 determines whether any of the timed text stream sets included in the inclusive media package 130 matches the timed text preference based on the timed text metadata sets in the package metadata set 132.

- 5 If the manifest customization application 170 finds a single matching timed text stream set, then the manifest customization application 170 selects the matching timed text stream set.

- [0083]** If, however, the manifest customization application 170 finds two different matching timed text stream sets, then the manifest customization application 170
10 selects one of the matching timed text streams based on compatibility with the selected video stream set. In some embodiments, if the language specified in the timed text preference matches the language associated with the selected video stream set, then the manifest customization application 170 selects the matching timed text stream that is associated with the timed text subtype of partial. Otherwise,
15 the manifest customization application 170 selects the matching timed text stream that is associated with the timed text subtype of full.

- [0084]** If the manifest customization application 170 does not find any matching timed text stream sets, then the manifest customization application 170 executes any number and/or types of heuristics to select a timed text stream set based on the user
20 metadata set 138, the package metadata set 132, optionally the client device metadata set 136, and optionally the selected video stream set.

- [0085]** The manifest customization application 170 selects at least one video stream from the video streams included in the selected video stream set for inclusion in a video track. Accordingly, the video track is a subset of the video streams
25 included in the selected video stream set. As used herein a “subset” can be a proper subset or an improper subset. The manifest customization application 170 selects at least one audio stream from the audio streams included in the selected audio stream set for inclusion in an audio track. Accordingly, the audio track is a subset of the audio streams included in the selected audio stream set. If a timed text stream set is
30 selected, then the manifest customization application 170 selects at least one timed text stream from the timed text streams included in the selected timed text stream set for inclusion in a timed text track. The manifest customization application 170 can

select the one or more streams that are included in a track in any technically feasible fashion.

[0086] In some embodiments the manifest customization application 170 selects at least one stream from the streams included in the selected stream set for inclusion in an track based the corresponding stream metadata sets, the associated media metadata sets, the client device metadata set 136, the CDN metadata set 134, the user metadata set 138, any amount and/or types of other data, any number of selection criteria, or any combination thereof. Some examples of selection criteria that the manifest customization application 170 can use to select the streams that are included in a track are network throughput, bitrates of streams, user preferences, compatibility with the client device, rights restrictions associated with the location of the client device, and rights restrictions associated with the user.

[0087] If the manifest customization application 170 selects at least one timed text stream for inclusion in a timed text track, then the manifest customization application 170 also groups the video track, the audio track, and the timed text track to generate a recommended presentation. Otherwise, the manifest customization application 170 groups the video track and the audio track to generate a recommended presentation.

[0088] The manifest customization application 170 defines zero or more other recommended presentations that each specify the video track and a different combination of an audio track and an optional timed text track. The manifest customization application 170 can define recommended presentations and any number of other audio tracks and/or other timed text tracks specified in the recommended presentations based on any number and/or types of heuristics and any number and/or types of criteria in any technically feasible fashion.

[0089] The manifest customization application 170 generates a video track description for the video track, a different audio track description for each audio track, and a different timed text track description for each timed text track. Video tracks, audio tracks, and timed text tracks are also referred to herein as “tracks.” Each track description specifies a track identifier, a subset of the associated media metadata set, the stream(s) included in the corresponding track, a subset of the stream metadata set for each stream, and one or more locator(s) for each stream.

[0090] The locator(s) for a stream are references that can be used in conjunction with any amount (including none) of other data in a manifest file to request individual segments of the stream from the recommended server device. In some embodiments, a track description could specify a different universal record locator (URL) for each segment of each stream in a corresponding track. In some other embodiments, a track description could specify a base universal record locator (URL) associated with a first segment of a first stream in a corresponding track and segment offsets for each of the other stream segments in a corresponding track.

[0091] Notably, the manifest customization application 170 can generate, for inclusion in a manifest file, a different alternate video track summary for each of zero or more “alternate” video tracks. Each alternate video track summary indicates that the associated alternate video track can be made available. More precisely, each alternate video track summary identifies a different alternate video track that can be derived from a different unselected video stream set included in the inclusive media package 130, specifies at least a subset of the associated video metadata set, and optionally specifies any amount and/or types of other relevant metadata. For instance, in some embodiments, each alternate video track summary specifies metadata that includes an alternate video track identifier, a language, a video type, and optionally an aspect ratio. In the same or other embodiments, the alternate video track identifier embeds an identifier for the corresponding video stream set.

[0092] The manifest customization application 170 generates a manifest file 174 that specifies, without limitation, each recommended server device, a title description set, each recommended presentation, each track description, and each alternate video track summary. The title description set identifies the media title and specifies any amount and/or types of metadata that is relevant to the playback of the media title. In some embodiments, the title description set is at least a subset of the title metadata set that is included in the inclusive media package 130. An example of the manifest file 174 is described in greater detail below in conjunction with Figure 3.

[0093] As depicted with the circle numbered 3, the manifest customization application 170 transmits the manifest file 174 to the endpoint application 180, thereby fulfilling the manifest request 172. As described in greater detail below, the manifest file enables the endpoint application 180 to playback the media title via any of the recommended presentations.

- [0094]** To playback the media title, the endpoint application 180 selects the tracks specified in one of the recommended presentations. For each segment of the media title, the endpoint application 180 selects a corresponding segment from one stream in each selected track. As depicted with the circle numbered 4, the endpoint
- 5 application 180 transmits segment requests 182 for the selected stream segments to the CDN edge server device 150. In response, the CDN edge server device 150 transmits stream segments 184 to the endpoint application 180 (depicted with the circle numbered 5). The endpoint application 180 decodes each stream segment to generate decoded media content and plays back the media content.
- 10 **[0095]** Advantageously, playing back a media title via a recommended presentation automatically tailors (e.g., personalizes) the media viewing experience for the user based, at least in part, on user preferences specified in the user metadata set 138. And, as described below, each alternate video track description in the manifest file 174 can provide the endpoint application 180 with an opportunity to
- 15 perform video track switching to further align the media viewing experience with user preferences during playback of the media title.
- [0096]** For instance, in some embodiments, the endpoint application 180 enables a user, via a graphical user interface, to select any of the alternate video track(s) based on the language(s) and/or aspect ratio(s) specified in the corresponding alternate
- 20 video track summary. In some other embodiments, including the embodiment depicted in Figure 1, the endpoint application 180 automatically selects an alternate video track based on any type of user information.
- [0097]** As depicted with the circle numbered 6, during the playback of the media title, the endpoint application 180 receives an audio language change request 198
- 25 from the user via the input device 196. In response, the endpoint application 180 determines that the language associated with one of the alternate video tracks matches the newly requested audio language. As depicted with the circle numbered 7, and while continuing to playback the media title, the endpoint application 180 transmits an alternate manifest request 176 to the manifest customization application
- 30 170. The alternate manifest request 176 identifies the client device, the user, the media title, and the alternate video track that matches the requested audio language.

Generating Alternate Manifest Files for Video Track Switching

[0098] The manifest customization application 170 generates alternate manifest files that are individually tailored to different users in response to alternate manifest requests received from client devices. Each alternate manifest request identifies a user, a media title, and a requested alternate video track. As used herein, an “alternate manifest file” refers to a manifest file for a media title that is generated based on a request for a manifest file that itemizes an alternate video track. In some embodiments, the alternate manifest request 176 is a request for a manifest file that identifies an alternate video track and/or a video stream set from which an alternate video track is to be derived.

[0099] In response to the alternate manifest request 176, the manifest customization application 170 re-retrieves the package metadata set 132, the CDN metadata set 134, the client device metadata set 136, and the user metadata set 138 from the media services database 162 (depicted with the circle numbered 8). The manifest customization application 170 selects a video stream set from the video stream sets included in the inclusive media package 130 based on the package metadata set 132 and metadata that is associated with the requested alternate video track and specified in the alternate manifest request 176. The manifest customization application 170 selects one or more video streams from the video streams included in the selected video stream set for inclusion in the alternate video track.

[0100] In some embodiments, the manifest customization application 170 preferentially selects an audio stream set from the audio stream sets included in the inclusive media package 130 based on associated metadata specified in the package metadata set 132 and a language that is associated with the alternate video track. The manifest customization application 170 selects one or more audio streams from the audio streams included in the selected audio stream set for inclusion in an audio track.

[0101] In some embodiments, the manifest customization application 170 optionally selects one or more timed text streams from inclusive media package 130 for inclusion in an optional timed text track using techniques similar to the techniques described previously herein in conjunction with generating the manifest file 174. If the manifest customization application 170 selects at least one timed text stream for inclusion in a timed text track, then the manifest customization application 170 also

groups the alternate video track, the audio track, and the timed text track to generate a recommended presentation. Otherwise, the manifest customization application 170 groups the alternate video track and the audio track to generate a recommended presentation.

- 5 **[0102]** The manifest customization application 170 defines zero or more other recommended presentations that each specify the alternate video track and a different combination of an audio track and an optional timed text track. The manifest customization application 170 generates a video track description for the alternate video track, a different audio track description for each audio track, and a different
10 timed text track description for each timed text track. Optionally, the manifest customization application 170 can generate any number of different alternate video track summaries for inclusion in the alternate manifest file 174.

- [0103]** The manifest customization application 170 generates an alternate manifest file 178 that specifies, without limitation, one or more recommended server devices,
15 the title description set, the one or more recommended presentations, the video track description for the requested alternate video track, an audio track description for each audio track included in at least one of the recommended presentations, a timed text track description for each timed text track included in at least one of the recommended presentations, and zero or more alternate video track summaries. An
20 example of the alternate manifest file 178 is described in greater detail below in conjunction with Figure 4.

- [0104]** As depicted with the circle numbered 9, the manifest customization application 170 transmits the alternate manifest file 178 to the endpoint application 180, thereby fulfilling the alternate manifest request 176. In response, the endpoint
25 application 180 deselects the currently selected tracks and selects the tracks specified in one of the recommended presentations specified in the alternate manifest request 176.

- [0105]** For each segment of the media title, the endpoint application 180 selects a corresponding stream segment from one stream in each newly selected track. As
30 depicted with the circle numbered 10, the endpoint application 180 transmits alternate segment requests 186 for the selected stream segments to the CDN edge server device 150. In response, the CDN edge server device 150 transmits alternate stream

segments 188 to the endpoint application 180 (depicted with the circle numbered 11). The endpoint application 180 decodes each alternate stream segment to generate decoded media content and plays back the media content. In this fashion, the manifest customization application 170 enables video switching during the playback
5 of the media title.

[0106] Please note that the techniques described herein are illustrative rather than restrictive and can be altered without departing from the broader spirit and scope of the embodiments. Many modifications and variations on the functionality of the media processing pipeline 120, the ingest application 122, the encoding application 124, the
10 deployment application 126, the cloud-based media services 160, the manifest customization application 170, the endpoint application 180, the origin server device 140, the CDN, and the CDN edge server device 150 as described herein will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments.

[0107] Similarly, the storage, organization, amount, and/or types of data described herein are illustrative rather than restrictive and can be altered without departing from the broader spirit and scope of the embodiments. In that regard, many modifications and variations on the media source files 102, the inclusive media package 130, the package metadata set 132, the CDN metadata set 134, the media services database
20 162, the client device metadata set 136, the user metadata set 138, the manifest request 172, the manifest file 174, the alternate manifest request 176, the alternate manifest file 178, the segment requests 182, the stream segments 184, the alternate segment requests 186, the alternate stream segments 188, and the audio language change request 198 as described herein will be apparent to those of ordinary skill in
25 the art without departing from the scope and spirit of the described embodiments.

[0108] For instance, in some embodiments, the ingest application 122 can generate video metadata sets that specify values for any number and/or types of characteristics that collectively differentiate between corresponding video stream sets and therefore different video tracks. The manifest customization application 170 can
30 use the differentiating characteristics to select a video stream set and then generate a manifest file based on the video stream. The endpoint application 180 can use the differentiating characteristics (specified in the manifest file) to request an alternate manifest file based on one of the alternate video tracks specified in the manifest file.

[0109] In some alternate embodiments, the media source files 102 include one or more trick play source files. Each trick play source file specifies a subset of the frames of an associated video source file. For each trick play source file included in the media source files 102, the ingest application 122 generates an associated trick
5 play metadata set and adds the trick play metadata set to the inclusive media package 130. Each trick play metadata set specifies a matching video ID, a trick play type, and zero or more other characteristics of trick plays that are relevant to streaming the media title. The matching video ID identifies the video source file and/or the associated video metadata set that the trick play source file is intended to
10 be used with in any technically feasible fashion. Some examples of trick play types are fast forward, skip mode, rewind, and search. For each trick play metadata set, the encoding application 124 generates a trick play stream set based on the associated trick play source file and/or the associated video source file and adds the trick play stream set to the inclusive media package 130. Each trick play stream
15 specifies a subset of the encoded frames in a corresponding video stream. The deployment application 126 adds the trick play metadata set(s) to the package metadata set 132. The manifest customization application 170 adds zero or more trick play track descriptions to the manifest file 174 and the alternate manifest file 178. The endpoint application 180 can use one or more trick play tracks specified in
20 a manifest file (e.g., the manifest file 174, the alternate manifest file 178) to provide visual feedback during non-normal playback modes, such as a fast forward mode, a skip mode, a rewind mode, a search mode, etc.

[0110] It will be appreciated that the system 100 shown herein is illustrative and that variations and modifications are possible. For example, the functionality
25 provided by the ingest application 122, the encoding application 124, and the deployment application 126 as described herein can be integrated into or distributed across any number of software applications (including one), and any number of components of the system 100. Further, the connection topology between the various units in Figure 1 can be modified as desired.

30 Exemplar Inclusive Media Package

[0111] Figure 2 is a more detailed illustration of the inclusive media package 130 of Figure 1, according to various embodiments. More specifically, the inclusive media package 130 depicted in Figure 2 is an exemplar inclusive media package. As

shown, in some embodiments, the inclusive media package 130 includes, without limitation, a title metadata set 210, a video stream set 220(1) - a video stream set 220(3), a video metadata set 230(1) - a video metadata set 230(3), an audio stream set 240(1) - an audio stream set 240(6), an audio metadata set 250(1) - an audio
5 metadata set 250(6), a timed text stream set 260(1) - a timed text stream set 260(6), and a timed text metadata set 270(1) - a timed text metadata set 270(6).

[0112] As described previously herein in conjunction with Figure 1, the title metadata set 210 specifies a title ID (titleID) associated with a media title, a runtime of the media title, segment definitions (segmentDefinitions) for the media title, a baseline
10 video ID (baselineVideoID), and a fallback video ID (fallbackVideoID).

[0113] As shown, the video metadata set 230(1) specifies a language of English (en) and a video type (videoType) of original. The video metadata set 230(1) is associated with the video stream set 220(1). The video stream set 220(1) therefore specifies one or more video streams and a different stream metadata set for each of
15 the video streams, where each of the video streams is a different encoded version of an English original video source file.

[0114] As shown, the video metadata set 230(2) specifies a language of French (fr) and a video type of localized. The video metadata set 230(2) is associated with the video stream set 220(2). The video stream set 220(2) therefore specifies one or
20 more video streams and a different stream metadata set for each of the video streams, where each of the video streams is a different encoded version of a French localized video source file.

[0115] As shown, the video metadata set 230(3) specifies a language of Spanish (es) and a video type of localized. The video metadata set 230(3) is associated with
25 the video stream set 220(3). The video stream set 220(3) therefore specifies one or more video streams and a different stream metadata set for each of the video streams, where each of the video streams is a different encoded version of a Spanish localized video source file.

[0116] As shown, the audio metadata set 250(1) specifies a language of English
30 and an audio type (audioType) of original. The audio metadata set 250(1) is associated with the audio stream set 240(1). The audio stream set 240(1) therefore

specifies one or more audio streams and a different stream metadata set for each of the audio streams, where each of the audio streams is a different encoded version of an English original audio source file.

5 **[0117]** As shown, the audio metadata set 250(2) specifies a language of French and an audio type of dubbed. The audio metadata set 250(2) is associated with the audio stream set 240(2). The audio stream set 240(2) therefore specifies one or more audio streams and a different stream metadata set for each of the audio streams, where each of the audio streams is a different encoded version of a French dubbed audio source file.

10 **[0118]** As shown, the audio metadata set 250(3) specifies a language of Spanish, a timed text type of subtitle, and a timed text subtype of full. The audio metadata set 250(3) is associated with the audio stream set 240(3). The audio stream set 240(3) therefore specifies one or more audio streams and a different stream metadata set for each of the audio streams, where each of the audio streams is a different encoded
15 version of a Spanish dubbed audio source file.

[0119] As shown, the audio metadata set 250(4) specifies a language of English and an audio type of description. The audio metadata set 250(4) is associated with the audio stream set 240(4). The audio stream set 240(4) therefore specifies one or more audio streams and a different stream metadata set for each of the audio
20 streams, where each of the audio streams is a different encoded version of an English audio description source file.

[0120] As shown, the audio metadata set 250(5) specifies a language of French and an audio type of description. The audio metadata set 250(5) is associated with the audio stream set 240(5). The audio stream set 240(5) therefore specifies one or
25 more audio streams and a different stream metadata set for each of the audio streams, where each of the audio streams is a different encoded version of a French audio description source file.

[0121] As shown, the audio metadata set 250(6) specifies a language of Spanish and an audio type of description. The audio metadata set 250(6) is associated with
30 the audio stream set 240(6). The audio stream set 240(6) therefore specifies one or more audio streams and a different stream metadata set for each of the audio

streams, where each of the audio streams is a different encoded version of a Spanish audio description source file.

5 **[0122]** As shown, the timed text metadata set 270(1) specifies a language of English, a timed text type (timedTextType) of subtitle, and a timed text subtype (timedTextSubtype) of full. The timed text metadata set 270(1) is associated with the timed text stream set 260(1). The timed text stream set 260(1) therefore specifies one or more timed text streams and a different stream metadata set for each of the timed text streams, where each of the timed text streams is a different encoded version of an English full subtitle source file.

10 **[0123]** As shown, the timed text metadata set 270(2) specifies a language of French, a timed text type of subtitle, and a timed text subtype of full. The timed text metadata set 270(2) is associated with the timed text stream set 260(2). The timed text stream set 260(2) therefore specifies one or more timed text streams and a different stream metadata set for each of the timed text streams, where each of the
15 timed text streams is a different encoded version of a French full subtitle source file.

[0124] As shown, the timed text metadata set 270(3) specifies a language of Spanish, a timed text type of subtitle, and a timed text subtype of full. The timed text metadata set 270(3) is associated with the timed text stream set 260(3). The timed text stream set 260(3) therefore specifies one or more timed text streams and a
20 different stream metadata set for each of the timed text streams, where each of the timed text streams is a different encoded version of a Spanish full subtitle source file.

[0125] As shown, the timed text metadata set 270(4) specifies a language of English, a timed text type of subtitle, and a timed text subtype of partial. The timed text metadata set 270(4) is associated with the timed text stream set 260(4). The
25 timed text stream set 260(4) therefore specifies one or more timed text streams and a different stream metadata set for each of the timed text streams, where each of the timed text streams is a different encoded version of an English partial subtitle source file.

[0126] As shown, the timed text metadata set 270(5) specifies a language of
30 French, a timed text type of subtitle, and a timed text subtype of partial. The timed text metadata set 270(5) is associated with the timed text stream set 260(5). The

timed text stream set 260(5) therefore specifies one or more timed text streams and a different stream metadata set for each of the timed text streams, where each of the timed text streams is a different encoded version of a French partial subtitle source file.

- 5 **[0127]** As shown, the timed text metadata set 270(6) specifies a language of Spanish, a timed text type of subtitle, and a timed text subtype of partial. The timed text metadata set 270(6) is associated with the timed text stream set 260(6). The timed text stream set 260(6) therefore specifies one or more timed text streams and a different stream metadata set for each of the timed text streams, where each of the
- 10 timed text streams is a different encoded version of a Spanish partial subtitle source file.

Exemplar Manifest File

- [0128]** Figure 3 is a more detailed illustration of the manifest file 174 of Figure 1, according to various embodiments. More specifically, the manifest file 174 depicted
- 15 in Figure 3 is an exemplar manifest file. For explanatory purposes, the manifest customization application 170 generates the manifest file 174 based on the example of the inclusive media package 130 that is depicted in Figure 2, a location of Spain, a preferred language (e.g., a profile language) of English, a preferred audio language of English, a preferred timed text language of English, and a preferred timed text type of
- 20 subtitle.

- [0129]** As shown, the manifest file 174 specifies, without limitation, a recommended server device 310, a media title description 320, a recommended presentation 330(1) - a recommended presentation 330(6), a video track description 340, an audio track description 350(1), an audio track description 350(2), a timed text
- 25 track description 360(1), a timed text track description 360(2), an alternate video track summary 370(1), and an alternate video track summary 370(2). As shown, the recommended server device 310 is the CDN edge server device 150 of Figure 1. The media title description 320 is equal to the title metadata set 210 specified in the inclusive media package 130.

- 30 **[0130]** The recommended presentation 330(1) specifies an English original video track that is associated with the video track description 340, an English original audio track that is associated with the audio track description 350(1), and an English partial

subtitle track that is associated with the timed text track description 360(1). The recommended presentation 330(2) specifies the English original video track, the English original audio track, and a Spanish full subtitle track that is associated with the timed text track description 360(2). The recommended presentation 330(3)

5 specifies the English original video track and the English original audio track.

[0131] The recommended presentation 330(4) specifies the English original video track, an English audio description track that is associated with the audio track description 350(2), and the English partial subtitle track. The recommended presentation 330(5) specifies the English original video track, the English audio
10 description track, and the Spanish full subtitle track. The recommended presentation 330(6) specifies the English original video track and the English audio description track.

[0132] The video track description 340 describes a video track that includes a subset of the video streams in the video stream set 220(1). Referring back to Figure
15 2, the video stream set 220(1) is an English original video stream set. In some embodiments, the video track description 340 specifies a video track identifier, a language of English, a video type of original, the constituent video streams, and stream-specific metadata for each of the constituent video streams. The stream-specific metadata for a given constituent video stream specifies a size in bytes, a
20 bitrate, a resolution, an encoding profile, a frame rate, a quality score, and one or more URLs that the CDN edge server device 150 can use to retrieve each segment of the video stream.

[0133] The audio track description 350(1) describes an audio track that includes a subset of the audio streams in the audio stream set 240(1). Referring back to Figure
25 2, the audio stream set 240(1) is an English original audio stream set. In some embodiments, the audio track description 350(1) specifies an audio track identifier, a language of English, an audio type of original, one or more constituent audio streams, and stream-specific metadata for each of the constituent audio streams. In a similar fashion, the audio track description 350(2) describes an audio track that includes a
30 subset of the audio streams in the audio stream set 240(4). Referring back to Figure 2, the audio stream set 240(4) is an English audio description stream set.

[0134] The timed text track description 360(1) describes a timed text track that includes a subset of the timed text streams in the timed text stream set 260(4).

Referring back to Figure 2, the timed text stream set 260(4) is an English partial subtitle stream set. The timed text track description 360(2) describes a timed text track that includes a subset of the timed text streams in the timed text stream set 260(3). Referring back to Figure 2, the timed text stream set 260(3) is a Spanish full subtitle stream set.

[0135] The alternate video track summary 370(1) identifies an alternate video track that can be derived from the video stream set 220(3) and specifies any amount of the video metadata set 230(3) that corresponds to the video stream set 220(3). Referring back to Figure 2, the video stream set 220(3) is a Spanish localized video stream set. In some embodiments, the alternate video track summary 370(1) therefore specifies an alternate video track identifier that embeds an identifier for the video stream set 220(3), a language of Spanish, and a video type of localized.

[0136] The alternate video track summary 370(2) identifies an alternate video track that can be derived from the video stream set 220(2) and specifies any amount of the video metadata set 230(2) that corresponds to the video stream set 220(2). Referring back to Figure 2, the video stream set 220(2) is a French localized video stream set. In some embodiments, the alternate video track summary 370(2) therefore specifies an alternate video track identifier that embeds an identifier for the video stream set 220(2), a language of French, and a video type of localized.

[0137] Advantageously, the alternate video track summary 370(1) and the alternate video track summary 370(2) enable the endpoint application 180 to orchestrate a video switching process. During the video switching process, the endpoint application 180 switches from playing back the media title via a recommended presentation that includes the English original video track to playing back the media title via a recommended presentation that includes either a Spanish localized video track or a French localized video track.

Exemplar Alternate Manifest File

[0138] Figure 4 is a more detailed illustration of the alternate manifest file 178 of Figure 1, according to various embodiments. More specifically, the alternate manifest file 178 depicted in Figure 4 is an exemplar alternate manifest file. For

explanatory purposes, the manifest customization application 170 generates the alternate manifest file 178 based on an identifier for the alternate video track summary 370(2) of Figure 3, the inclusive media package 130 that is depicted in Figure 2, a preferred timed text language of English, and a preferred timed text type of subtitle.

[0139] More specifically, the manifest customization application 170 selects the video stream set 220(2) based on the identifier for the alternate video track summary 370(2). The manifest customization application 170 then generates the alternate manifest file 178 based on the video stream set 220(2), the inclusive media package 130, and a timed text preference of English subtitles. Referring back to Figure 2, the video stream set 220(2) is a French localized video stream set and therefore the manifest customization application 170 generates that alternate manifest file 178 based on a French localized video track.

[0140] As shown, the alternate manifest file 178 specifies, without limitation, a recommended server device 410, the media title description 320, a recommended presentation 430(1) - a recommended presentation 430(6), a video track description 440, an audio track description 450(1), an audio track description 450(2), a timed text track description 460(1), a timed text track description 460(2), an alternate video track summary 470(1), and an alternate video track summary 470(2). As shown, the recommended server device 410 is the CDN edge server device 150 of Figure 1. The media title description 320 is equal to the title metadata set 210 specified in the inclusive media package 130.

[0141] The recommended presentation 430(1) specifies a French localized video track that is associated with the video track description 440, a French dubbed audio track that is associated with the audio track description 450(1), and an English full subtitle track that is associated with the timed text track description 460(1). The recommended presentation 430(2) specifies the French localized video track, the French dubbed audio track, and a French partial subtitle track that is associated with the timed text track description 460(2). The recommended presentation 430(3) specifies the French localized video track and the French dubbed audio track.

[0142] The recommended presentation 430(4) specifies the French localized video track, an English original audio track that is associated with the video track description

440, a French audio description track that is associated with the audio track description 450(2), and the English full subtitle track. The recommended presentation 430(5) specifies the French localized video track, the French audio description track, and the French partial subtitle track. The recommended presentation 430(3) specifies
5 the French localized video track and the French audio description track.

[0143] The video track description 440 describes a French localized video track that includes a subset of the video streams in the video stream set 220(2). In some embodiments, the video track description 440 specifies a video track identifier, a language of French, a video type of localized, the constituent video streams, and
10 stream-specific metadata for each of the constituent video streams.

[0144] The audio track description 450(1) describes a French dubbed audio track that includes a subset of the audio streams in the audio stream set 240(2). The audio track description 450(2) describes a French audio description track that includes a subset of the audio streams in the audio stream set 240(5). The timed text track
15 description 460(1) describes an English full subtitle track that includes a subset of the timed text streams in the timed text stream set 260(1). The timed text track description 460(2) describes a French partial subtitle track that includes a subset of the timed text streams in the timed text stream set 260(5).

[0145] The alternate video track summary 470(1) identifies an alternate English
20 original video track that can be derived from the video stream set 220(1) and specifies any amount of the video metadata set 230(1) that corresponds to the video stream set 220(1). The alternate video track summary 470(2) identifies an alternate Spanish localized video track that can be derived from the video stream set 220(3) and specifies any amount of the video metadata set 230(3) that corresponds to the video
25 stream set 220(3).

Techniques for Providing Streams Associated with a Media Title for Playback

[0146] Figure 5 is a flow diagram of method steps for generating and deploying streams that are associated with a media title to a content delivery network, according to various embodiments. Although the method steps are described with reference to
30 the systems of Figures 1 - 4, persons skilled in the art will understand that any system configured to implement the method steps, in any order, falls within the scope of the embodiments.

[0147] As shown, a method 500 begins at step 502, where the ingest application 122 generates an initial version of an inclusive media package that includes a title metadata set based on a baseline video source file associated with a media title. At step 504, for each video source file associated with the media title, the ingest application 122 adds a video metadata set specifying a video type and a language to the inclusive media package

[0148] At step 506, for each audio source file associated with the media title, the ingest application 122 adds an audio metadata set specifying an audio type and a language to the inclusive media package. At step 508, for each of zero or more timed text source files associated with the media title, the ingest application 122 adds a timed text metadata set specifying a language, a timed text type, and an optional timed text subtype to the inclusive media package.

[0149] At step 510, the encoding application 124 generates one or more streams for each media source file, where each segment of each stream can be decoded independently. At step 512, the encoding application 124 propagates each media metadata set to the corresponding streams and generates a stream metadata set for each stream. At step 514, for each media source file, the encoding application 124 adds a stream set that includes the corresponding stream(s) and the associated stream metadata set(s) to the inclusive media package.

[0150] At step 516, the deployment application 126 generates a package metadata set for the media title based on the title metadata set, the media metadata sets, and the stream metadata sets. At step 518, the deployment application 126 transmits the package metadata set to a software application that generates customized manifest files for the media title. At step 520, the deployment application 126 stores the inclusive media package in an origin server device and deploys the stored inclusive media package from the origin server device to a CDN. The method 500 then terminates.

[0151] Figure 6 is a flow diagram of method steps for generating a manifest file that itemizes streams that are associated with a media title, according to various embodiments. Although the method steps are described with reference to the systems of Figures 1 - 4, persons skilled in the art will understand that any system

configured to implement the method steps, in any order, falls within the scope of the embodiments.

[0152] As shown, a method 600 begins at step 602, where the manifest customization application 170 receives a request for a manifest file from a client device. At step 604, the manifest customization application 170 acquires a package metadata set corresponding to an inclusive media package associated with the media title, a user metadata set, a client device metadata set, and CDN metadata set 134 based on the request.

[0153] At step 606, the manifest customization application 170 sets a recommended server device equal to an edge server device in a CDN based on the CDN metadata set 134 and the client device metadata set. At step 608, the manifest customization application 170 selects one of the video stream sets, one of the audio stream sets, and optionally one of the timed text stream sets that are in the inclusive media package based on the package metadata set and the user metadata set. At step 610, the manifest customization application 170 defines a first recommended presentation that identifies a video track derived from the selected video stream set and a different track derived from each of the other selected stream set(s).

[0154] At step 612, the manifest customization application 170 defines zero or more other recommended presentations that each identify the video track and a different combination of an audio track and an optional timed text track that are derived from the inclusive media package. At step 614, for each identified track, the manifest customization application 170 generates a track description specifying a subset of the streams in the corresponding stream set, relevant metadata, and locator(s) for the segment(s) of each specified stream

[0155] At step 616, the manifest customization application 170 generates a different alternate video track summary specifying a language and a video type for each of any number (including none) of the unselected video stream sets in the inclusive media package. At step 618, the manifest customization application 170 generates a manifest file specifying the recommended server device, the recommended presentation(s), the track descriptions, and the one or more alternate video track summaries. At step 620, the manifest customization application 170 transmits the manifest file to the client device. The method 600 then terminates.

[0156] Figure 7 is a flow diagram of method steps for generating a manifest file that enables video tracks to be switched during playback of a media title, according to various embodiments. Although the method steps are described with reference to the systems of Figures 1 - 4, persons skilled in the art will understand that any system
5 configured to implement the method steps, in any order, falls within the scope of the embodiments.

[0157] As shown, a method 700 begins at step 702, where the manifest customization application 170 transmits a manifest file that enables streaming of one video track and identifies one or more alternate video tracks to a client device. At
10 step 704, the manifest customization application 170 receives a request from the client device to generate an alternate manifest file that enables streaming of an alternate video track.

[0158] At step 706, the manifest customization application 170 acquires a package metadata set corresponding to an inclusive media package associated with
15 the media title, a user metadata set, a client device metadata set, and CDN metadata set 134. At step 708, the manifest customization application 170 selects the video stream set that corresponds to the requested alternate video track and selects the associated language based on the package metadata set.

[0159] At step 710, the manifest customization selects an audio stream sets that is
20 in the inclusive media package based on the selected language and the package metadata set. At step 712, the manifest customization application 170 optionally selects a timed text stream set that is in the inclusive media package based on the user metadata set, the package metadata set, and the selected language.

[0160] At step 714, the manifest customization application 170 defines a first
25 recommended presentation that identifies the recommended video track and a different track derived from each of the other selected stream set(s). At step 716, the manifest customization application 170 defines zero or more other recommended presentations that each identify the recommended video track and different combinations of an audio and an optional timed text track derived from the inclusive
30 media package.

[0161] At step 718, the manifest customization application 170 generates an alternate manifest file that enables the client device to stream the recommended presentation and therefore the corresponding tracks from an edge server in a CDN. At step 720, the manifest customization application 170 transmits the alternate
5 manifest file to the client device. The method 700 then terminates.

[0162] In sum, the disclosed techniques can be used to generate and deploy an inclusive media package for a media title that enables the media title to be streamed and played back via recommended presentations that are flexibly tailored for different users. In some embodiments, a media title is associated with an original language
10 video file, one or more other video files, one or more audio files, zero or more timed text files, and zero or more other media files. An ingest application included in a media processing pipeline generates a title metadata set identifying and describing any number of characteristics of the media title based on an original language video file associated with the media title. The ingest application adds the title metadata set
15 to an initially empty inclusive media package for the media title. For each media file (including the original language video file), the ingest application generates and adds a corresponding media metadata set to the inclusive media package. Notably, the media metadata sets describe relevant characteristics of, distinguish between, and define compatibility between the associated stream sets. In particular, each media
20 metadata set specifies a language. And each timed text metadata set can specify an optional timed text type indicating language-based compatibility with video stream sets.

[0163] An encoding application included in the media processing pipeline encodes each of the media files one or more times based on different encoding profiles to
25 generate one or more corresponding streams. The encoding application propagates each media metadata set to the corresponding streams and generates a stream metadata set for each stream. The stream metadata sets included in a given stream set describe relevant characteristics of and distinguish between, the associated streams. For each media metadata set, the encoding application generates a
30 corresponding stream set that includes the corresponding streams and stream metadata sets and adds the corresponding stream set to the inclusive media package.

[0164] A deployment application included in the media processing pipeline generates a package metadata set that includes the title metadata set, the media metadata sets, and the stream sets. The deployment application stores the package metadata set in a media services database that is accessible to a manifest

5 customization application and any number of other applications included in cloud-based media services. The deployment application stores the inclusive media package in an origin server device. A software application executing on the origin server device provides the inclusive media package on-request to software application executing on server devices included in a CDN.

10 **[0165]** In some embodiments, the manifest customization application 170 receives a request for a manifest file for the media title from an endpoint application executing on a client device that is associated with a user. In response, the manifest customization application 170 retrieves the package metadata set associated with the media title, a CDN metadata, client device metadata, and user metadata from the

15 media services database. The user can specify any number and/or types of preferences (e.g., a timed text language, a timed text type) associated with the user. The manifest customization application 170 selects one of the video stream sets in the inclusive media package based on the package metadata set and a preference (e.g., a preferred language, a preferred aspect ratio) associated with the user that is

20 specified in the user metadata. The manifest customization application 170 selects at least one of the video streams specified in the selected video stream set for inclusion in a video track and generates one or more recommended presentations based on the video track, the package metadata set, and the user metadata. Each recommended presentation includes the video track and a different combination of an

25 audio track and an optional timed text track. The manifest customization application 170 generates a different alternate video track summary specifying an alternate video track identifier, a language, and a video type based for each of any number (including none) of the unselected video stream sets in the inclusive media package

[0166] Subsequently, the manifest customization application 170 generates a

30 manifest file that specifies a recommended server device included in the CDN, the recommended presentations, a track description for each track specified in at least one recommended presentation, and zero or more alternate video track summaries. Each track description specifies a track identifier, a language, a media type, the

constituent streams, and stream-specific metadata for each constituent stream. The stream-specific metadata for a given constituent stream enables the endpoint application to request each segment of the constituent stream from the recommended server device. Each alternate video track summary provides an opportunity to switch
5 from streaming the video track specified in the recommended presentations to streaming a video track that is associated with a different language. The manifest customization application transmits the manifest file to the endpoint application.

[0167] In some embodiments, the endpoint application transmits a request for an alternate manifest file that specifies an alternate video track identifier to the manifest
10 customization application 170 while streaming and playback the media title via one of the recommended presentations. In response, the manifest customization application 170 selects the video stream set included in the inclusive media package based on the alternate video track identifier and the package metadata set. The manifest customization application 170 designates one or more of the video streams specified
15 in the selected video stream set as the alternate video track and generates one or more new recommended presentations that each include the alternate video track based on the package metadata set, and the user metadata. Subsequently, the manifest customization application 170 generates an alternate manifest file that specifies the recommended server device, the new recommended presentations, a
20 track description for each track specified in at least one new recommended presentation, and zero or more alternate video track summaries. The manifest customization application 170 then transmits the alternate manifest file to the endpoint application. Upon receiving the alternate manifest file, the endpoint application switches from streaming a recommended presentation specified in the manifest file to
25 streaming a recommended presentation specified in the alternate manifest file.

[0168] At least one technical advantage of the disclosed techniques relative to the prior art is that, with the disclosed techniques, source files are not redundantly processed by the media processing pipeline when generating and deploying a single inclusive media package that more comprehensively represents a media title to a
30 content delivery network relative to conventional video-specific packages. Therefore, the amounts of storage resources, processing resources, and time consumed in generating and deploying a media title can be substantially reduced relative to prior art approaches. Another technical advantage of the disclosed techniques is that,

unlike prior art approaches, the tracks specified in a manifest file generated using a single inclusive media package can be selected on-the-fly from all stream sets derived from all source files associated with a corresponding media title based on the preferences of a given user. Therefore, the disclosed techniques are able to generate

5 tracks for playback that more closely align with the preferences of users relative to what can be achieved using prior art approaches where manifest files are generated based on a subset of streams included in a single video-specific package. These technical advantages provide one or more technological advancements over prior art approaches.

10 **[0169]** 1. In some embodiments, a computer-implemented method for generating and distributing streams for playback comprises encoding a first video source file that is associated with a media title to generate a first plurality of video streams; encoding a second video source file that is associated with the media title to generate a second plurality of video streams; generating first video metadata associated with the first

15 plurality of video streams that includes a first set of values for a set of video characteristics; generating second video metadata associated with the second plurality of video streams that includes a second set of values for the set of video characteristics; generating a media package that includes the first plurality of video streams, the first video metadata, the second plurality of video streams, and the

20 second video metadata; and transmitting the media package to at least one server device.

[0170] 2. The computer-implemented method of clause 1, further comprising generating metadata associated with the media package upon which a manifest file that itemizes a video track derived from either the first plurality of video streams or the

25 second plurality of video streams is based.

[0171] 3. The computer-implemented method of clauses 1 or 2, wherein generating the first video metadata comprises specifying a different combination of a resolution and a bitrate for each video stream included in the first plurality of video streams.

30 **[0172]** 4. The computer-implemented method of any of clauses 1-3, wherein the set of video characteristics comprises at least one of a video type, a language, or an aspect ratio.

[0173] 5. The computer-implemented method of any of clauses 1-4, wherein the first video metadata and the second video metadata specify different languages.

[0174] 6. The computer-implemented method of any of clauses 1-5, wherein generating the media package comprises partitioning the first video source file into a plurality of segments; and generating metadata that is associated with the media title based on the first video source file and the plurality of segments.

[0175] 7. The computer-implemented method of any of clauses 1-6, wherein generating the media package comprises encoding an audio source file that is associated with the media title to generate one or more audio streams; and generating metadata associated with the one or more audio streams that includes a third set of values for a set of audio characteristics.

[0176] 8. The computer-implemented method of any of clauses 1-7, wherein generating the media package comprises encoding a timed text source file that is associated with the media title to generate one or more timed text streams; and generating metadata associated with the one or more timed text streams that includes a third set of values for a set of timed text characteristics.

[0177] 9. The computer-implemented method of any of clauses 1-8, wherein the metadata associated with the one or more timed text streams specifies at least a first language and indicates whether the one or more timed text streams are compatible with at least one video stream that is associated with the first language or at least one video stream that is not associated with the first language.

[0178] 10. The computer-implemented method of any of clauses 1-9, wherein the second video source file comprises a modified version of the first video source file.

[0179] 11. In some embodiments, one or more non-transitory computer readable media include instructions that, when executed by one or more processors, cause the one or more processors to generate and distribute streams for playback by performing the steps of encoding a first video source file that is associated with a media title to generate a first plurality of video streams; encoding a second video source file that is associated with the media title to generate a second plurality of video streams; generating first video metadata associated with the first plurality of video streams that includes a first set of values for a set of video characteristics;

generating second video metadata associated with the second plurality of video streams that includes a second set of values for the set of video characteristics; generating a media package that includes the first plurality of video streams, the first video metadata, the second plurality of video streams, and the second video
5 metadata; and transmitting the media package to at least one server device.

[0180] 12. The one or more non-transitory computer readable media of clause 11, further comprising generating metadata associated with the media package upon which a manifest file that itemizes a video track derived from either the first plurality of video streams or the second plurality of video streams is based.

10 **[0181]** 13. The one or more non-transitory computer readable media of clauses 11 or 12, wherein generating the first video metadata comprises specifying a different combination of a resolution and a bitrate for each video stream included in the first plurality of video streams.

15 **[0182]** 14. The one or more non-transitory computer readable media of any of clauses 11-13, wherein the set of video characteristics comprises at least one of a video type, a language, or an aspect ratio.

[0183] 15. The one or more non-transitory computer readable media of any of clauses 11-14, wherein the first video metadata and the second video metadata specify different aspect ratios.

20 **[0184]** 16. The one or more non-transitory computer readable media of any of clauses 11-15, wherein generating the media package comprises encoding an audio source file that is associated with the media title to generate one or more audio streams; and generating metadata associated with the one or more audio streams that includes a third set of values for a set of audio characteristics.

25 **[0185]** 17. The one or more non-transitory computer readable media of any of clauses 11-16, wherein the metadata associated with the one or more audio streams specifies at least a first language and an audio type of original, dubbed, or audio description.

30 **[0186]** 18. The one or more non-transitory computer readable media of any of clauses 11-17, wherein generating the media package comprises encoding a timed

text source file that is associated with the media title to generate one or more timed text streams; and generating metadata associated with the one or more timed text streams that includes a third set of values for a set of timed text characteristics.

5 **[0187]** 19. The one or more non-transitory computer readable media of any of clauses 11-18, wherein the metadata associated with the one or more timed text streams specifies at least a first language and a timed text type of subtitle, closed caption, or forced narrative.

10 **[0188]** 20. In some embodiments, a system comprises one or more memories storing instructions and one or more processors coupled to the one or more memories that, when executing the instructions, perform the steps of encoding a first video source file that is associated with a media title to generate a first plurality of video streams; encoding a second video source file that is associated with the media title to generate a second plurality of video streams; generating first video metadata associated with the first plurality of video streams that includes a first set of values for
15 a set of video characteristics; generating second video metadata associated with the second plurality of video streams that includes a second set of values for the set of video characteristics; generating a media package that includes the first plurality of video streams, the first video metadata, the second plurality of video streams, and the second video metadata; and transmitting the media package to at least one server
20 device.

[0189] Any and all combinations of any of the claim elements recited in any of the claims and/or any elements described in this application, in any fashion, fall within the contemplated scope of the present invention and protection.

25 **[0190]** The descriptions of the various embodiments have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments.

30 **[0191]** Aspects of the present embodiments may be embodied as a system, method or computer program product. Accordingly, aspects of the present disclosure may take the form of an entirely hardware embodiment, an entirely software

embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “module” or “system.” Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or
5 more computer readable medium(s) having computer readable program code embodied thereon.

[0192] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may
10 be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access
15 memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory, Flash memory, an optical fiber, a portable compact disc read-only memory, an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in
20 connection with an instruction execution system, apparatus, or device.

[0193] Aspects of the present disclosure are described above with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the disclosure. It will be understood that each block of the flowchart illustrations and/or block diagrams, and
25 combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine. The instructions, when executed via the processor of the computer or other
30 programmable data processing apparatus, enable the implementation of the functions/acts specified in the flowchart and/or block diagram block or blocks. Such processors may be, without limitation, general purpose processors, special-purpose processors, application-specific processors, or field-programmable gate arrays.

[0194] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present disclosure. In this regard, each block in the flowchart or block diagrams may

5 represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may

10 sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware

15 and computer instructions.

[0195] While the preceding is directed to embodiments of the present disclosure, other and further embodiments of the disclosure may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

WHAT IS CLAIMED IS:

1. A computer-implemented method for generating and distributing streams for playback, the method comprising:
 - encoding a first video source file that is associated with a media title to
 - 5 generate a first plurality of video streams;
 - encoding a second video source file that is associated with the media title to
 - generate a second plurality of video streams;
 - generating first video metadata associated with the first plurality of video
 - streams that includes a first set of values for a set of video
 - 10 characteristics;
 - generating second video metadata associated with the second plurality of
 - video streams that includes a second set of values for the set of video
 - characteristics;
 - generating a media package that includes the first plurality of video streams,
 - 15 the first video metadata, the second plurality of video streams, and the
 - second video metadata; and
 - transmitting the media package to at least one server device.
2. The computer-implemented method of claim 1, further comprising generating
- 20 metadata associated with the media package upon which a manifest file that itemizes
- a video track derived from either the first plurality of video streams or the second
- plurality of video streams is based.
3. The computer-implemented method of claim 1, wherein generating the first
- 25 video metadata comprises specifying a different combination of a resolution and a
- bitrate for each video stream included in the first plurality of video streams.
4. The computer-implemented method of claim 1, wherein the set of video
- characteristics comprises at least one of a video type, a language, or an aspect ratio.
- 30
5. The computer-implemented method of claim 1, wherein the first video
- metadata and the second video metadata specify different languages.
6. The computer-implemented method of claim 1, wherein generating the media

package comprises:

partitioning the first video source file into a plurality of segments; and
generating metadata that is associated with the media title based on the first
video source file and the plurality of segments.

5

7. The computer-implemented method of claim 1, wherein generating the media
package comprises:

encoding an audio source file that is associated with the media title to generate
one or more audio streams; and

10 generating metadata associated with the one or more audio streams that
includes a third set of values for a set of audio characteristics.

8. The computer-implemented method of claim 1, wherein generating the media
package comprises:

15 encoding a timed text source file that is associated with the media title to
generate one or more timed text streams; and
generating metadata associated with the one or more timed text streams that
includes a third set of values for a set of timed text characteristics.

20 9. The computer-implemented method of claim 8, wherein the metadata
associated with the one or more timed text streams specifies at least a first language
and indicates whether the one or more timed text streams are compatible with at least
one video stream that is associated with the first language or at least one video
stream that is not associated with the first language.

25

10. The computer-implemented method of claim 1, wherein the second video
source file comprises a modified version of the first video source file.

30 11. One or more non-transitory computer readable media including instructions
that, when executed by one or more processors, cause the one or more processors to
generate and distribute streams for playback by performing the steps of:

encoding a first video source file that is associated with a media title to
generate a first plurality of video streams;
encoding a second video source file that is associated with the media title to

- generate a second plurality of video streams;
generating first video metadata associated with the first plurality of video
streams that includes a first set of values for a set of video
characteristics;
5 generating second video metadata associated with the second plurality of
video streams that includes a second set of values for the set of video
characteristics;
generating a media package that includes the first plurality of video streams,
the first video metadata, the second plurality of video streams, and the
10 second video metadata; and
transmitting the media package to at least one server device.
12. The one or more non-transitory computer readable media of claim 11, further
comprising generating metadata associated with the media package upon which a
15 manifest file that itemizes a video track derived from either the first plurality of video
streams or the second plurality of video streams is based.
13. The one or more non-transitory computer readable media of claim 11, wherein
generating the first video metadata comprises specifying a different combination of a
20 resolution and a bitrate for each video stream included in the first plurality of video
streams.
14. The one or more non-transitory computer readable media of claim 11, wherein
the set of video characteristics comprises at least one of a video type, a language, or
25 an aspect ratio.
15. The one or more non-transitory computer readable media of claim 11, wherein
the first video metadata and the second video metadata specify different aspect
ratios.
30
16. The one or more non-transitory computer readable media of claim 11, wherein
generating the media package comprises:
encoding an audio source file that is associated with the media title to generate
one or more audio streams; and

generating metadata associated with the one or more audio streams that includes a third set of values for a set of audio characteristics.

17. The one or more non-transitory computer readable media of claim 16, wherein
5 the metadata associated with the one or more audio streams specifies at least a first language and an audio type of original, dubbed, or audio description.

18. The one or more non-transitory computer readable media of claim 11, wherein
generating the media package comprises:
10 encoding a timed text source file that is associated with the media title to generate one or more timed text streams; and
generating metadata associated with the one or more timed text streams that includes a third set of values for a set of timed text characteristics.

15 19. The one or more non-transitory computer readable media of claim 18, wherein the metadata associated with the one or more timed text streams specifies at least a first language and a timed text type of subtitle, closed caption, or forced narrative.

20. A system comprising:
20 one or more memories storing instructions; and
one or more processors coupled to the one or more memories that, when executing the instructions, perform the steps of:
encoding a first video source file that is associated with a media title to generate a first plurality of video streams;
25 encoding a second video source file that is associated with the media title to generate a second plurality of video streams;
generating first video metadata associated with the first plurality of video streams that includes a first set of values for a set of video characteristics;
30 generating second video metadata associated with the second plurality of video streams that includes a second set of values for the set of video characteristics;
generating a media package that includes the first plurality of video streams, the first video metadata, the second plurality of video

streams, and the second video metadata; and
transmitting the media package to at least one server device.