IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA
ALEXANDRIA DIVISION

AUDIO POD IP, LLC, Plaintiff, v. AMAZON.COM, INC., AMAZON.COM LLC, AND AMAZON WEB SERVICES, INC., Defendants.	Case No. 1:24-cv-00915 Jury Trial Demanded

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Audio Pod IP, LLC ("Audio Pod") files this complaint against Amazon.com, Inc., Amazon.com LLC, and Amazon Web Services, Inc. (hereinafter collectively "Amazon" or "Defendants") for infringement of United States Patent Nos. 9,319,720; 9,954,922; 10,091,266; and 10,735,488 (the "Patents-in-Suit"), attached here as Exhibits 1-4. Audio Pod alleges that Defendants have willfully and/or otherwise infringed one or more of the Patents-in-Suit.

NATURE OF THE ACTION

1. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§1 *et seq*.

THE PARTIES

2. Audio Pod is a limited liability company organized under the laws of the Commonwealth of Virginia with its principal place of business at 8609 Westwood Center Drive, Suite 110, Tysons Corner, Virginia 22182.

Audio Pod EX2002
Amazon v. Audio Pod
IPR2025-01041

3. On information and belief, defendant Amazon.com, Inc. is a corporation organized and existing under the laws of the state of Delaware with a principal place of business at 410 Terry Ave N, Seattle, Washington 98109-5210.

4. On information and belief, Amazon.com, Inc. may be served with process through its registered agent, Corporation Service Company, 251 Little Falls Drive, Wilmington, Delaware 19808, or anywhere it may be found.

5. Amazon.com, Inc. does business across the United States, including in the Commonwealth of Virginia and, more specifically, in the Eastern District of Virginia.

6. On information and belief, defendant Amazon.com LLC is a limited liability corporation organized and existing under the laws of the state of Delaware and a wholly-owned subsidiary of Amazon.com, Inc., with a principal place of business at 410 Terry Ave N, Seattle, Washington 98109-5210.

7. On information and belief, Amazon.com LLC may be served with process through its registered agent, Corporation Service Company, 251 Little Falls Drive, Wilmington, Delaware 19808, or anywhere it may be found.

8. Amazon.com LLC does business across the United States, including in the Commonwealth of Virginia and, more specifically, in the Eastern District of Virginia.

9. On information and belief, defendant Amazon Web Services, Inc. ("AWS") is a corporation organized and existing under the laws of the state of Delaware with a principal place of business at 410 Terry Ave N, Seattle, Washington 98109-5210.

10. AWS is a subsidiary and controlled affiliate of defendant Amazon.com, Inc. and a so-called Amazon Group Company.

11. On information and belief, AWS may be served with process through its registered agent, Corporation Service Company, 100 Shockoe Slip, 2nd Floor, Richmond, VA, 23219 - 4100 or anywhere it may be found.

12. AWS does business across the United States, including in the Commonwealth of Virginia and, more specifically, in the Eastern District of Virginia.

13. On information and belief, AWS has been authorized to transact business in the Commonwealth of Virginia and the Eastern District of Virginia since on or about January 25, 2013, under Virginia Entity ID F1918947.

14. On information and belief, Defendants sell and offers to sell products and services throughout the Commonwealth of Virginia, including in this judicial district, as well as throughout the United States, and introduces products and services that perform infringing processes into the stream of commerce knowing that they would be used, offered for sale, or sold in this judicial district and elsewhere in the United States.

15. On information and belief, Amazon has made, used, offered to sell, offered to sell access to, sold, and/or sold access to products and services, including the following specifically accused products and services: (1) Amazon Video and Amazon Prime Video (collectively "Prime Video"); (2) Amazon Music, Amazon Prime Music, and Amazon Music Prime (collectively "Prime Music"); (3) current or legacy products or services, which use, or have used, one or more of the foregoing products and services as a component product or component service; (4) combinations of products and/or services comprising, in whole or in part, two or more of the foregoing products and services; and (5) all other current or legacy products and services imported, made, used, sold, or offered for sale by Amazon that operate, or have operated in a substantially similar manner as the above-listed products and services. (As used herein, one

or more of the forgoing products and services are individually and collectively referred to as the accused "Amazon Products and Services").

16. On information and belief, Amazon, as well as the hardware and software components comprising the Amazon Products and Services and/or that enable the Amazon Products and Services to operate, including but not limited to servers, server software, webserver software, webserver hardware, email server hardware, email server software, website client software, mobile computing device client application software, networked communications hardware, network routers, network switches, network hubs, WIFI access point hardware, WIFI access point software, point-of-sale hardware, point-of-sale software, back-end hardware, back-end software, cloud-based software, cloud-based hardware, and other hardware and software computing systems and components infringes (literally and/or under the doctrine of equivalents) at least one claim of each of the Patents-in-Suit.

JURISDICTION AND VENUE

17. This civil action arises under the Patent Laws of the United States, 35 U.S.C. § 1 *et seq*. Accordingly, this Court has subject matter jurisdiction under at least 28 U.S.C. §§ 1331 and 1338(a).

18. This Court has general and specific personal jurisdiction over the Defendants because it regularly conducts and solicits business, or otherwise engages in other persistent courses of conduct in this judicial district, and/or derives substantial revenue from the use, sale, and distribution of goods and services, including but not limited to the accused Amazon Products and Services provided to individuals and businesses in the Eastern District of Virginia.

19. On information and belief, Amazon infringes the Patents-in-Suit in the Eastern District of Virginia, at least, by making, using, offering to sell access to, and/or selling access to the accused Amazon Products and Services in this district.

20. Amazon is the world's largest online retailer and marketplace and provider of cloud computing services through AWS. Amazon distributes a variety of downloadable and streaming content through its Amazon Prime Video and Amazon Music services. Amazon also produces retail consumer electronics including the Amazon Echo and Amazon Alexa devices.

21. On information and belief, Amazon is the second largest private employer in the United States. According to the Virginia Economic Development Partnership, Amazon has since 2010 invested more than $109 billion in Virginia, including infrastructure and compensation to employees, and has created more than 36,000 jobs in the Commonwealth.[1]

22. Amazon officially opened its "HQ2"—*i.e.*, its second headquarters—in Arlington, Virginia and plans to add more than 25,000 new jobs to the more than 30,000 employees it already has in the Virginia and Washington, D.C. metro area.[2] Amazon's new headquarters are within this judicial district in the National Landing neighborhood of Arlington, Virginia. According to Amazon, the new Arlington campus features energy-efficient offices, neighborhood retail, and new public and green spaces including 1.1 acres of new public open space, designed for a variety of uses, including a dog park, recreation areas, farmers markets, and more to help realize the community's vision for a large, centrally-located park. *Id*.

[1] *See* https://www.vedp.org/press-release/2023-09/amazon-virginiabeach#:~:text=Since%202010%2C%20Amazon%20has%20invested,direct%20jobs%20in%20the%20Commonwealth.
[2] See https://www.aboutamazon.com/workplace/corporate-offices.

23. On information and belief, the accused Amazon Products and Services are made, used, sold and offered for sale by Amazon throughout the Eastern District of Virginia.

24. On information and belief, Amazon customers located in the Eastern District of Virginia have obtained access to and used the accused Amazon Products and Services while located in the Eastern District of Virginia.

25. This Court has personal jurisdiction over Amazon because, inter alia, Amazon, on information and belief: (1) has committed acts of patent infringement in this Eastern District of Virginia; (2) maintains a regular and established place of business, namely its HQ2 in Arlington, within the Eastern District of Virginia; (3) has substantial, continuous, and systematic contacts with this Commonwealth and the Eastern District of Virginia; (4) owns, manages, and operates facilities in this Commonwealth and the Eastern District of Virginia; (5) enjoys substantial income from its operations and sales in this Commonwealth and the Eastern District of Virginia; (6) employs Virginia residents in this Commonwealth and the Eastern District of Virginia, and (7) solicits business using the Amazon Products and Services in this Commonwealth and the Eastern District of Virginia.

26. On April 9, 2020, this Court held:

> It must be said that Amazon is nothing if not ubiquitous in the United States. Furthermore, after considering 238 cities, Amazon chose Arlington in the Eastern District of Virginia as the location for its HQ2 and will invest $2.5 billion and 25,000 jobs in the undertaking. As such, Amazon cannot in good faith represent to the Court that E.D. Va. is an undesirable or inconvenient location to operate and do business. Litigating should not be an additional significant strain.

Maglula, Ltd. v. Amazon.com, Inc., No. 1:19-cv-01570, ECF No. 52 at 32-33 (E.D. Va. Apr. 9, 2020).

27. Venue is proper pursuant to 28 U.S.C. §§ 1391 and/or 1400(b), at least because Amazon has committed acts of infringement in this judicial district, and has a regular and

established places of business in this judicial district. Venue is also proper for the reasons set forth by the Court in its *Maglula* decision. *See* 1:2019-cv-01570 (E.D. Va. Apr. 9, 2020), D.I. 52.

28. In fact, Amazon has already admitted that venue is proper in this District. In *Amazon.com, Inc. v. WDC Holdings LLC*, No. 1:20-cv-484, ECF No. 1, ¶ 26 (E.D. Va. Apr. 27, 2020), Amazon argued that venue in this district was proper because "it is a district in which Plaintiff [Amazon] maintains headquarters and/or substantial business operations…".

THE ASSERTED PATENTS

U.S. Patent No. 9,319,720

29. On April 19, 2016, the USPTO duly and legally issued United States Patent No. 9,319,720 ("the '720 patent") entitled "System and method for rendering digital content using time offsets" to inventors John McCue, Robert McCue, Gregory Shostakovsky, and Glenn McCue.

30. The '720 patent is presumed valid under 35 U.S.C. § 282.

31. Audio Pod owns all rights, title, and interest in the '720 patent.

32. Audio Pod has not granted Defendants an approval, an authorization, or a license to the rights under the '720 patent.

33. The '720 patent relates to, among other things, the delivery, management and rendering of multiple segmented streams of digital content in a synchronized rendering in dependence of the timeline of a media stream defined in a virtual media descriptor.

34. The '720 patent also describes a descriptor file, which according to the claims includes "time information for each digital data file in each media stream, the time information for synchronizing the plurality of media streams and determined relative to a timeline of an audio recording." '720 patent, Col. 40, lines 32-36; *see also* Abstract.

35. The patent further notes that the "digital content to be rendered is selected using the time information in the descriptor file and a time offset external to the descriptor file." *Id*.

36. The '720 patent solves, among other things, problems with, and improves upon, the speed and availability of access to digital content by a consumer. Further, it allows the consumer to select the structure of the multimedia rendering takes and customize it as needed.

U.S. Patent No. 9,954,922

37. On April 24, 2018, the USPTO duly and legally issued United States Patent No. 9,954,922 ("the '922 patent") entitled "Method and system for rendering digital content across multiple client devices" to inventors John McCue, Robert McCue, Gregory Shostakovsky, and Glenn McCue.

38. The '922 patent is presumed valid under 35 U.S.C. § 282.

39. Audio Pod owns all rights, title, and interest in the '922 patent.

40. Audio Pod has not granted Defendants an approval, an authorization, or a license to the rights under the '922 patent.

41. The '922 patent relates to, among other things, "segmenting an audio stream into a plurality of small digital audio files." '922 patent, Col. 2, 43-46.

42. These small digital audio files are then "transmitted, loaded, and played, in a specific order, such that from the user's perspective, the audio stream is reproduced in an apparently seamless manner." *Id*. at 46-49.

43. This manner of creating digital audio files also allows for media to be consumed by an end user without significant delay and in a timely manner. *Id*. at 50-58.

44. The invention of the ’922 patent also includes a descriptor, “which includes a record of the position of each small digital audio file in the audio stream, to increase tracking options.” *Id*. at 60-63.

U.S. Patent No. 10,091,266

45. On October 2, 2018, the USPTO duly and legally issued United States Patent No. 10,091,266 (“the ’266 patent”) entitled “Method and system for rendering digital content across multiple client devices” to inventors John McCue, Robert McCue, Gregory Shostakovsky, and Glenn McCue.

46. The ’266 patent is presumed valid under 35 U.S.C. § 282.

47. Audio Pod owns all rights, title, and interest in the ’266 patent.

48. Audio Pod has not granted Defendants an approval, an authorization, or a license to the rights under the ’266 patent.

49. The ’266 patent specification is nearly identical to the specification of the ’922 patent and solves the problems described in the ’922 patent.

U.S. Patent No. 10,735,488

50. On October 13, 2020, the USPTO duly and legally issued United States Patent No. 10,735,488 (“the ’488 patent”) entitled “Method of downloading digital content to be rendered” to inventors John McCue, Robert McCue, Gregory Shostakovsky, and Glenn McCue.

51. The ’488 patent is presumed valid under 35 U.S.C. § 282.

52. Audio Pod owns all rights, title, and interest in the ’488 patent.

53. Audio Pod has not granted Defendants an approval, an authorization, or a license to the rights under the ’488 patent.

54. The ’488 patent relates to, among other things, “tracking service level statistics … for content servers,” and “in the event of a degradation in service from the first content server, selecting a second content server to replace the first content server … wherein the server replacement is imperceptible to a user of the client device.” ’488 patent, Col. 16, 36-50.

55. The ’488 patent specification is nearly identical to the specification of the ’922 patent and solves the problems described in the ’922 patent.

BACKGROUND OF THE INVENTIONS

56. Computer scientists and brothers John, Robert and Glenn McCue, along with engineer Gregory Shostakovsky, are inventors on the Patents-in-Suit. The McCues, all software architects, conceptualized the streaming audiobook idea to help their mother enjoy literature in spite of her failing eyesight. The entrepreneurs incorporated Audio Pod Inc. in 2005, with Gregory Shostakovsky as CEO, and went on to invent several key media streaming technologies.

57. By proposing streaming functionalities where users could simply connect to a server and listen to an audio file, their solutions solved the cumbersome process that, at the time, required users to download audio files, store the files on their computers, and find a player to play such files. They invented, among other things, means to segment and sequence the streaming files so that these files could be streamed irrespective of their sizes, a major limitation to downloading content at the time (when large book downloads were taking hours over slow networks).

58. Further, they invented the idea of bookmarking the digital content such that a stream could be paused and played at a later time, even on a different device. Additionally, Audio Pod developed a subscriber service where users could stream content for a monthly

service rather than purchasing whole downloads. In 2006, they launched a subscriber-paid service to stream audiobooks to consumer devices.

59. Audio Pod's innovative concepts and early technology development was captured on the front-page of The Ottawa Citizen newspaper in January 2008.



BLACK AND WHITE AND HEARD ALL OVER

From left, brothers John, Robert, Glenn McCue and Gregory Shostakovsky have created software call Audio Pod, which allows audio books to be enjoyed more easily through laptops, phones and other devices.

Entrepreneurs tap into growing audio book market

Electronic bookmark among Ottawa company's innovations

BY JEFF BUCKSTEIN

Ten years ago John McCue began looking for a way to help his mother, Monica, continue her lifelong love affair with literature in spite of her failing eyesight.

Now Mr. McCue's Ottawa firm is ready to market innovative technology that allows users to easily listen to audio books and other spoken word streams on their laptop computers, portable PDAs, or any device that can play an MP3 file.

And as a bonus, the software program offers an electronic bookmark system that lets users stop listening and then return to the book exactly where they left off — even on a different device.

That adaptability and ease of use are key signatures of the patent-pending technology offered by Audio Pod Inc., which Mr. McCue, its senior software architect, co-founded with Gregory Shostakovsky in 2006. Robert McCue and Glenn McCue, John's brothers, are also software architects with the firm.

The technology is "a significant software development that has taken a lot of hours of work" Mr. Shostakovsky claims proudly. "No one has even been close to anything like this."

Because Audio Pod is a pure software design, there is no need for the user to manually download, store and transfer audio content. There are no CDs to manage or files to transfer.

It taps into a hot market. According to the Audio Publishers Association, sales of audio books in 2006 in the U.S. alone were estimated at $925 million, up six per cent over 2005.

"What's wonderful about audio books is that as more people discover them (they) pass on the word and we continue to grow," says association president Michele Cobb, who is based in North Kingstown, Rhode Island.

This tremendous growth in popularity suits a busy society of time-pressed people who like to read books "and still be able to multi-task," she adds.

The entrepreneurs stress that no matter how large an audio file is, it is accessible in a fast, reliable manner using their technology, and can also easily be bookmarked and picked up again wherever and whenever the reader chooses.

See AUDIO on PAGE E4

See Exhibit 5.

CONTACT WITH AMAZON

60. On information and belief, Amazon acquired audiobook publisher Brilliance Audio in May 2007.

61. A few months after this acquisition, in July 2007, Audio Pod met with Brilliance Audio, the then-leading audiobook publisher selling books on cassette tapes and CDs. During this meeting, Audio Pod demonstrated its technology in real-time and explained ways in which Brilliance could revolutionize its product offering using the Audio Pod technology.

62. Within eighteen months of that meeting, Amazon had incorporated much of the technology Audio Pod disclosed to Brilliance into its Kindle and Audible platforms.

63. After the Brilliance Audio acquisition, Audio Pod approached Amazon several times to collaborate on or acquire the Audio Pod technology.

64. In or about December 2012 and January 2013, Audio Pod CEO Gregory Shostakovsky wrote to Amazon's Vice President of IP Acquisitions:

> I would like to draw your attention to the latest Kindle product offering and the Intellectual Property owned by Audio Pod Inc. In our opinion, there is a marked similarity between some of the newest features contained within the Kindle and some of our Intellectual property. Specifically in technology areas related to "Whispersynch for Voice" and "Immersion Reading," among others.

65. As of the date of the filing of the instant Complaint, Amazon has never responded to any of Audio Pod's attempts to engage in licensing discussions.

66. Amazon has been aware of the Audio Pod technology and the Patents-in-Suit since as early as 2007 and no later than 2013 and has continued to willfully infringe, thereby warranting enhanced damages and attorneys' fees as set forth more fully below.

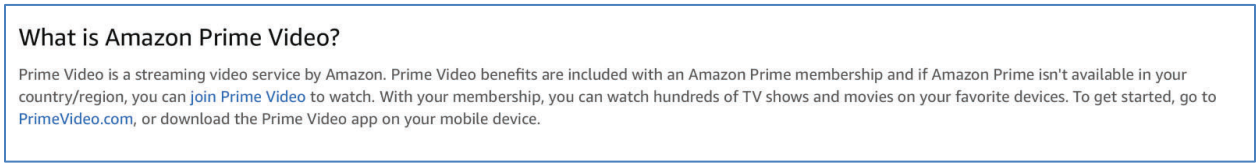
CLAIMS FOR RELIEF

COUNT I - Infringement of the ’720 patent by Prime Video

67. Audio Pod repeats, realleges, and incorporates by reference, as if fully set forth here, the allegations of the preceding paragraphs above.

68. On information and belief, Defendants (or those acting on their behalf) make, use, sell, sell access to, import, offer to sell and/or offer to sell access to Prime Video in the United States that infringes (literally and/or under the doctrine of equivalents) at least claim 1 of the ’720 patent.

69. On information and belief, one or more components of Prime Video provide a method for rendering digital content.

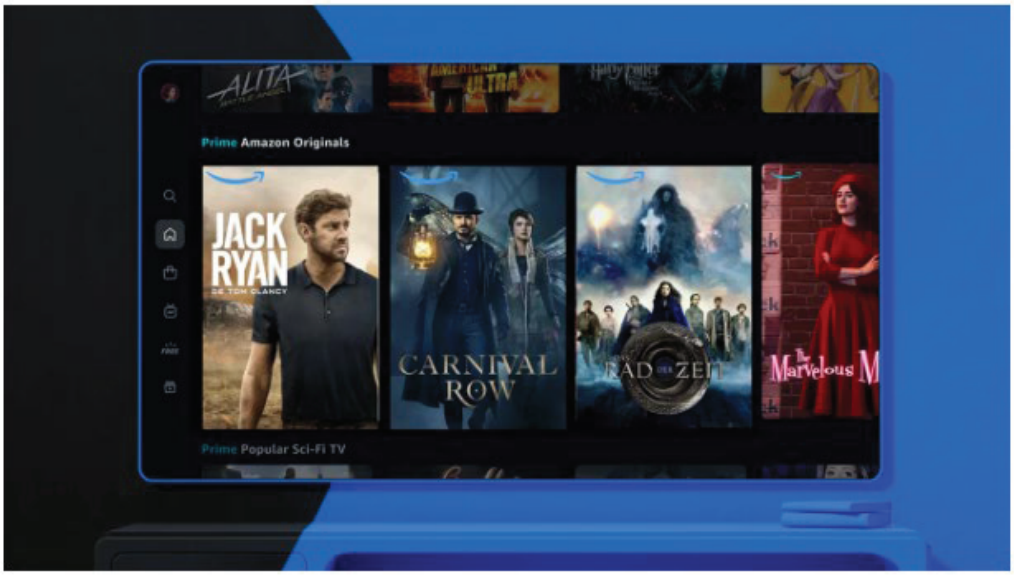


See https://www.primevideo.com/region/na/splash/getTheApp/ref=atv_hp_cnt.

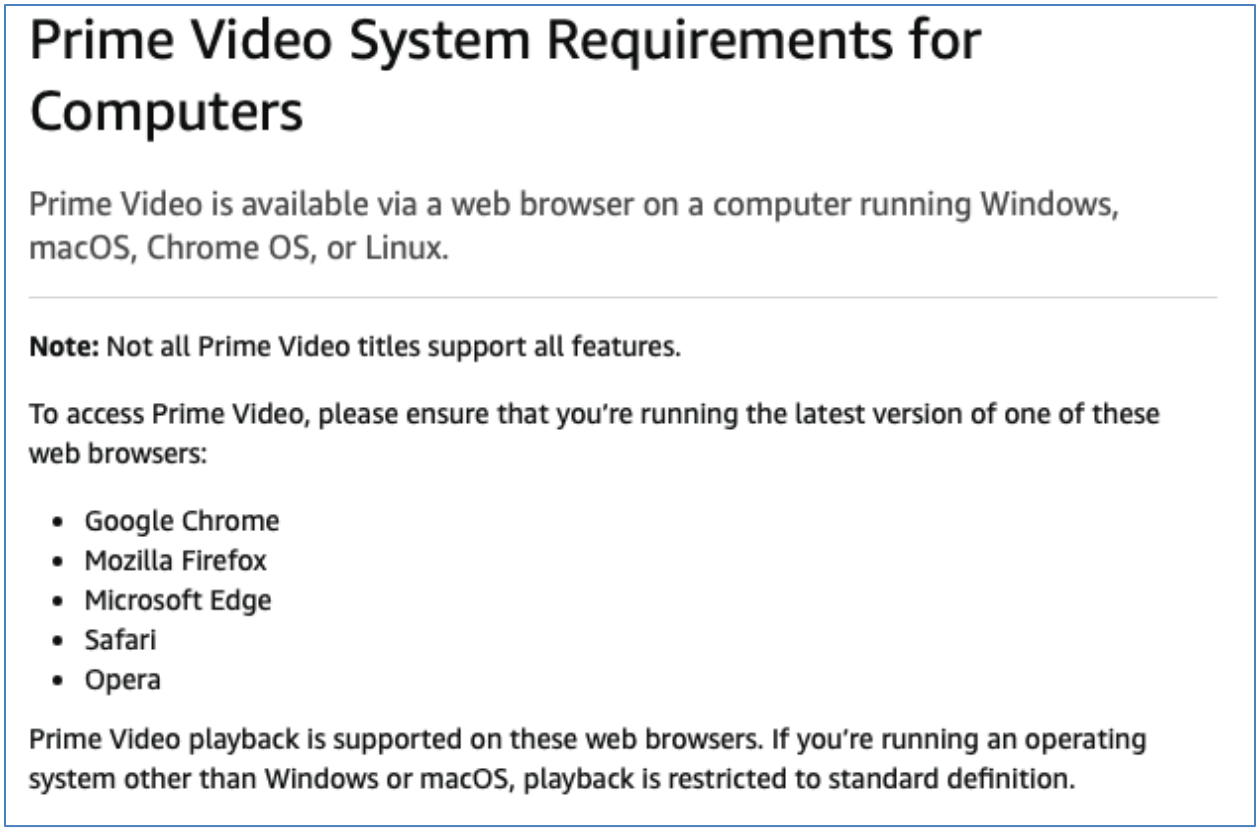
Prime Video offers a massive library of movies, series, and sports.

Read this article in Spanish.

Prime Video is home to more content than you might realize. If you're ready to stream blockbuster movies, hit shows, award-winning Amazon Originals, premium channels, and sports, read on for everything you need to know about Prime Video.



See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.



Prime Video System Requirements for Computers

Prime Video is available via a web browser on a computer running Windows, macOS, Chrome OS, or Linux.

Note: Not all Prime Video titles support all features.

To access Prime Video, please ensure that you're running the latest version of one of these web browsers:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari
- Opera

Prime Video playback is supported on these web browsers. If you're running an operating system other than Windows or macOS, playback is restricted to standard definition.

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GUVGB3QMQRYRERYW.

70. On information and belief, one or more components of Prime Video provide a method for rendering digital content comprising providing a media player having access to at least one server via a network (*e.g.*, the Prime Video player is accessed by a client device and accesses the Prime Video backend servers), the at least one server having stored thereon a descriptor file and a plurality of media streams (*e.g.*, Prime Video contains, e,g., the video component and the audio component (plurality of streams)) derived from a same originating written work (*e.g.*, Prime Video uses MPEG-DASH for adaptive streaming, which can include, e.g., video streams, audio streams, and subtitle streams corresponding to a written artistic work (e.g., a script upon which a movie is based)), each media stream including a plurality of digital data files (*e.g.*, each media stream is broken down into fragments (e.g., a plurality of digital data files)), the descriptor file

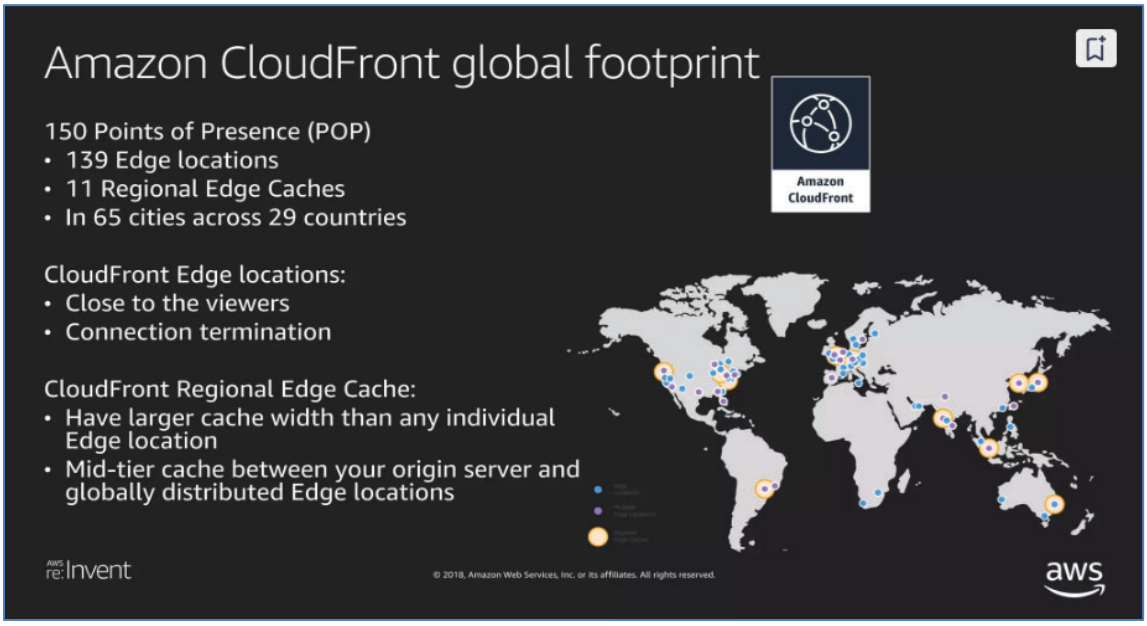
including information allowing a simultaneous synchronized rendering of the plurality of media streams (*e.g.*, Prime Video (using, e.g., MPEG-DASH) synchronizes video, audio, and subtitles (media streams)) that provides a literary experience of the originating written work (e.g., a storyline experience in the form of a movie that is based on a script).

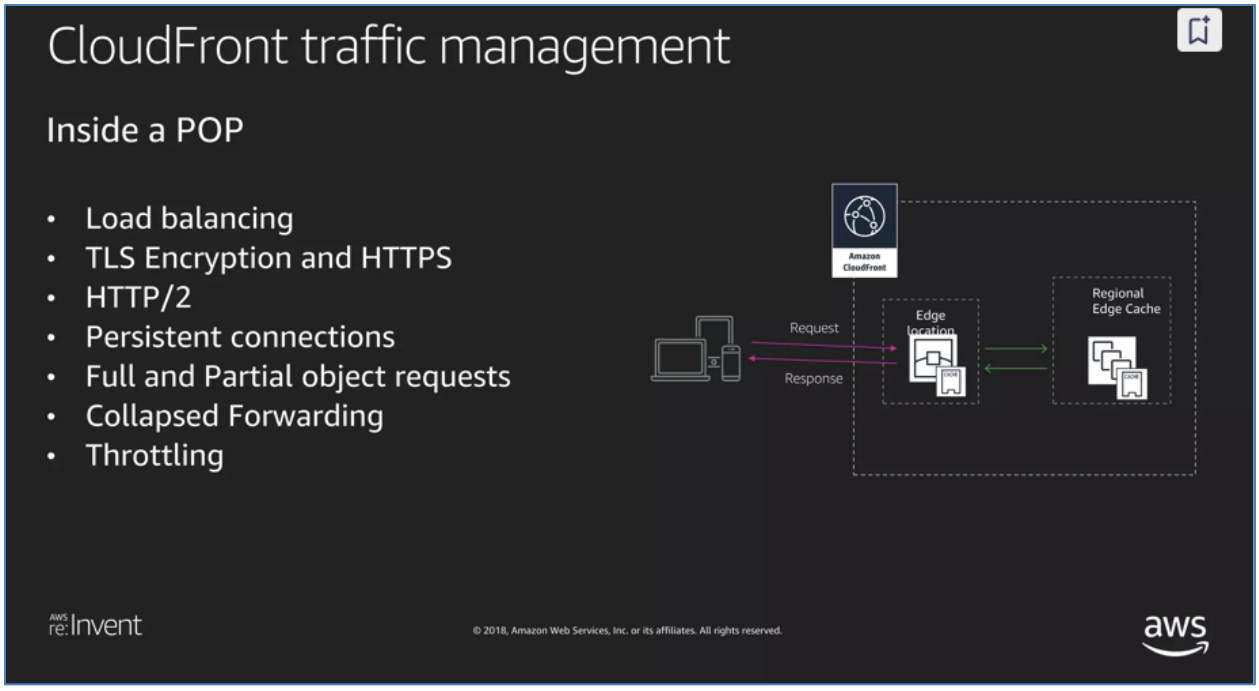
Amazon Prime Video: Delivering the Amazing Video Experience (CTD203-R1) - AWS re:Invent 2018

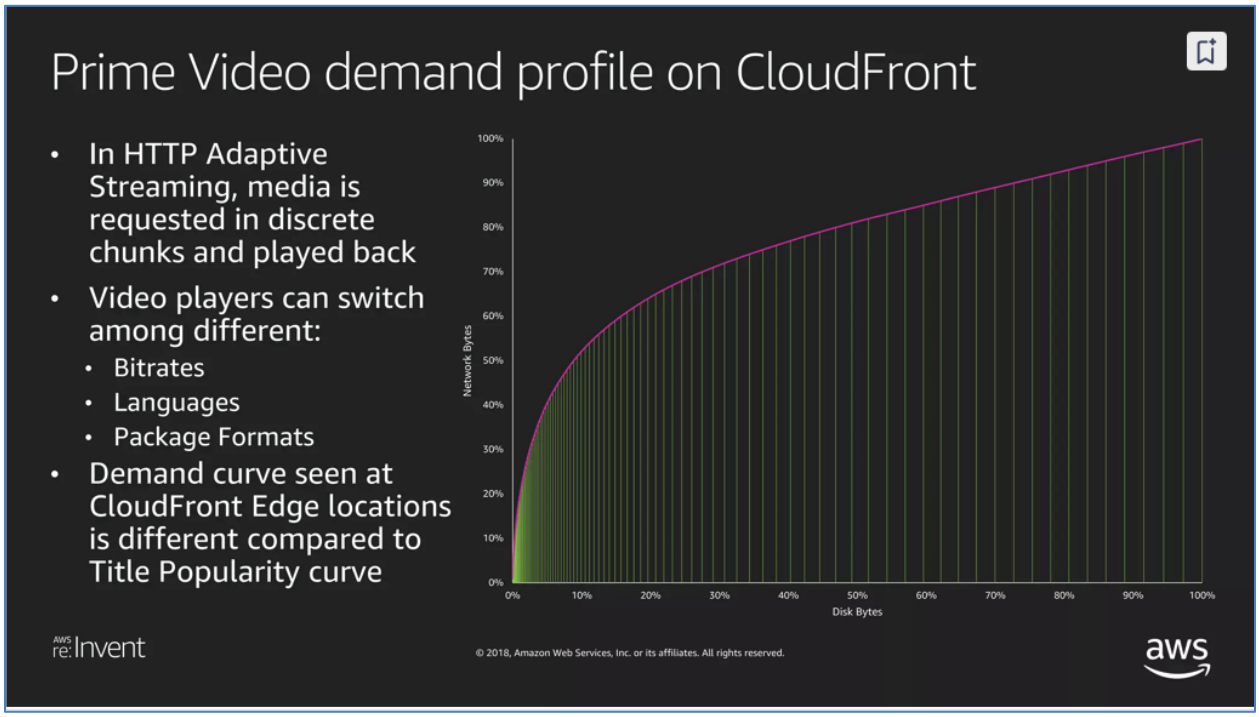
Nov 29, 2018 • 4 likes • 12,604 views

Amazon Web Services Follow

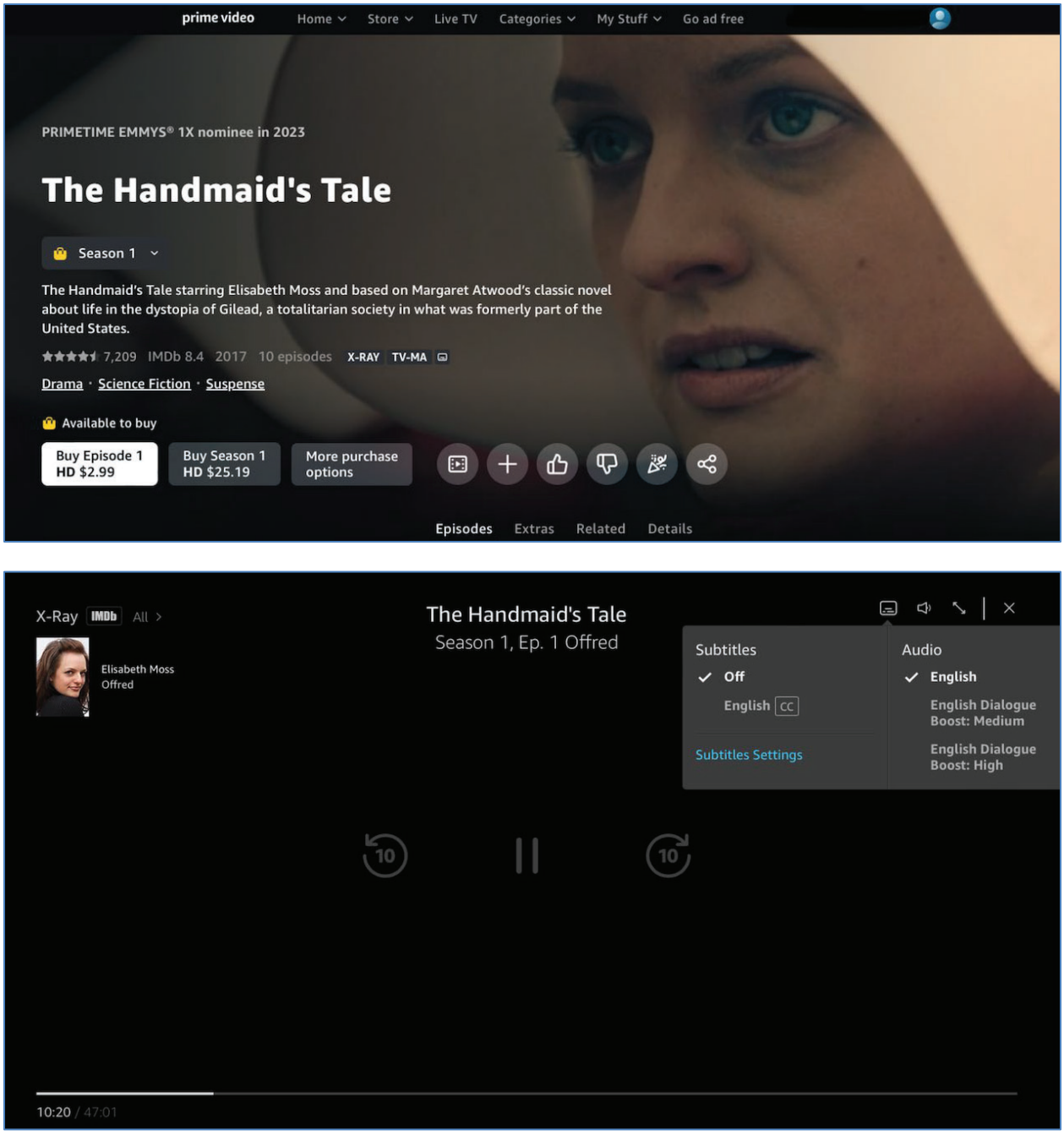
In this session, hear engineers from Amazon Prime Video and Amazon CloudFront discuss how they have architected and optimized their video delivery for scaled global audiences. Topics include optimizing the application and video pipeline for use with content delivery networks (CDN), optimizations in the CDN for efficient and performant video delivery, measuring quality, and effectively managing multi-CDN performance and policy. Learn how CloudFront delivers the performance that Prime Video demands, and hear best practices and lessons learned through scaling this fast-growing service.







See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018#19.



See e.g., exemplary screenshot captures from Amazon Prime Video running on a web browser on a computer.

4.3 DASH data model overview

DASH is intended to support a media-streaming model for delivery of continuous media content in which control lies primarily with the client. Clients may request data using the HTTP protocol from standard web servers that have no DASH-specific capabilities. Consequently, this document focuses not on client or server procedures but on the data formats used to provide a DASH Media Presentation.

The collection of encoded and deliverable versions of continuous media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent or certain periods of a Media Presentation may belong to the same Asset, for example a Media Presentation is a collection of main program composed of multiple periods, each assigned to the same Asset, and interleaved with inserted advertisement periods. Each media content period is composed of one or multiple **media content components,** for example audio components in various languages, different video components providing different views of the same program, subtitles in different languages, etc. Each media content component has an assigned **media content component type**, for example audio or video.

See ISO/IEC 230009-1, Third ed. 2019-08 at 10.

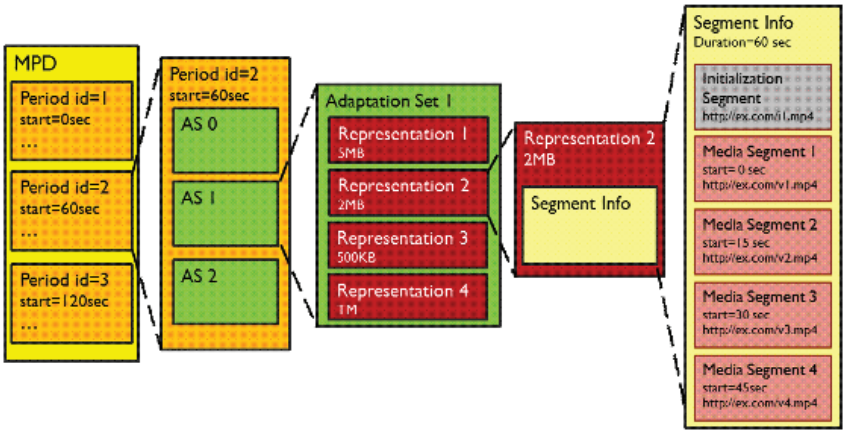


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 4.

NOTE This is not strictly true, since the MPD can also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client can in principle construct a single request for multiple Segments, but this is not the typical case.

DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components. This timeline is referred as Media Presentation timeline. The Media Segments themselves contain accurate Media Presentation timing information enabling synchronization of components and seamless switching.

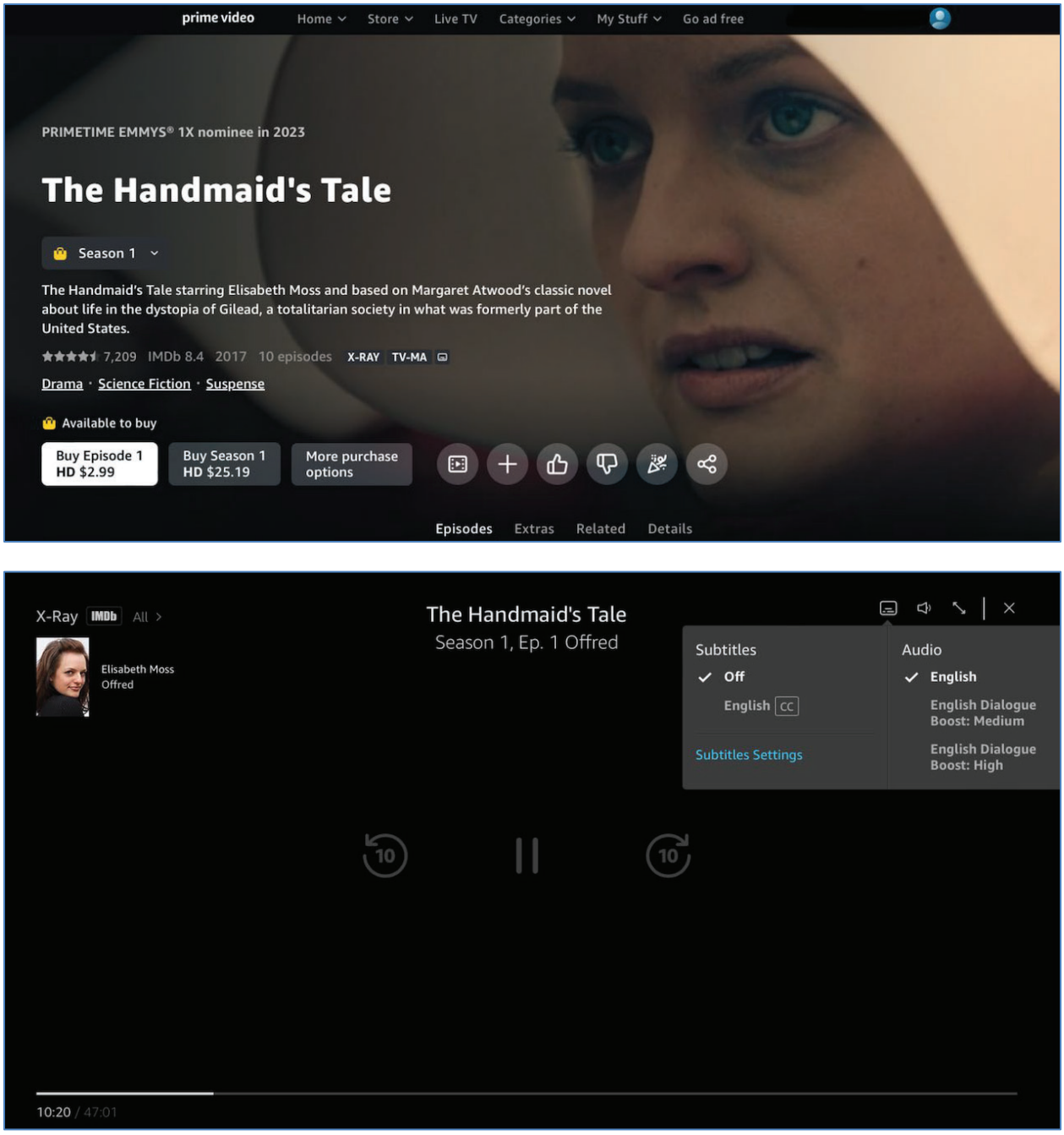
See ISO/IEC 230009-1, Third ed. 2019-08 at 12.

14.5.3. Descriptive Audio

In an effort to further support blind and visually impaired customers, Prime Video fully supports descriptive audio or "Narration". Descriptive audio consists of a narrator talking through the presentation, describing what is happening on the screen or stage during the natural pauses in the audio, and sometimes during dialogue if deemed necessary. In order to deliver descriptive audio, partners will need to follow the below instructions, depending on delivery method.

See https://m.media-amazon.com/images/G/01/CooperWebsite/dvp/downloads/Prime_Video-GlobalContentGuide-v5.2.2.pdf at 56.

71. On information and belief, one or more components of Prime Video provide a method for rendering digital content comprising determining a first digital data file from the plurality of digital data files in a first media stream from the plurality of media streams (e.g. when the user taps the play button, the player determines the correct encoded alternative based on the current network conditions (determining) to download the segment (digital data file) of the requested video content (e.g., first video stream)), the first digital data file having digital content to be rendered with the media player, and determined using the time information in the descriptor file and a predetermined time offset, the predetermined time offset external to the descriptor file and determined relative to the timeline of the audio recording (e.g. when the user taps the play button, the video continues to play from where the user left off).



See e.g., screenshot captures from Amazon Prime Video running on a web browser on a computer.

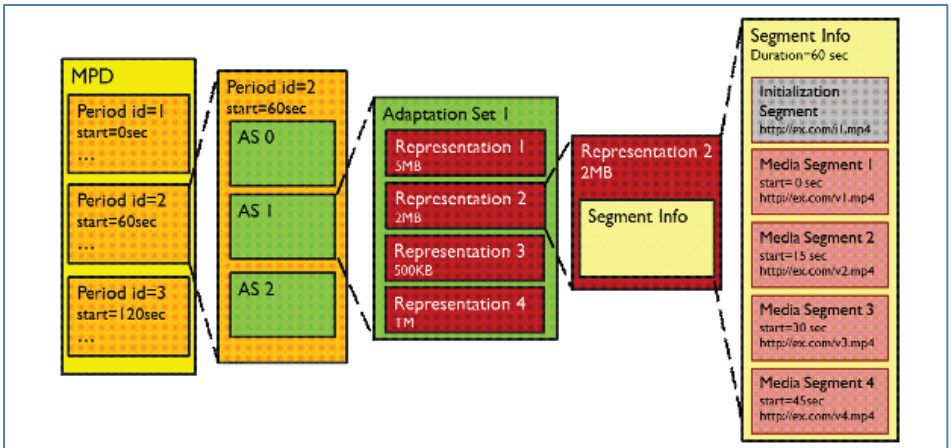


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 4.

> In order to play the content, the DASH client first obtains the MPD. The MPD can be delivered using HTTP, email, thumb drive, broadcast or other transports. By parsing the MPD, the DASH client learns about the timing of the program, the availability of media content, the media types, resolutions, minimum and maximum bandwidths and the existence of various encoded alternatives of multimedia components, the accessibility features and the required digital right management (DRM), the location of each media component on the network and other characteristic of the content. Using this information, the DASH client selects the appropriate encoded alternative and starts streaming of the content by fetching the segments using HTTP GET requests.

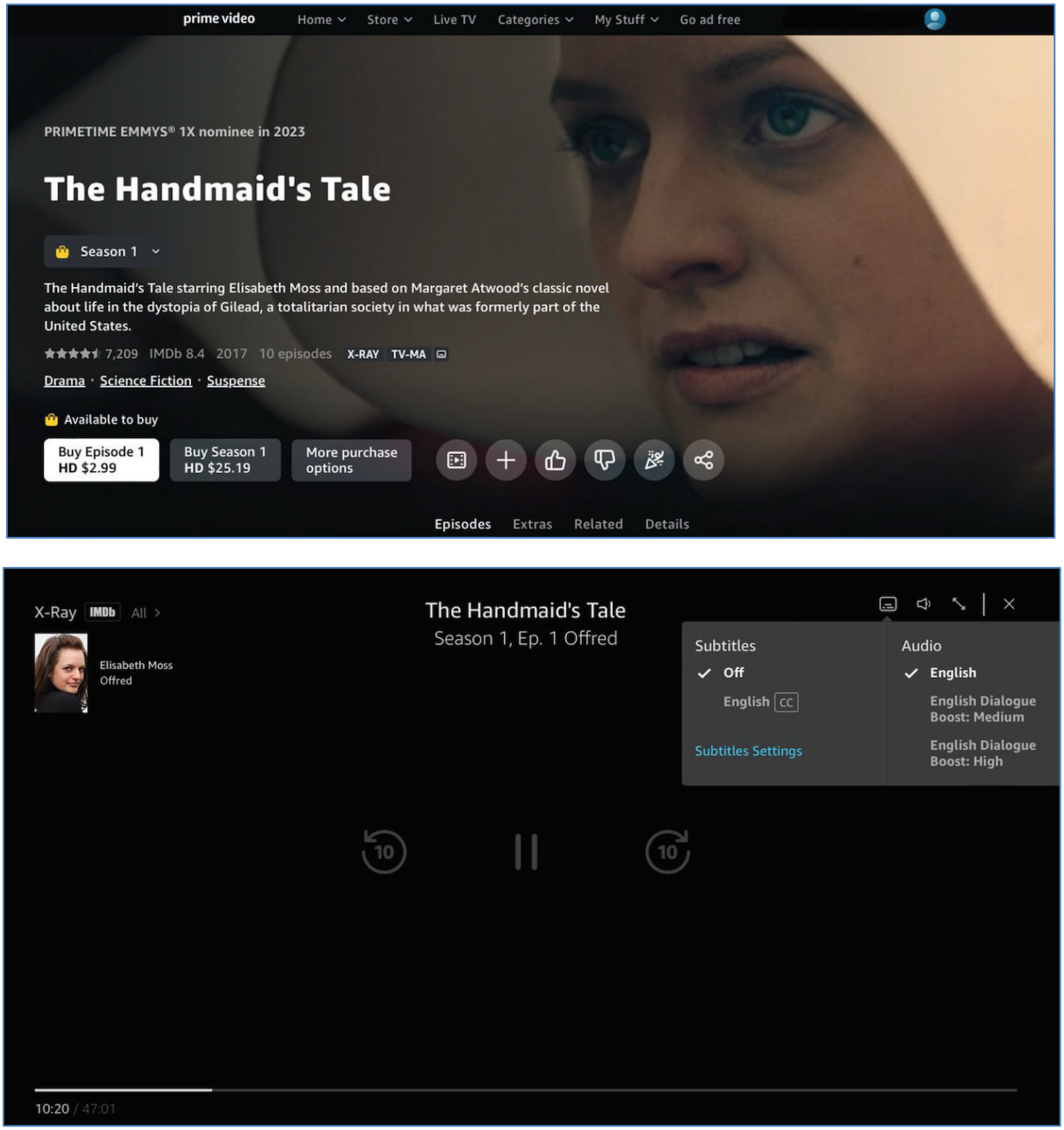
See id. at 3.

> NOTE This is not strictly true, since the MPD can also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client can in principle construct a single request for multiple Segments, but this is not the typical case.
>
> DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components. This timeline is referred as Media Presentation timeline. The Media Segments themselves contain accurate Media Presentation timing information enabling synchronization of components and seamless switching.

See ISO/IEC 230009-1, Third ed. 2019-08 at 12.

72. On information and belief, one or more components of Prime Video provide a method for rendering digital content comprising downloading the first digital data file or a digital data segment contained in the first digital data file and determined using the time information in the descriptor file from the at least one server so that the digital content is resident with the

media player (e.g. when the user taps on the play button, the player downloads and buffers the stream from where the user left off).

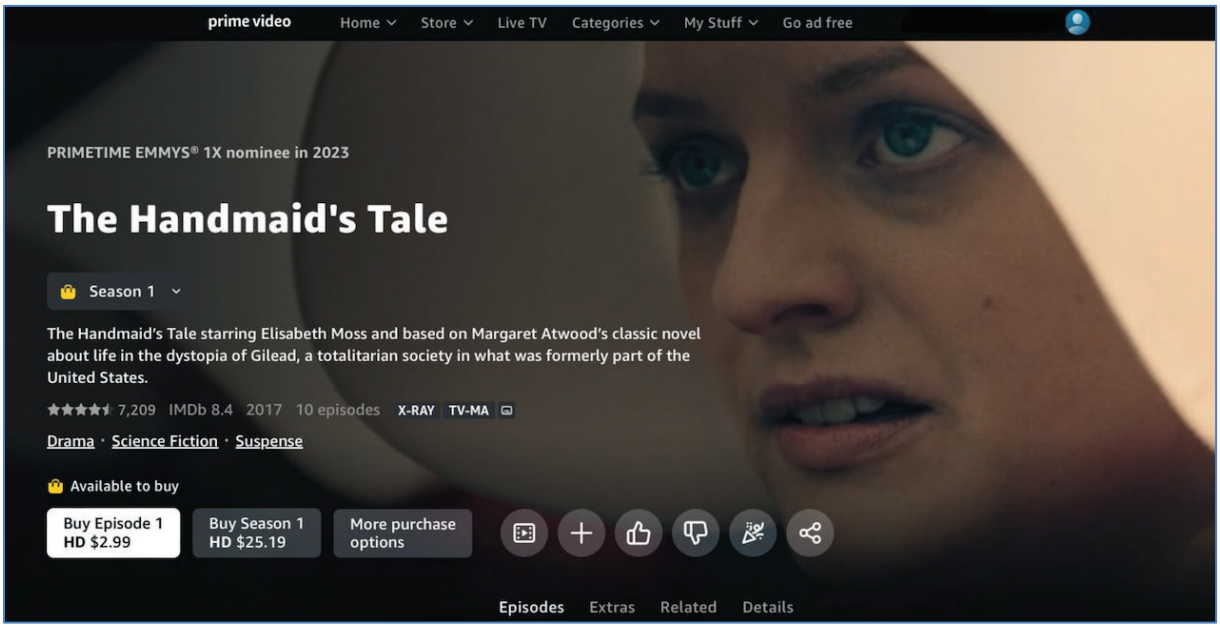


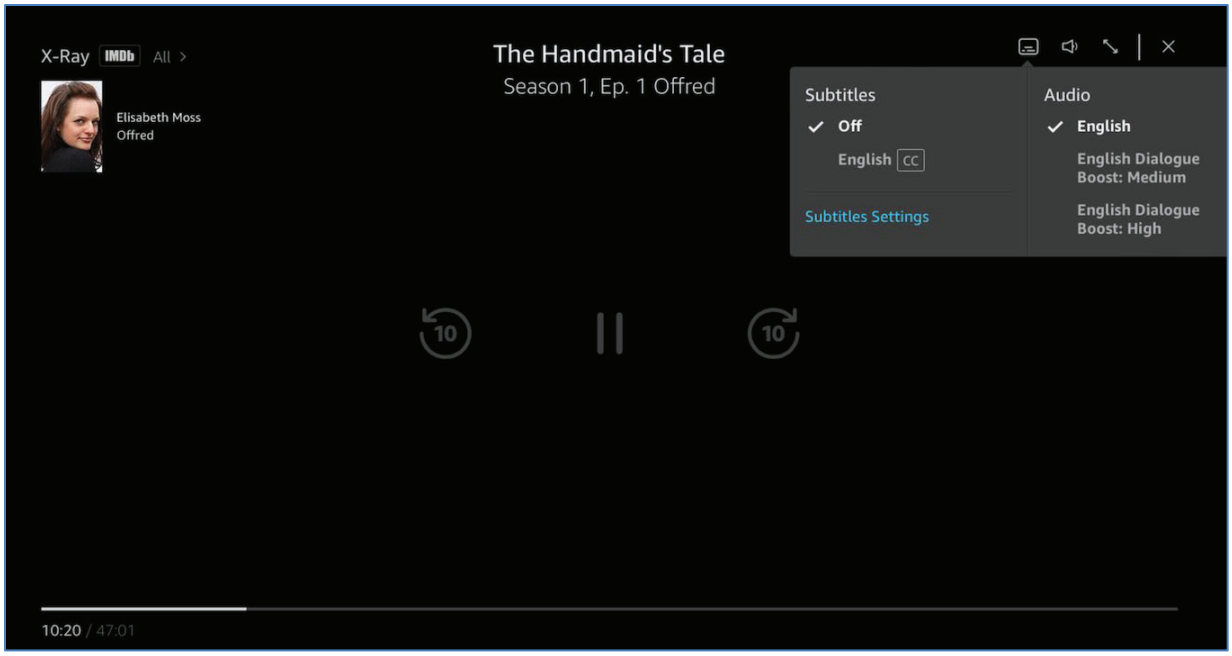
See e.g., screenshot captures from Amazon Prime Video running on a web browser on a computer.

73. On information and belief, one or more components of Prime Video provide a method for rendering digital content comprising rendering the digital content using the media player at an arbitrary point determined using the predetermined time offset.

> **Watch anywhere, anytime**
>
> Watch movies and TV shows on the web at Amazon.com/primevideo or with the Prime Video app on your iOS and Android phone, tablet, or select Smart TVs. To learn more, go to How to Watch a Prime Video Title and Devices Compatible with Prime Video.

See https://www.amazon.com/gp/video/splash/t/getTheApp/ref=atv_dl_rdr#:~:text=After%20you%20download%20the%20app,from%20the%20video%20detail%20page.





See e.g., screenshot captures from Amazon Prime Video running on a web browser on a computer.

74. On information and belief, Defendants directly infringe at least claim 1 of the '720 patent in violation of 35 U.S.C. § 271(a) by making, using, selling, selling access to, importing, offering for sale, and/or offering to sell access to Prime Video.

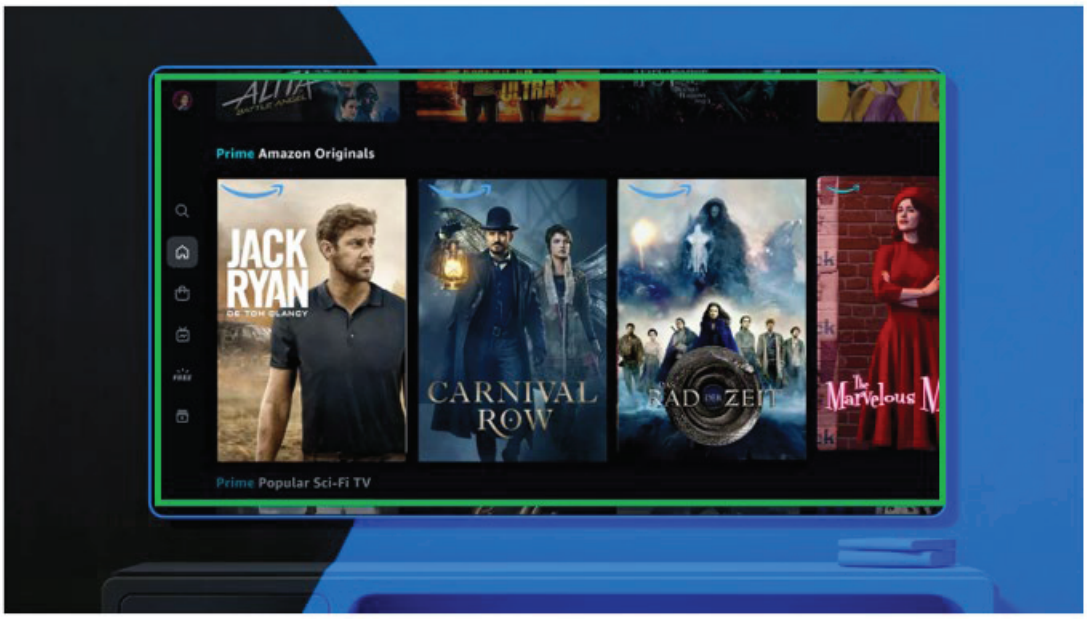
75. Defendants' infringement has damaged Audio Pod and caused/continues to cause it to suffer irreparable harm and damages.

COUNT II - Infringement of the '922 patent by Prime Video

76. Audio Pod repeats, realleges, and incorporates by reference, as if fully set forth here, the allegations of the preceding paragraphs above.

77. On information and belief, Defendants (or those acting on their behalf) make, use, sell, sell access to, import, offer to sell and/or offer to sell access to Prime Video in the United States that infringes (literally and/or under the doctrine of equivalents) at least claim 1 of the '922 patent.

78. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices (*e.g.*, a desktop, laptop, smartphone, smart-TV (e.g., Amazon Fire TV), tablets).



See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

How can I access Prime Video?

Prime Video is available on hundreds of compatible devices. Stream from the web or using the Prime Video
on your smartphone, tablet, set-top box, game console, or select smart TVs.

See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

Purchased Videos

- **Ways to Watch:** When you purchase a video for on-demand viewing, we'll make it available to you to stream and, in most cases, download as follows:
  - **Streaming:** You may stream purchased videos online through your web browser and compatible Internet-connected TVs, Blu-ray players, set-top-boxes, Fire tablets, and other compatible devices. For a list of devices compatible with our service, visit the Compatible Devices page on the website you are using to access Amazon Prime Video at the following links, PrimeVideo.com, Amazon.com, Amazon.co.uk, Amazon.de or Amazon.co.jp (device compatibility may vary by location). You may stream up to three videos at the same time using the same Amazon account. You may stream the same video to no more than two devices at a time.

See https://www.primevideo.com/help?nodeId=202095500.

79. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising downloading first digital content corresponding to a media work (*e.g.*, Prime Video uses MPEG-DASH for adaptive streaming, which can include, e.g., video streams, audio streams, and subtitle streams corresponding to a media work (e.g., a movie or visual work))from a network accessible library (e.g., Prime Video CDN) to a first client device via a network, the first digital content including at least a first portion (e.g., "video chunk") of a first media stream.

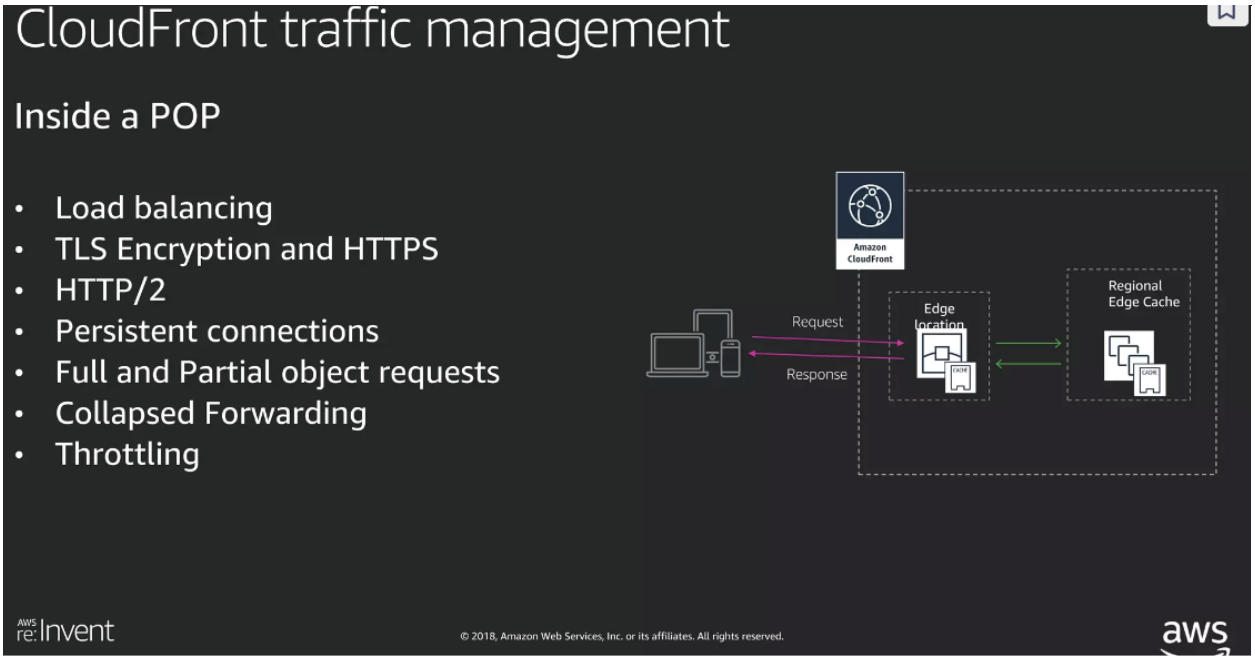
Another case study is Amazon Prime Video who operates a video on demand service (transactional and subscription based) with a very large catalog where 12% of titles account for 90% of playbacks. For calculating performance scores of CDNs, they mostly index on the percentage of sessions without rebuffering, the percentage of sessions without fatal errors, bitrates, response times, and time to first frame. Watch this talk from re:Invent 2018, where Amazon Prime Video explains how they deliver their video using multiple CDNs, including CloudFront. You can also hear how CloudFront engineers work backwards with Amazon Prime Video to optimize performance and gain more traffic share.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

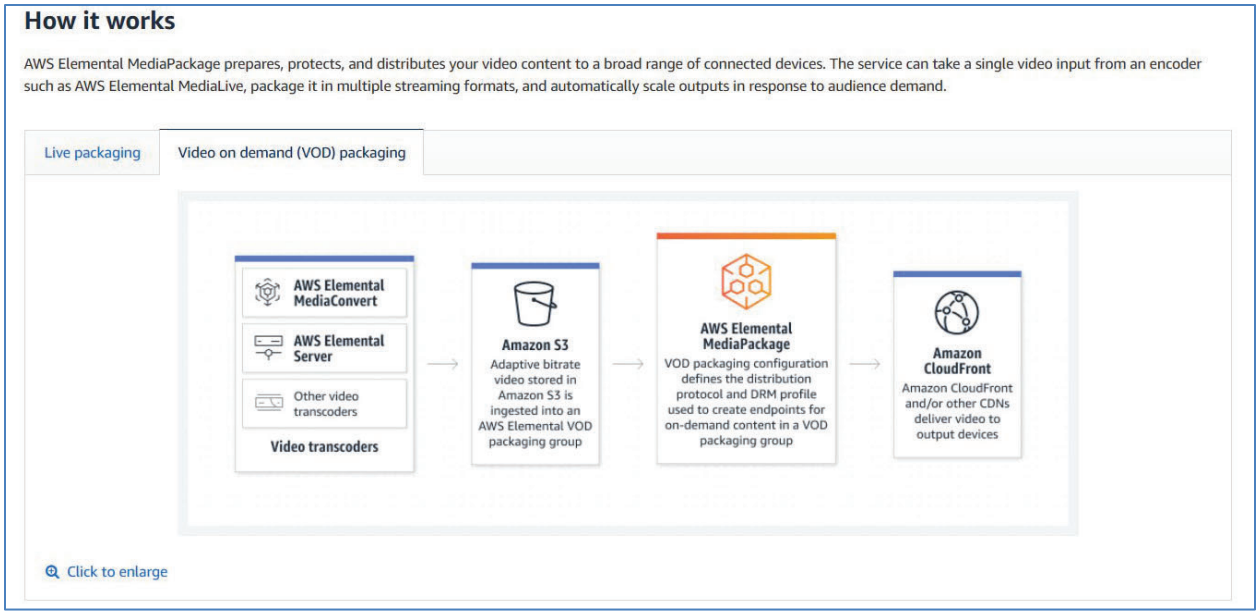
What can I watch on Prime Video?

Once you're signed up for Prime Video, you'll have access to a massive library of content.

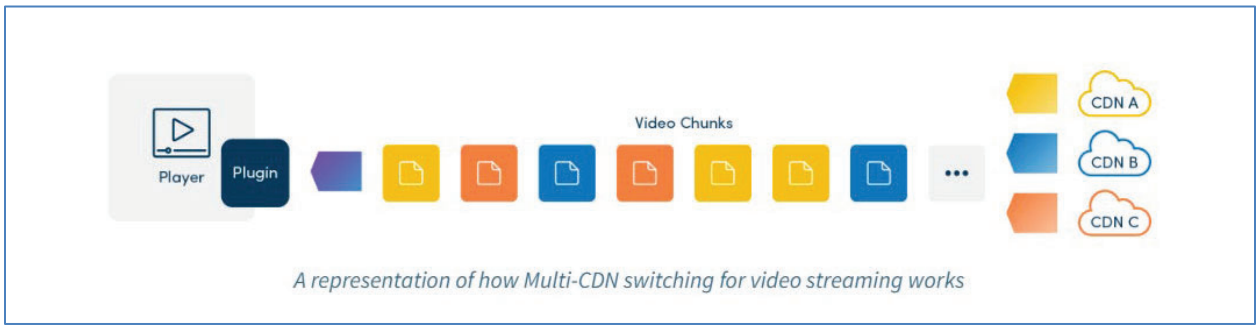
See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.



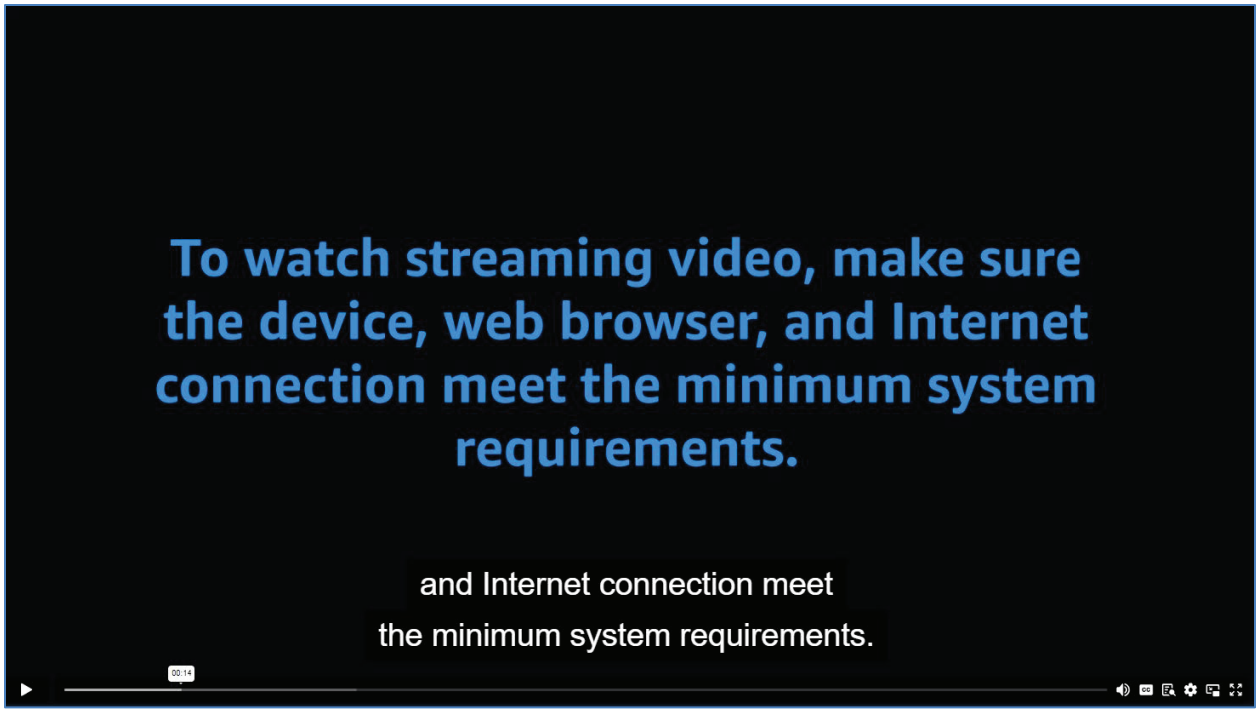
See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.



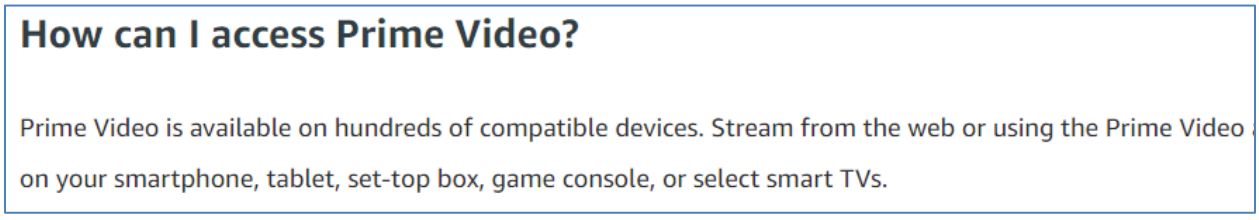
See https://aws.amazon.com/mediapackage/.



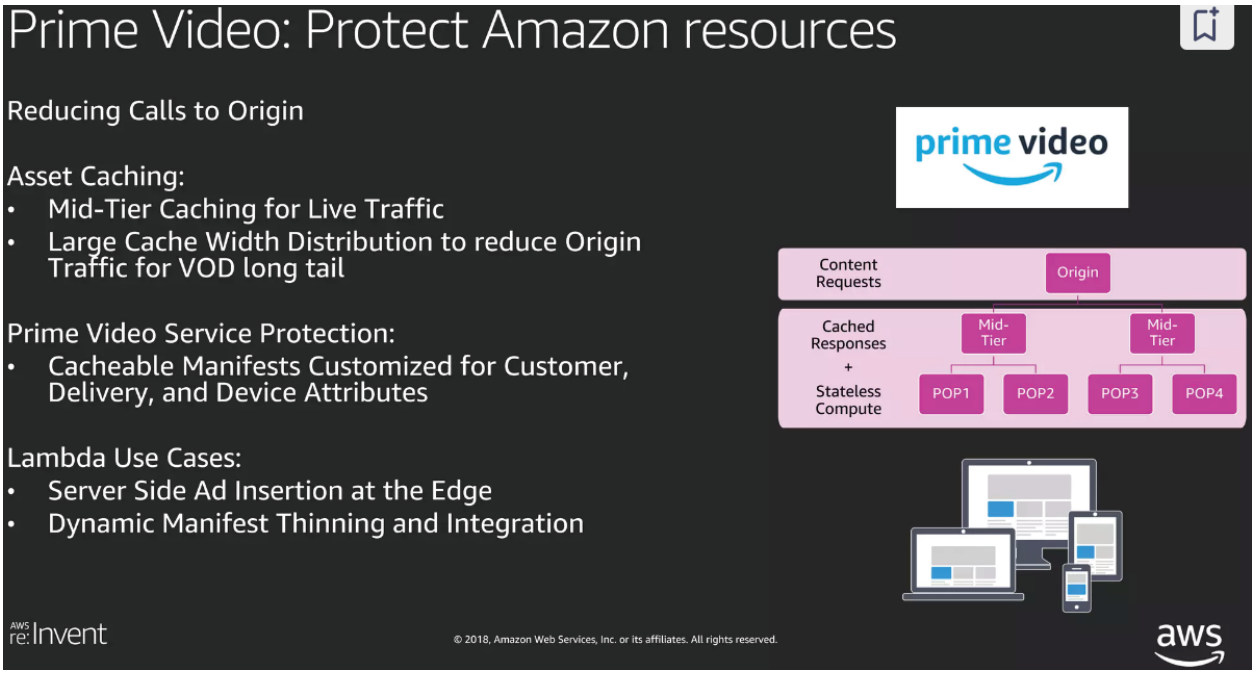
See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.



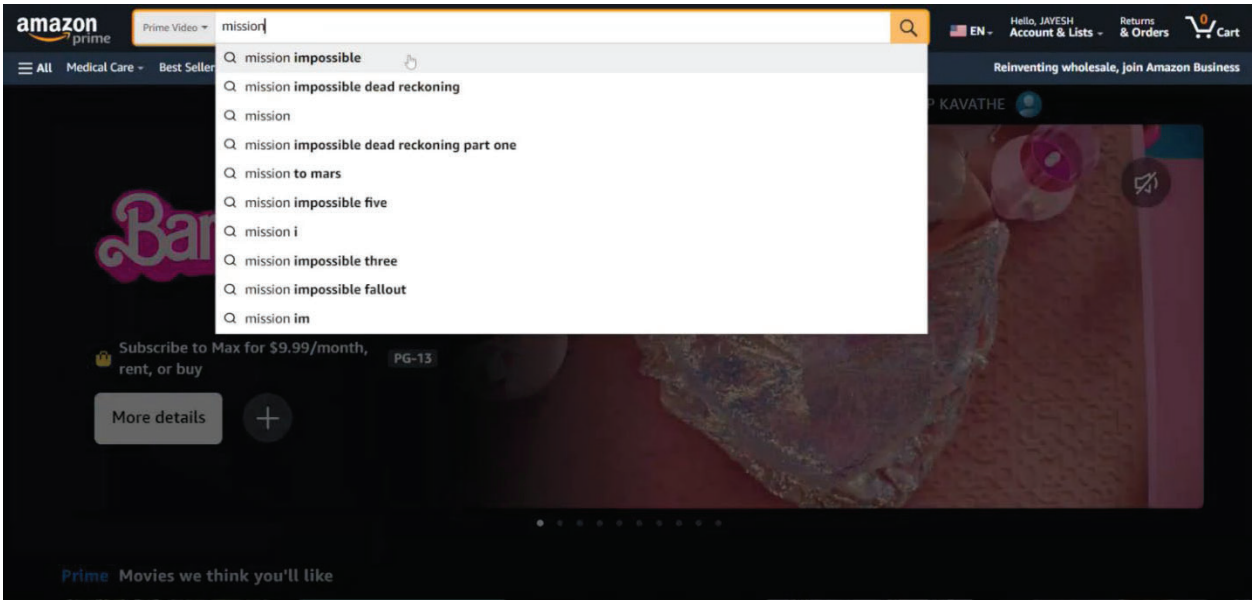
See https://www.amazon.com/gp/help/customer/display.html?ref_=hp_left_v4_sib&nodeId=G93JTA5QNSJT8WWP.



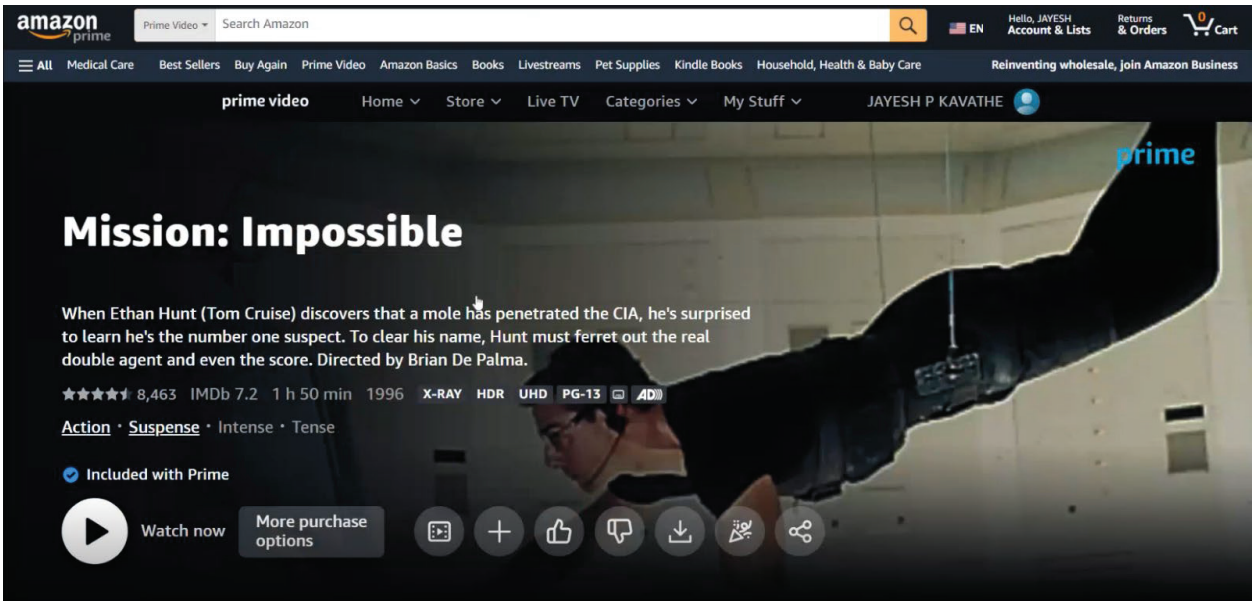
See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.

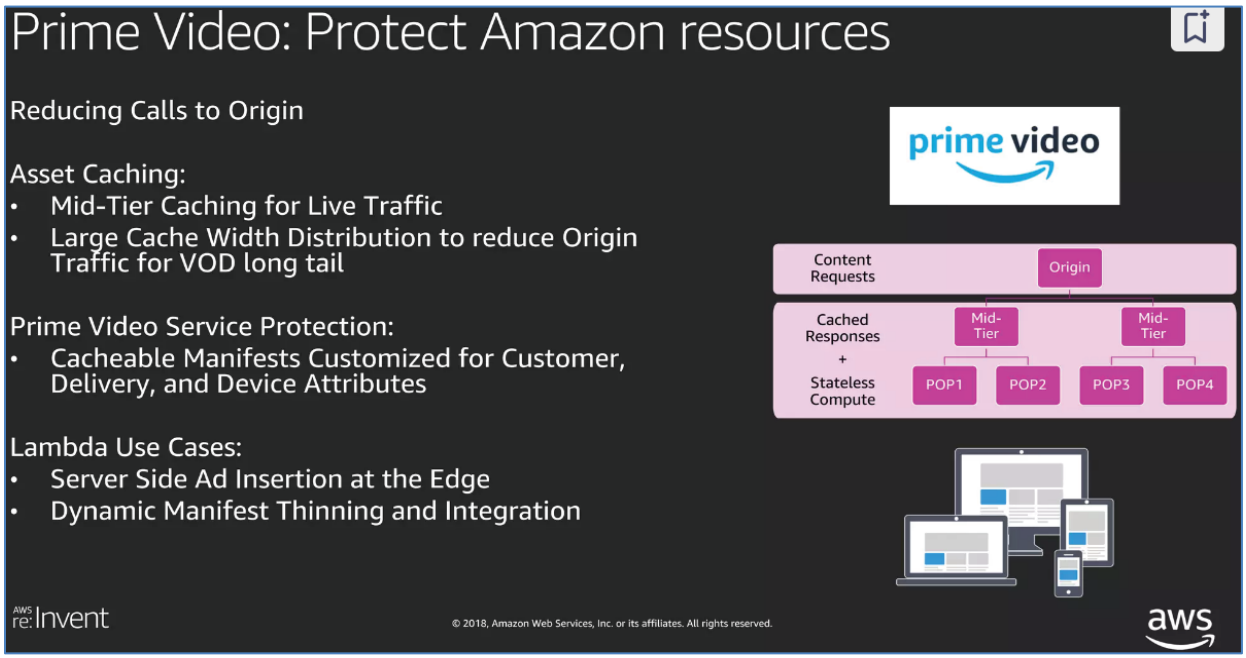


See Screenshot taken during product testing.

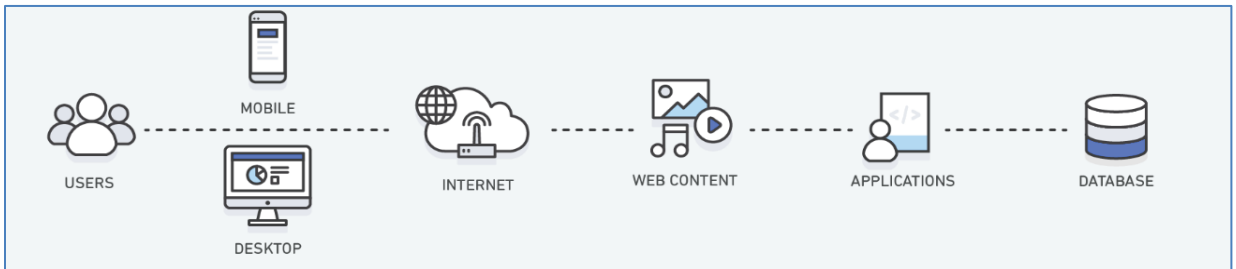


See Screenshot taken during product testing.

80. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising storing (e.g., "retrieval of web content" by the client-side device) the first digital content on the first client device.



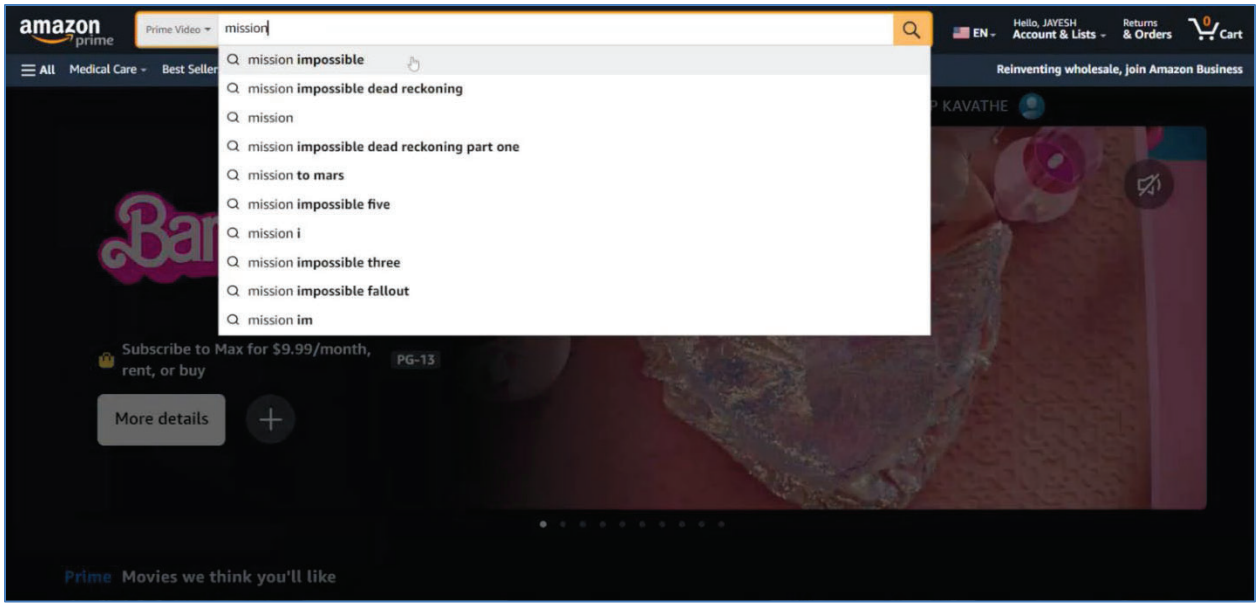
See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.



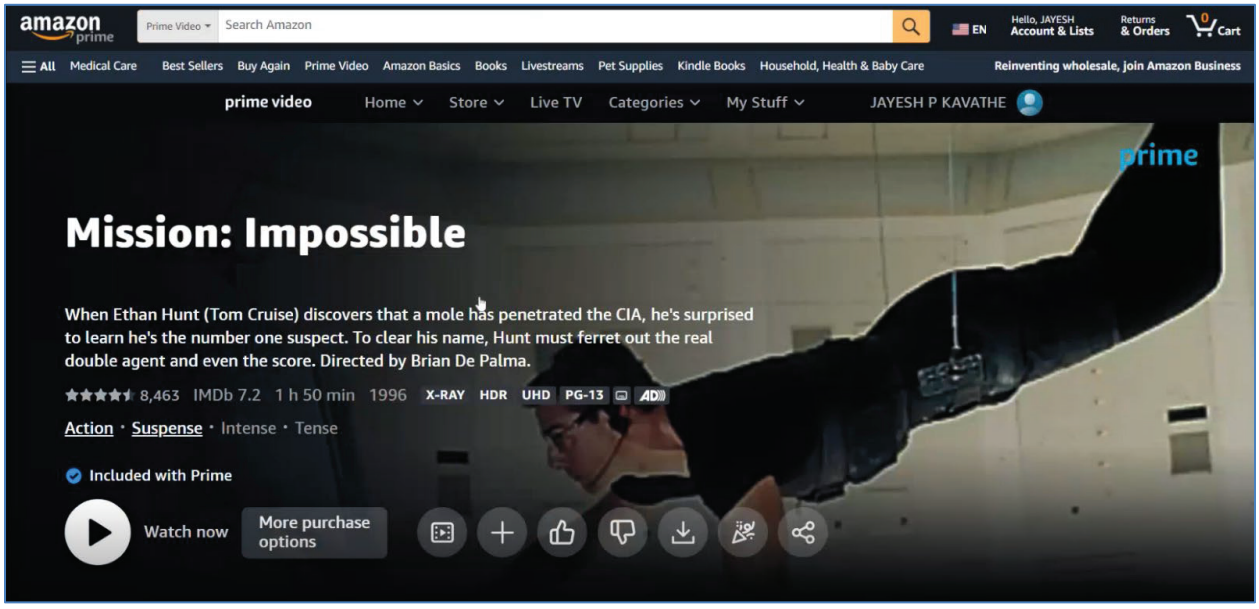
Layer	Client-Side	DNS	Web	App	Database
Use Case	Accelerate retrieval of web content from websites (browser or device)	Domain to IP Resolution	Accelerate retrieval of web content from web/app servers. Manage Web Sessions (server side)	Accelerate application performance and data access	Reduce latency associated with database query requests
Technologies	HTTP Cache Headers, Browsers	DNS Servers	HTTP Cache Headers, CDNs, Reverse Proxies, Web Accelerators, Key/Value Stores	Key/Value data stores, Local caches	Database buffers, Key/Value data stores
Solutions	Browser Specific	Amazon Route 53	Amazon CloudFront, ElastiCache for Redis, ElastiCache for Memcached, Partner Solutions	Application Frameworks, ElastiCache for Redis, ElastiCache for Memcached, Partner Solutions	ElastiCache for Redis, ElastiCache for Memcached

See https://aws.amazon.com/caching/.

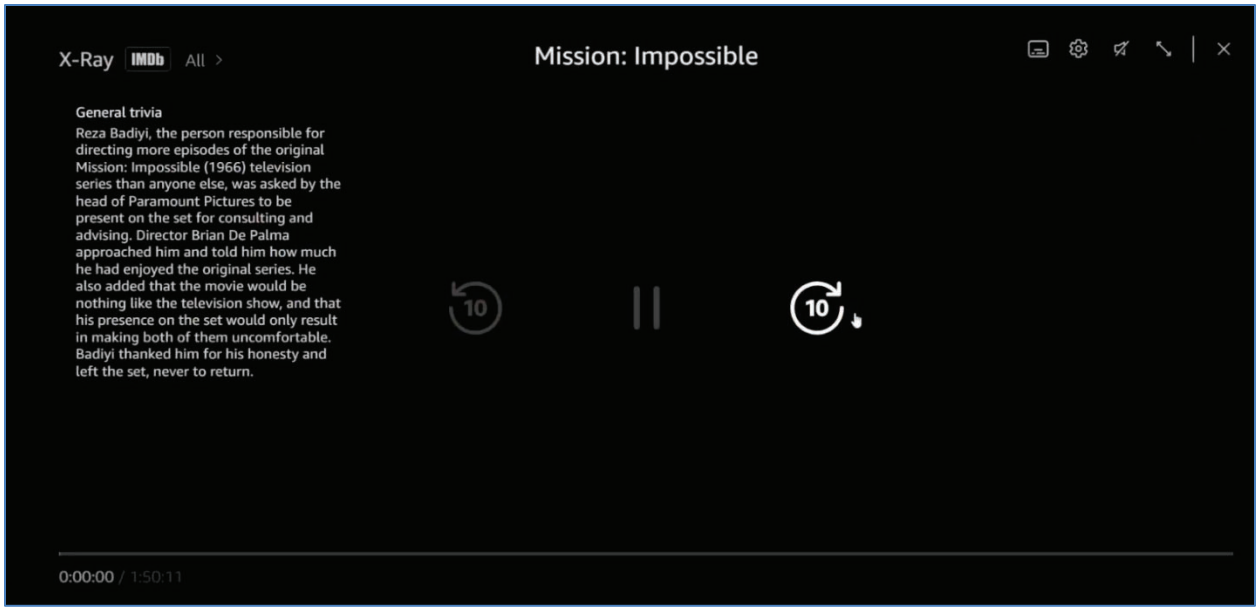
81. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising rendering (e.g., presenting the media content) at least a portion (e.g., "video chunk") of the first digital content on the client device.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



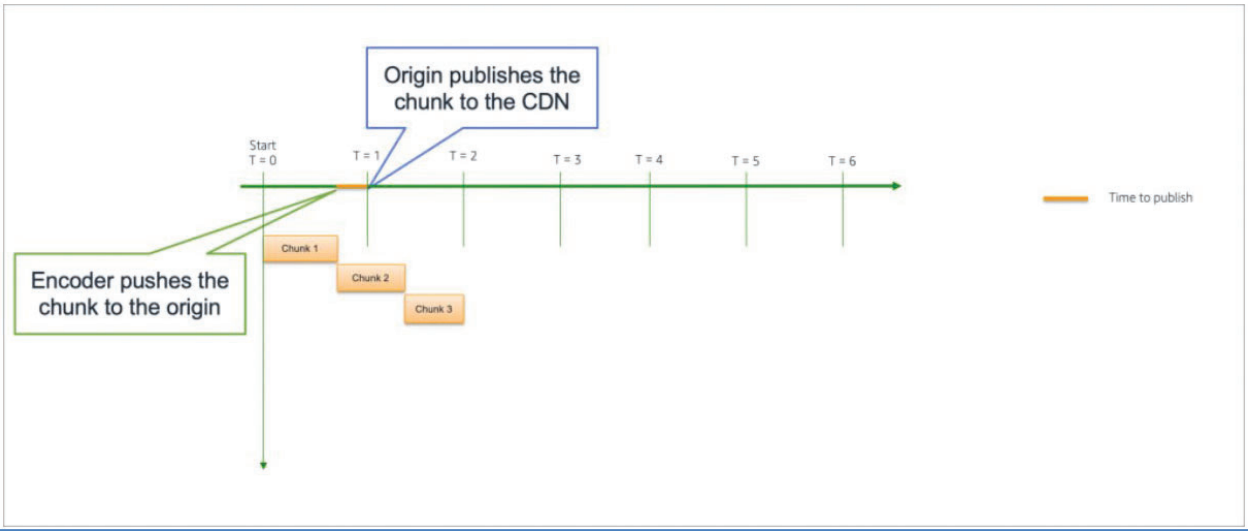
See Screenshot taken during product testing.

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

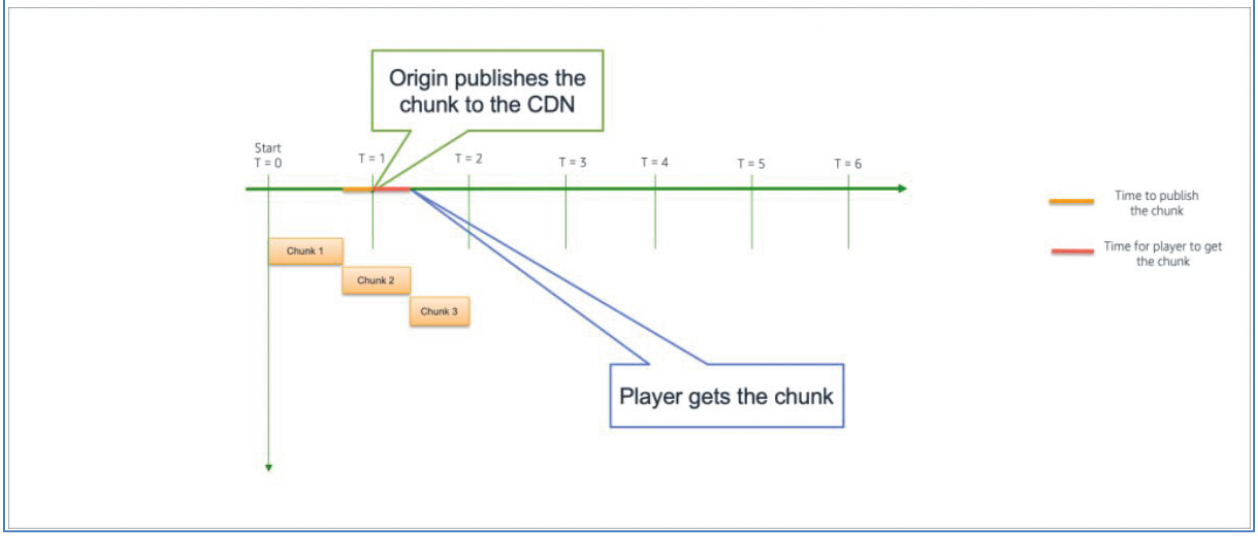
This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

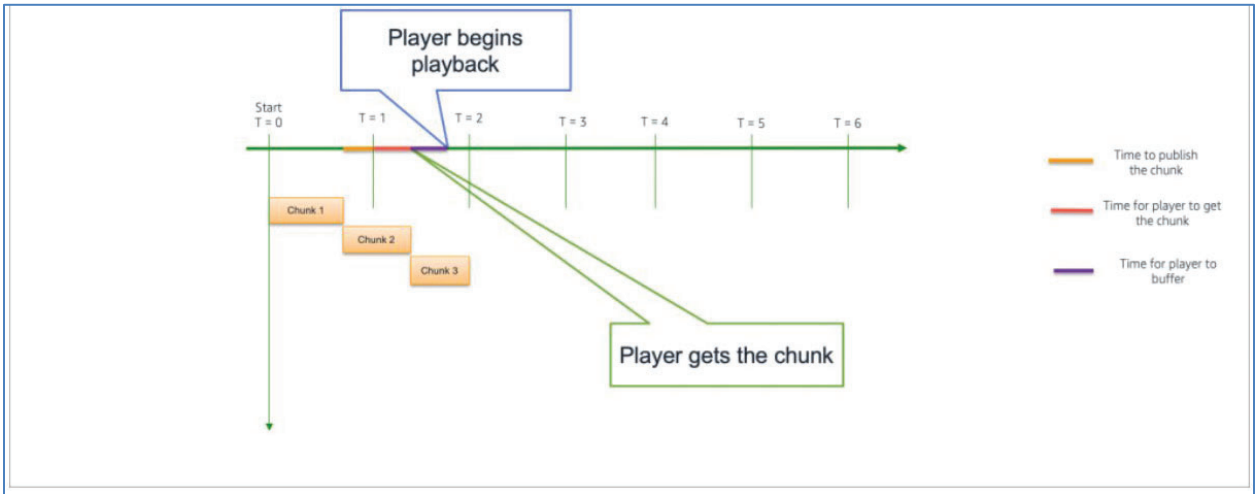
When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.

When using chunked transfer encoding, video delivery an start much sooner.



Once the chunk is pushed to MediaStore, it is available for delivery via a CDN like CloudFront. CloudFront will now be able to pull this chunk, and collapse all corresponding requests for this chunk.





The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

82. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising tracking a current

position in the first media stream as the first digital content is rendered (e.g., determining and displaying the progress bar for the movie/media content).

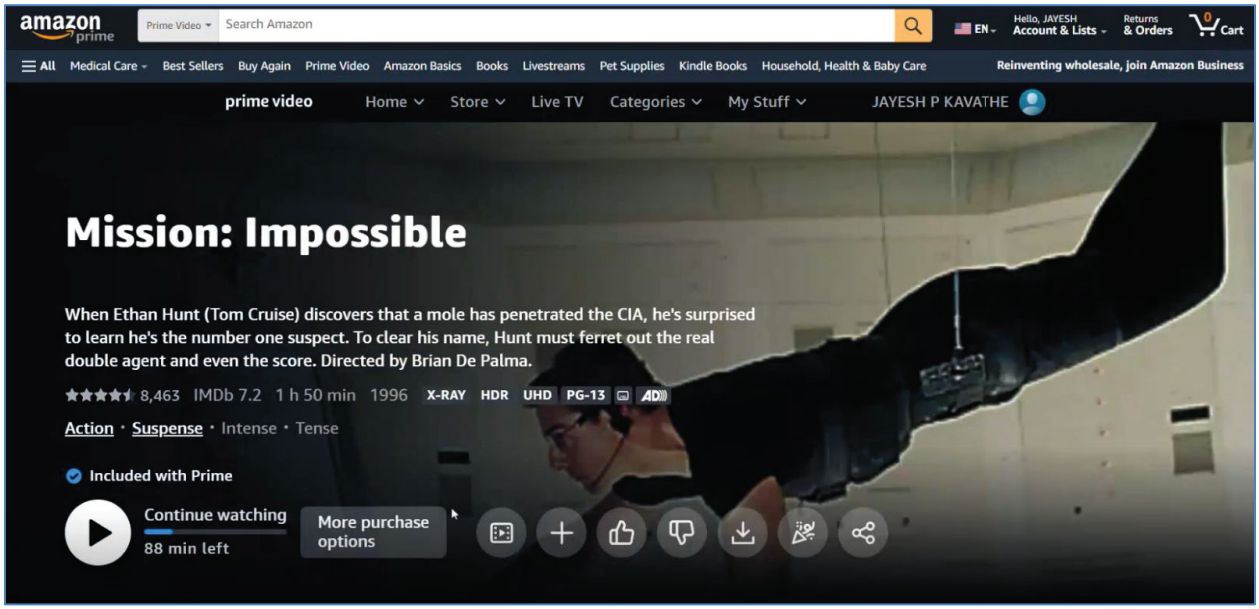


See Screenshot taken during product testing.

83. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising creating a bookmark (e.g., creating a data component that includes timing information to indicate the stop/pause position for the media stream) by setting the current position as a bookmarked position, the bookmark including information for identifying the media work and the bookmarked position.



See Screenshot taken during product testing.



See Screenshot taken during product testing.

84. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising transferring the bookmark to a second client device via the network (e.g., the "Continue Watching" feature).

Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

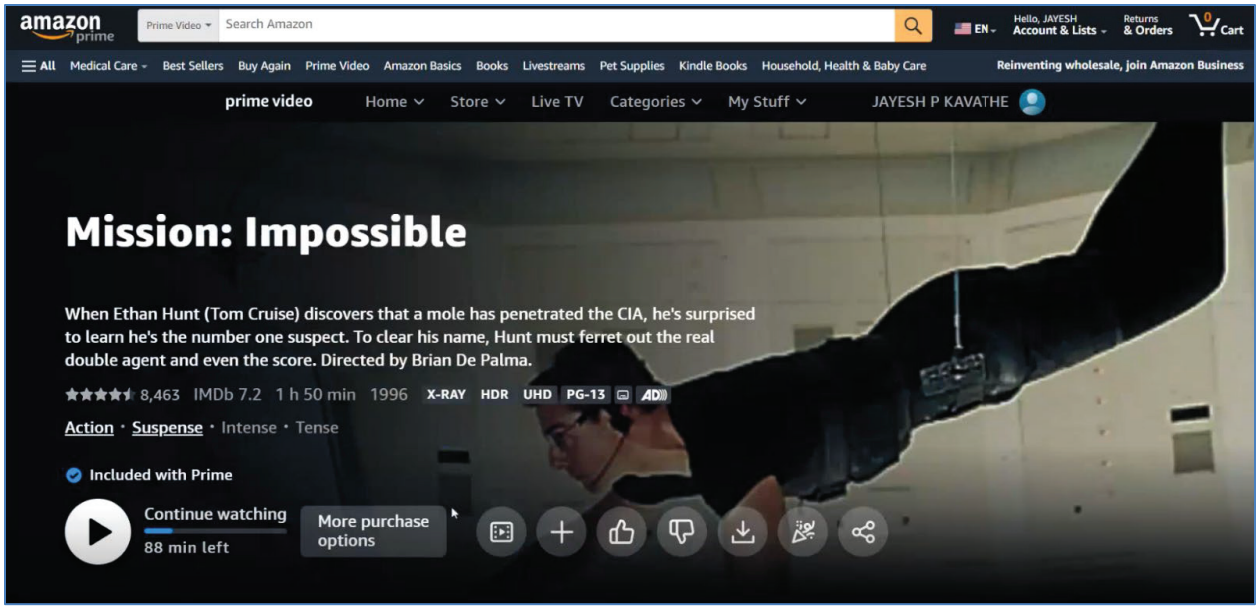
While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

7. Multi-Device Sync: Amazon Prime Video offers multi-device syncing, allowing you to start watching a TV show or movie on one device and continue on another. For example, if you start streaming on your smartphone, you can pause and resume playback seamlessly on your smart TV or tablet using the same Amazon account.

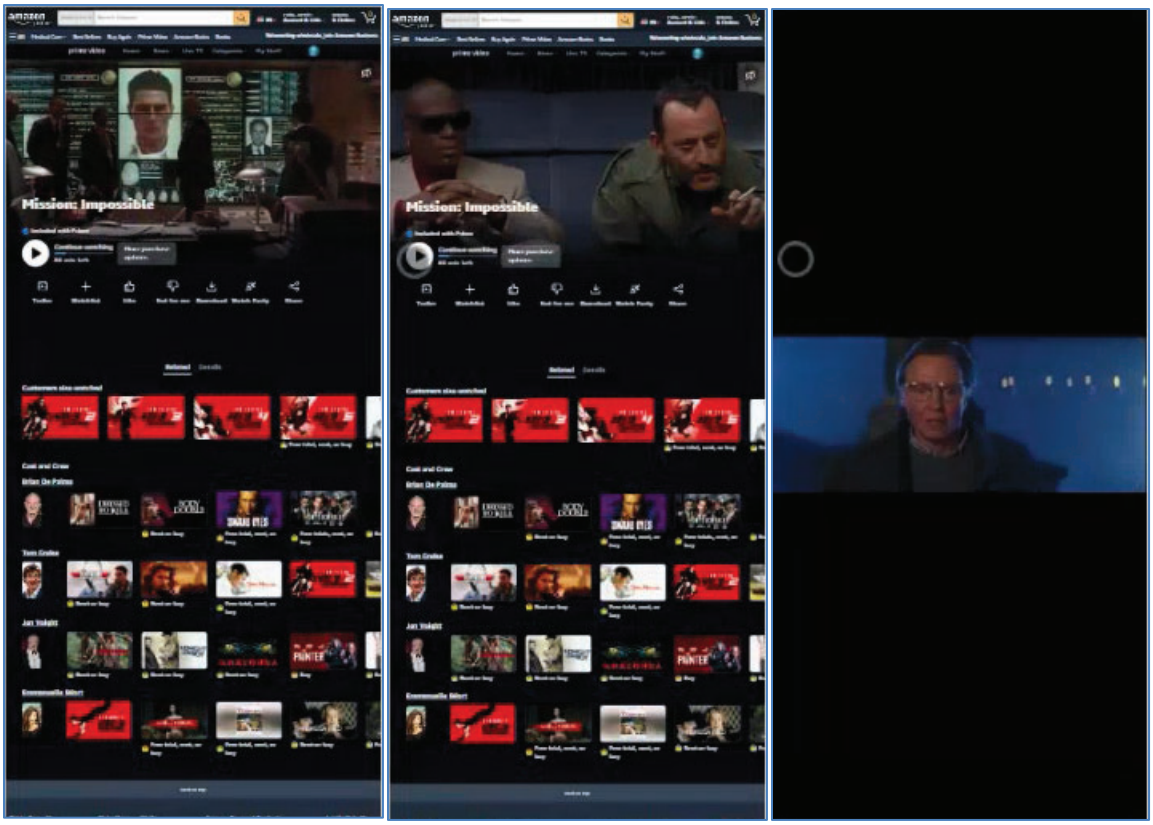
See https://citizenside.com/entertainment/how-to-watch-tv-with-amazon-prime/.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



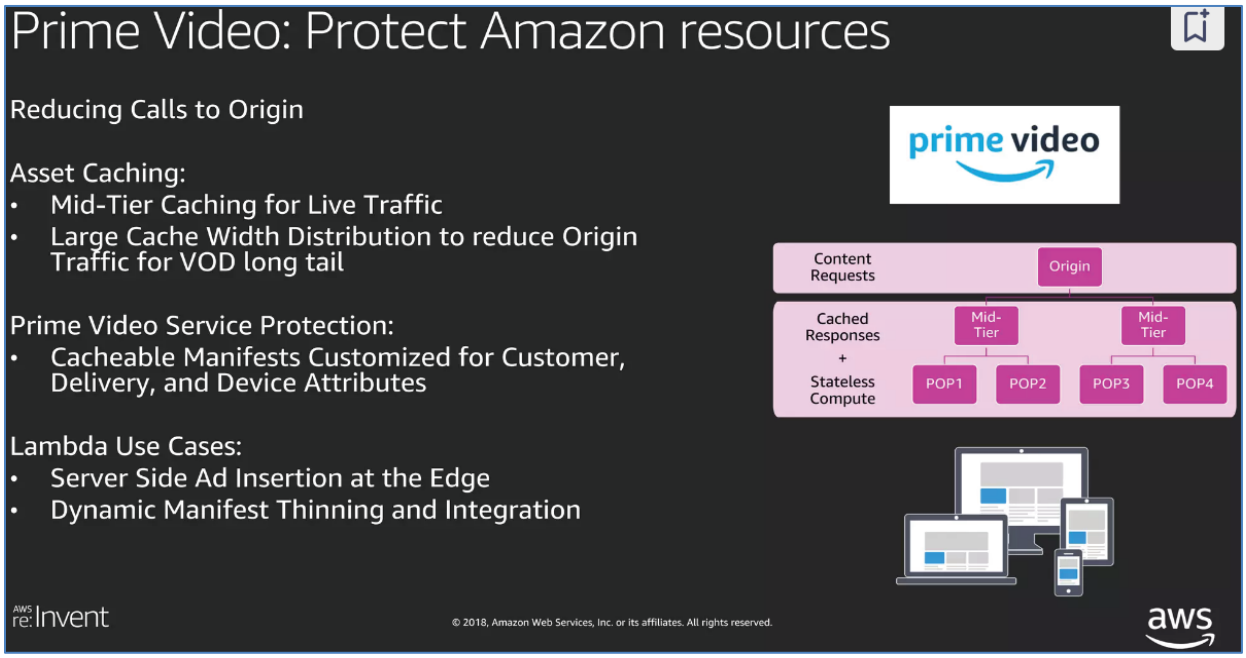
See Screenshot taken during product testing.

85. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising downloading second digital content corresponding to the media work from the network accessible library to the second client device via the network, the second digital content including at least a second portion of the first media stream (e.g., a second "video chunk" of the first media stream) or at least a portion of a second media stream (e.g., a first "video chunk" of the second media stream), the second portion of the first media stream or the portion of the second media stream including the bookmarked position.

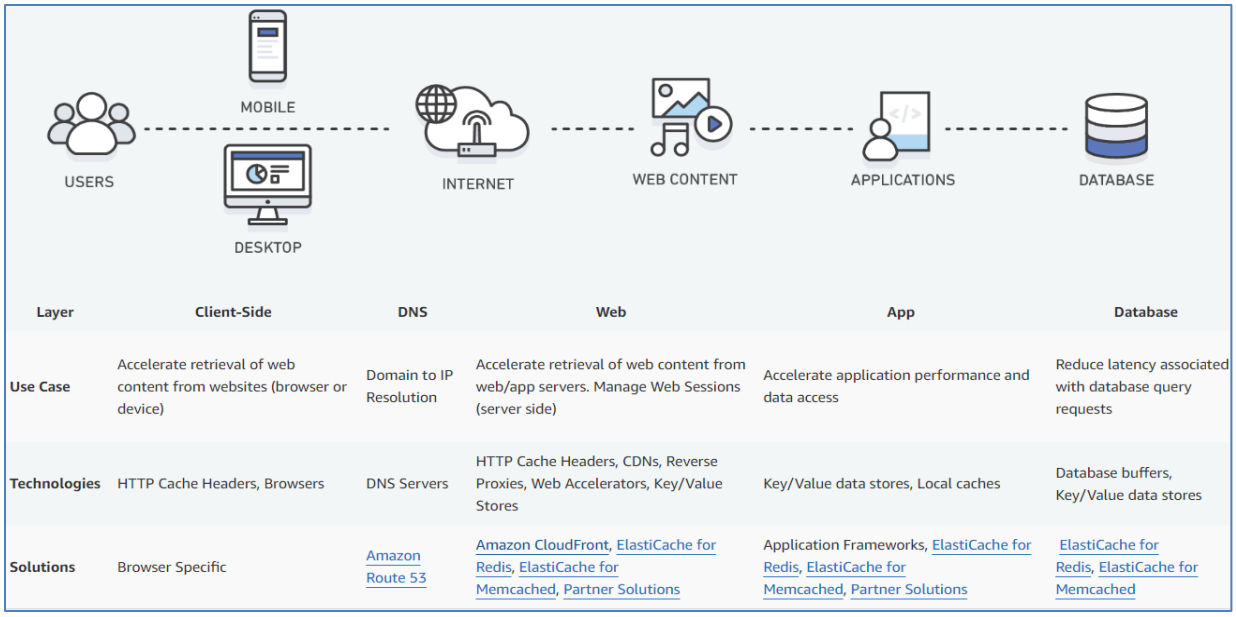
> 7. Multi-Device Sync: Amazon Prime Video offers multi-device syncing, allowing you to start watching a TV show or movie on one device and continue on another. For example, if you start streaming on your smartphone, you can pause and resume playback seamlessly on your smart TV or tablet using the same Amazon account.

See https://citizenside.com/entertainment/how-to-watch-tv-with-amazon-prime/.

86. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising the step of storing (e.g., "retrieval of web content" by a client-side device) the second digital content on the second client device .



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.



Layer	Client-Side	DNS	Web	App	Database
Use Case	Accelerate retrieval of web content from websites (browser or device)	Domain to IP Resolution	Accelerate retrieval of web content from web/app servers. Manage Web Sessions (server side)	Accelerate application performance and data access	Reduce latency associated with database query requests
Technologies	HTTP Cache Headers, Browsers	DNS Servers	HTTP Cache Headers, CDNs, Reverse Proxies, Web Accelerators, Key/Value Stores	Key/Value data stores, Local caches	Database buffers, Key/Value data stores
Solutions	Browser Specific	Amazon Route 53	Amazon CloudFront, ElastiCache for Redis, ElastiCache for Memcached, Partner Solutions	Application Frameworks, ElastiCache for Redis, ElastiCache for Memcached, Partner Solutions	ElastiCache for Redis, ElastiCache for Memcached

See https://aws.amazon.com/caching/.

87. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising rendering (e.g., presenting the media content) at least a portion (e.g., "video chunk") of the second digital content on the second client device in dependence upon the bookmarked position (e.g., through the "Continue Watching" feature).

Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

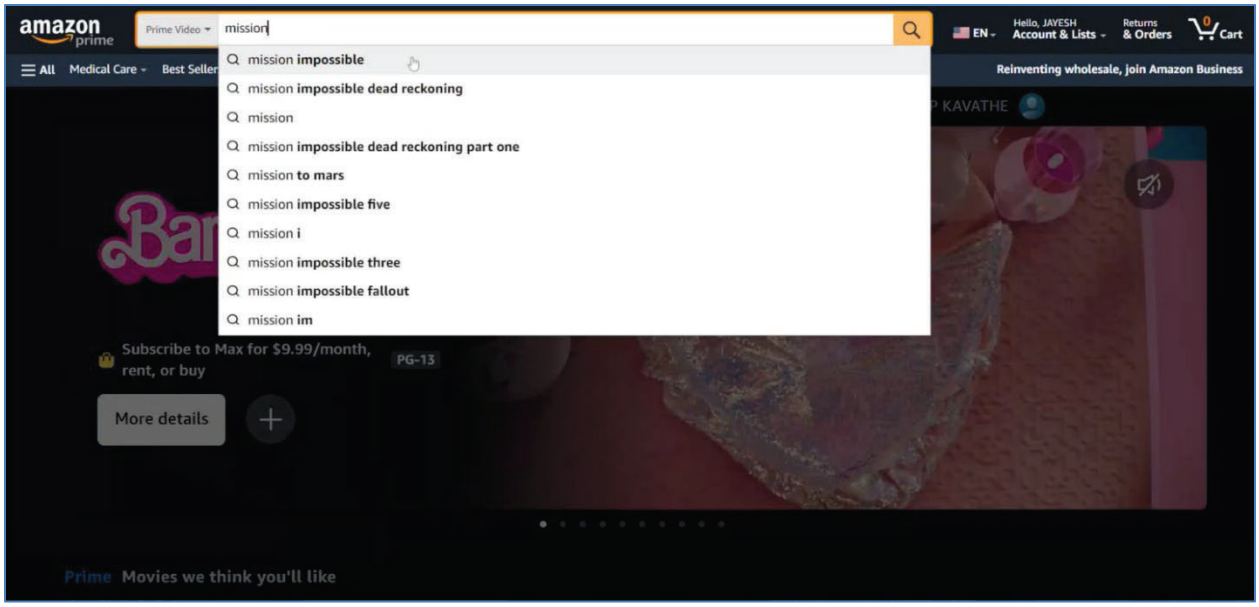
While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

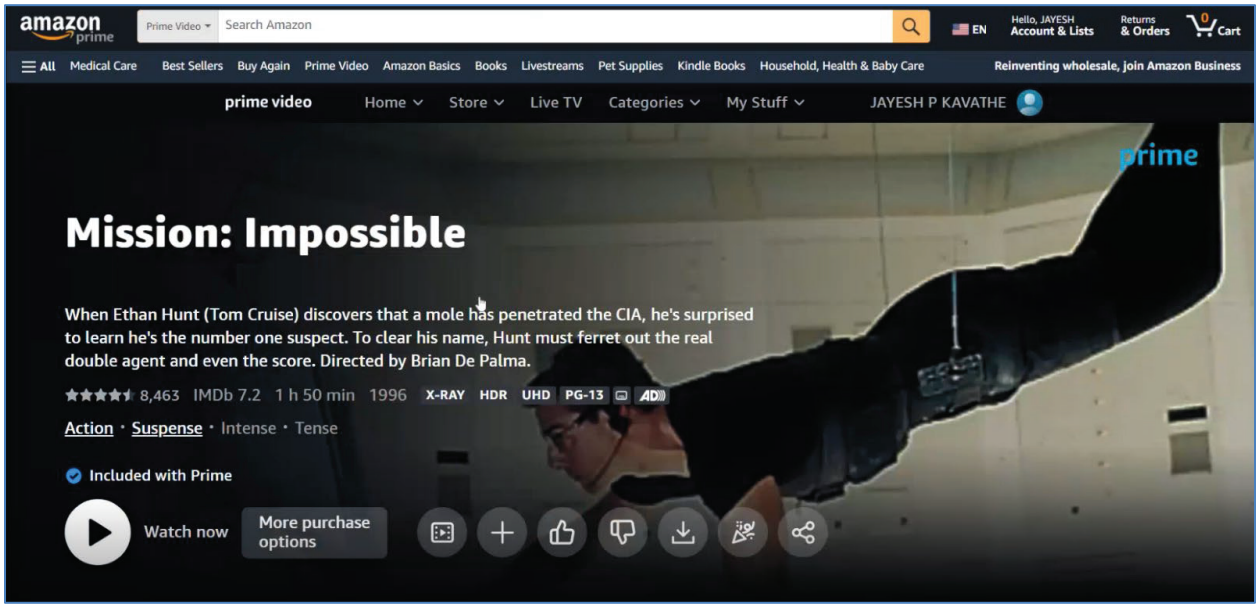
See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

7. Multi-Device Sync: Amazon Prime Video offers multi-device syncing, allowing you to start watching a TV show or movie on one device and continue on another. For example, if you start streaming on your smartphone, you can pause and resume playback seamlessly on your smart TV or tablet using the same Amazon account.

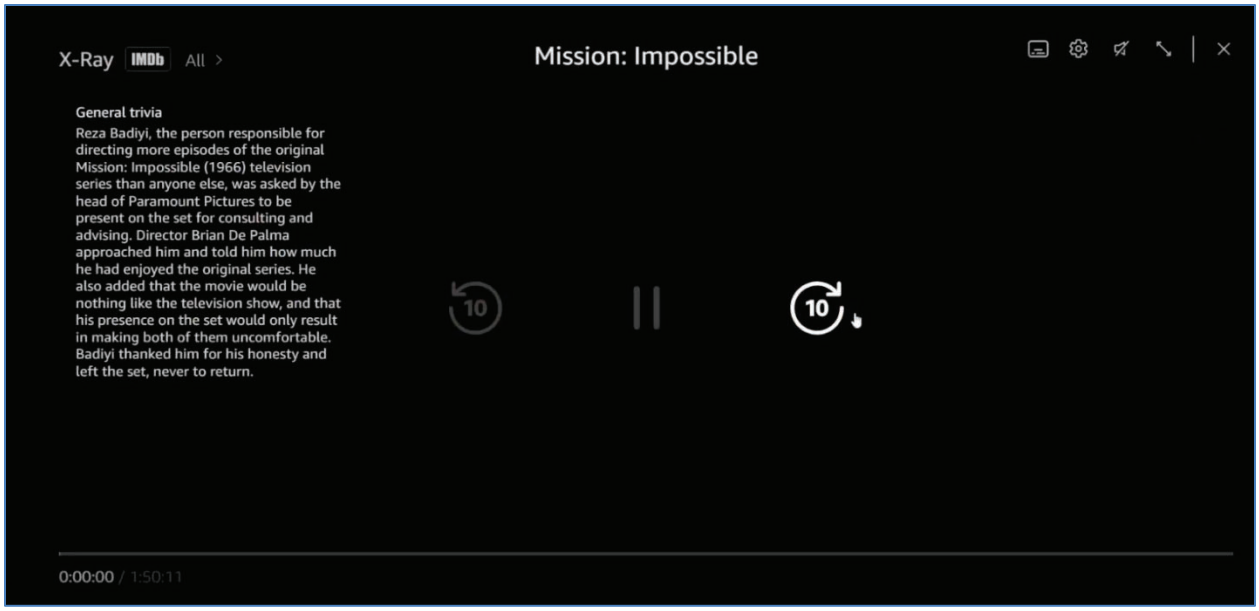
See https://citizenside.com/entertainment/how-to-watch-tv-with-amazon-prime/.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



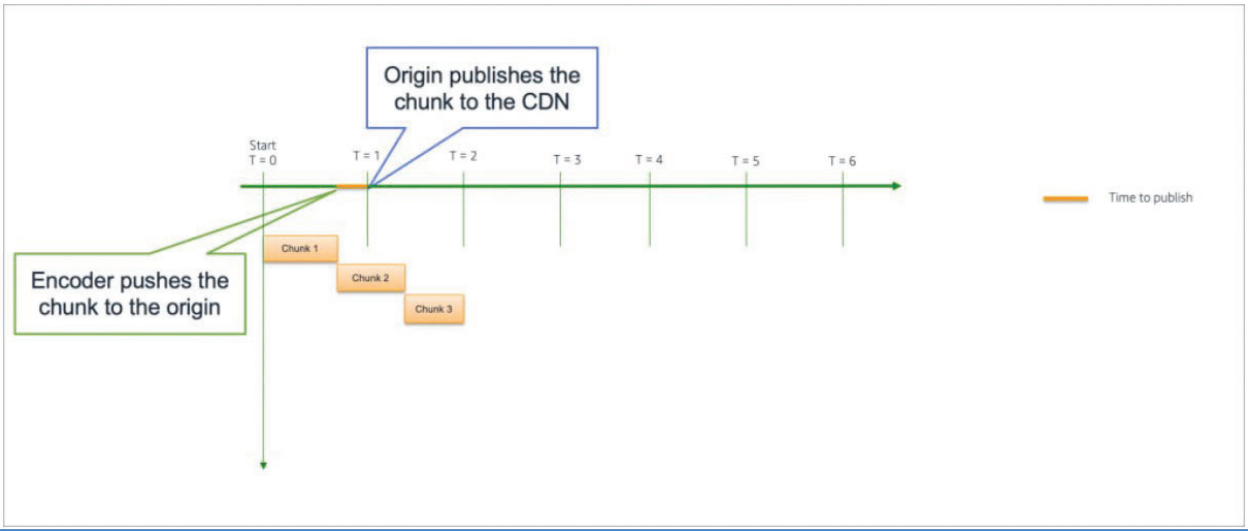
See Screenshot taken during product testing.

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

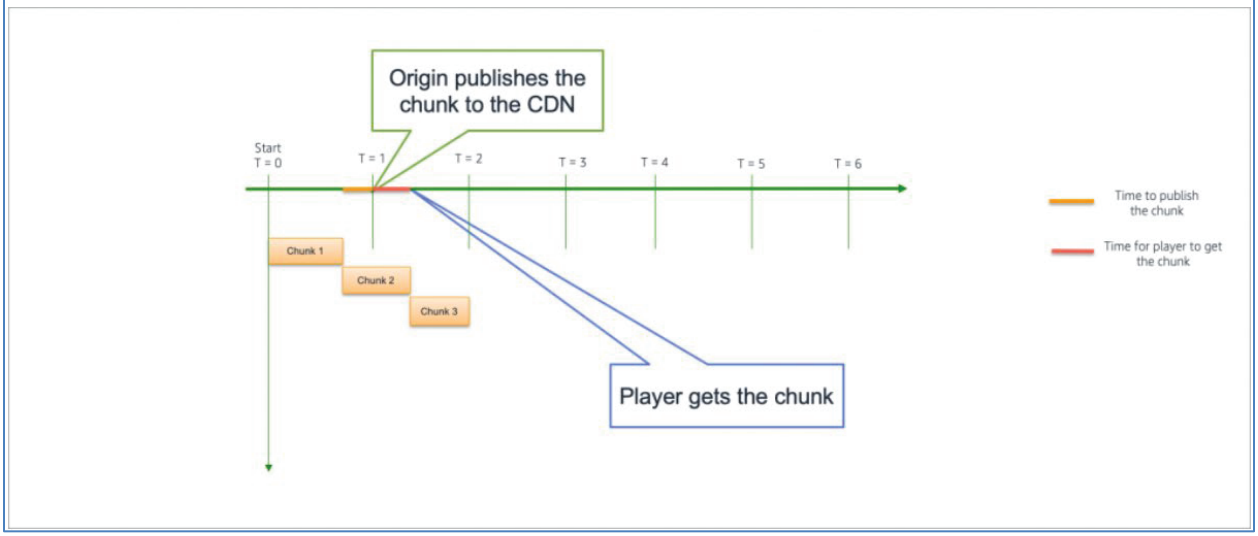
This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

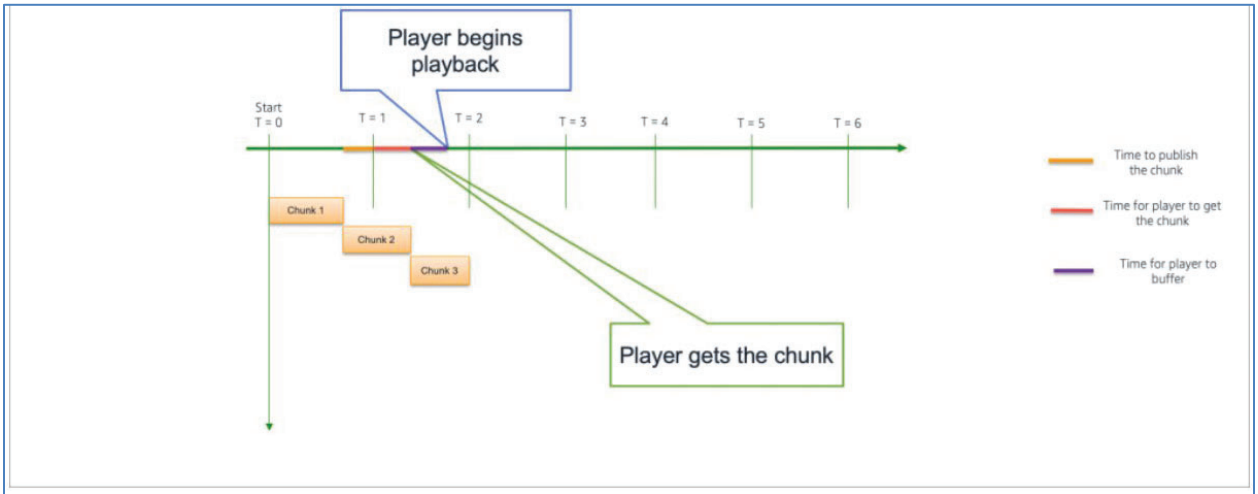
When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.

When using chunked transfer encoding, video delivery an start much sooner.



Once the chunk is pushed to MediaStore, it is available for delivery via a CDN like CloudFront. CloudFront will now be able to pull this chunk, and collapse all corresponding requests for this chunk.





The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

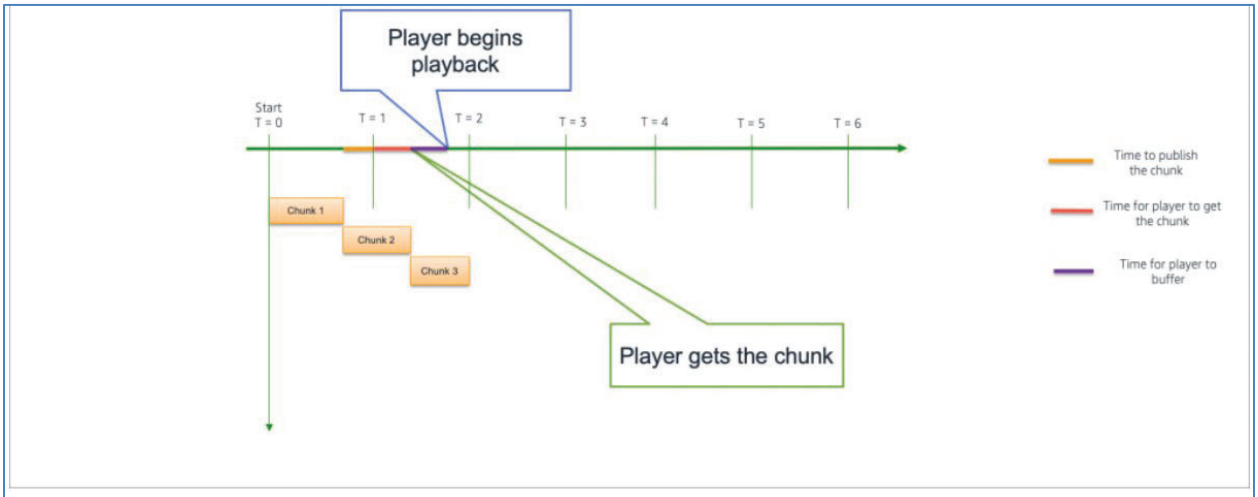
See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

88. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising identifying a range of content surrounding the bookmarked position in the second digital content as content to

be retained (e.g., content of a next chunk overlapping a buffer time of a previous chunk, and/or the ability go forward and backward from a progress bar position).

> In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.
>
> This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.
>
> When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

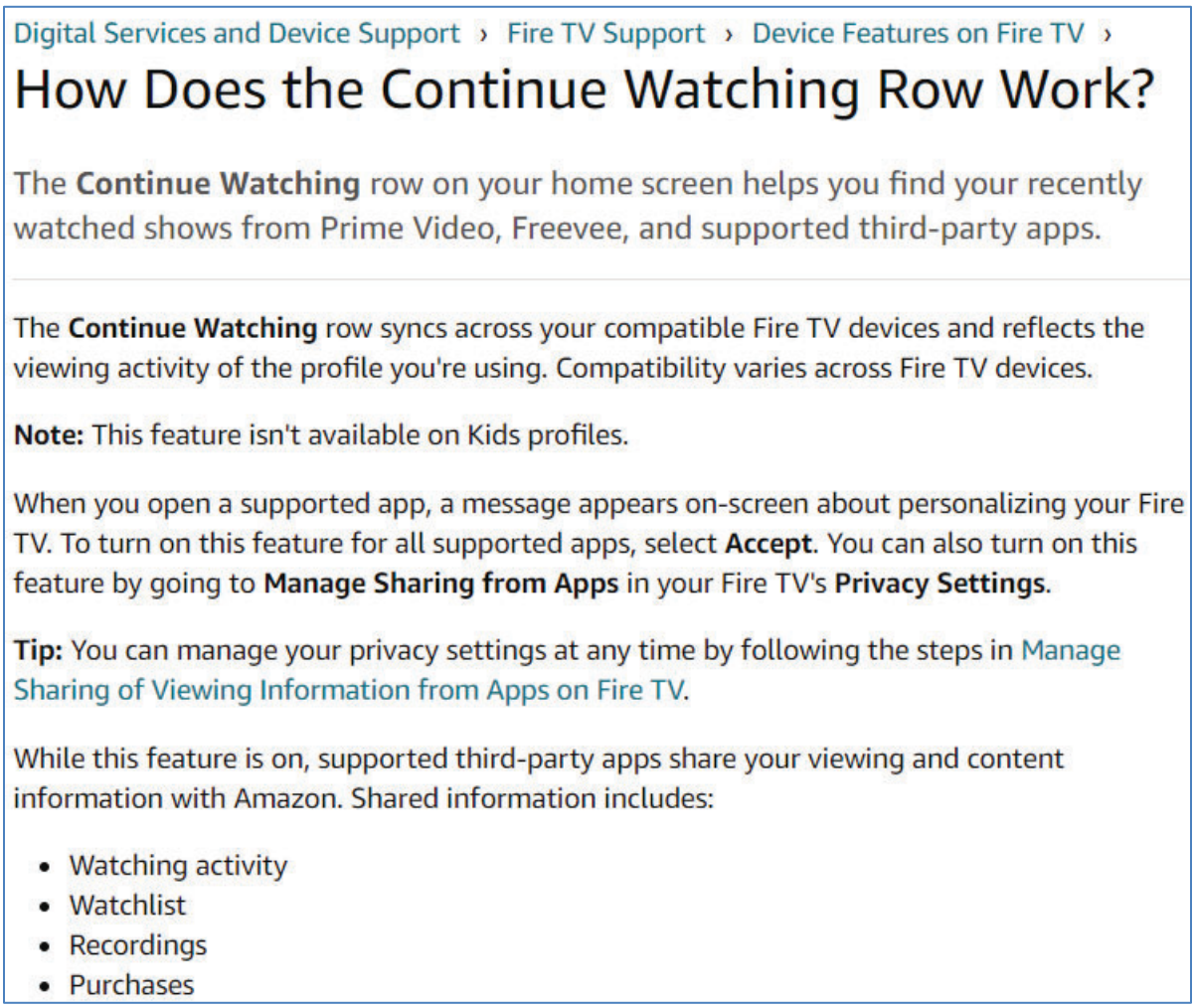


> The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

89. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising releasing storage

resources allocated to all content of the second digital content that is not identified as content to be retained on the second client device (e.g., clearing App data, and/or the "storage optimizing feature").



Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

Digital Services and Device Support › Fire TV Support › Content Issues on Fire TV ›

Clear App Data and Cache on Fire TV

To resolve intermittent app issues, clear the app data and cache.

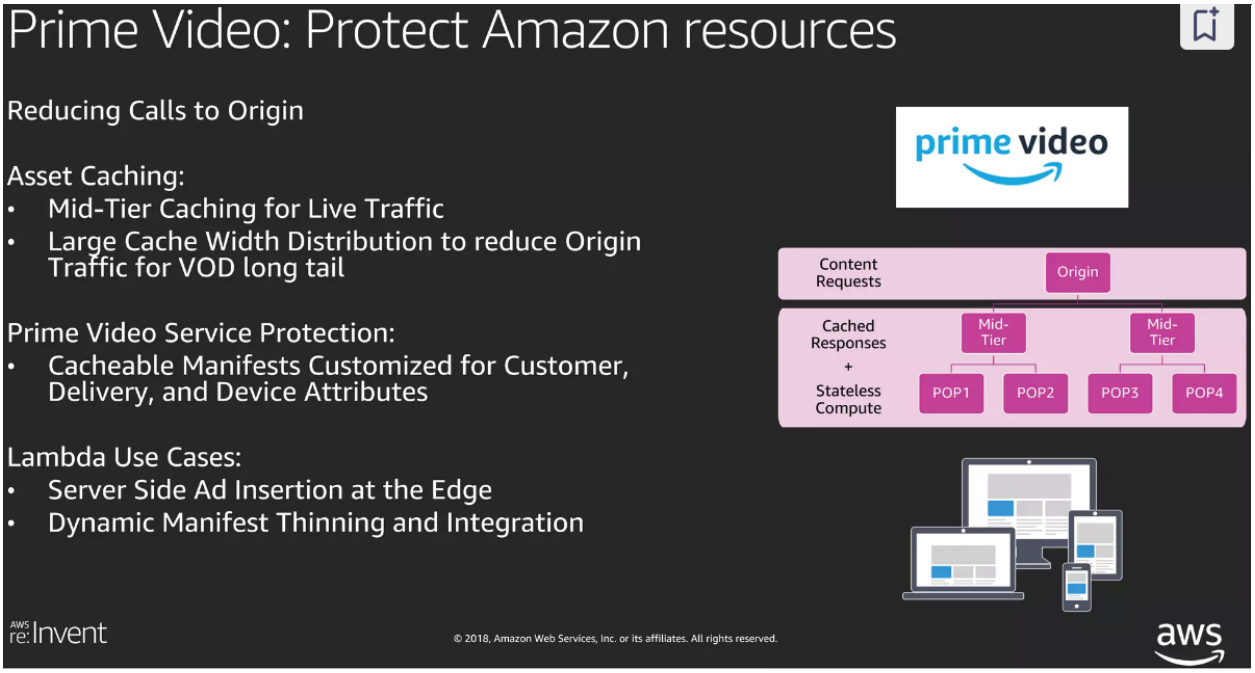
1. Go to **Settings** on your Fire TV.
2. Select **Applications** .
3. In **Manage Installed Applications**, select the app you are having trouble with.
4. Select **Clear Cache**, and then **Clear Data**.

If you need to uninstall an app from your device, go to Uninstall Games and Apps on Your Fire TV.

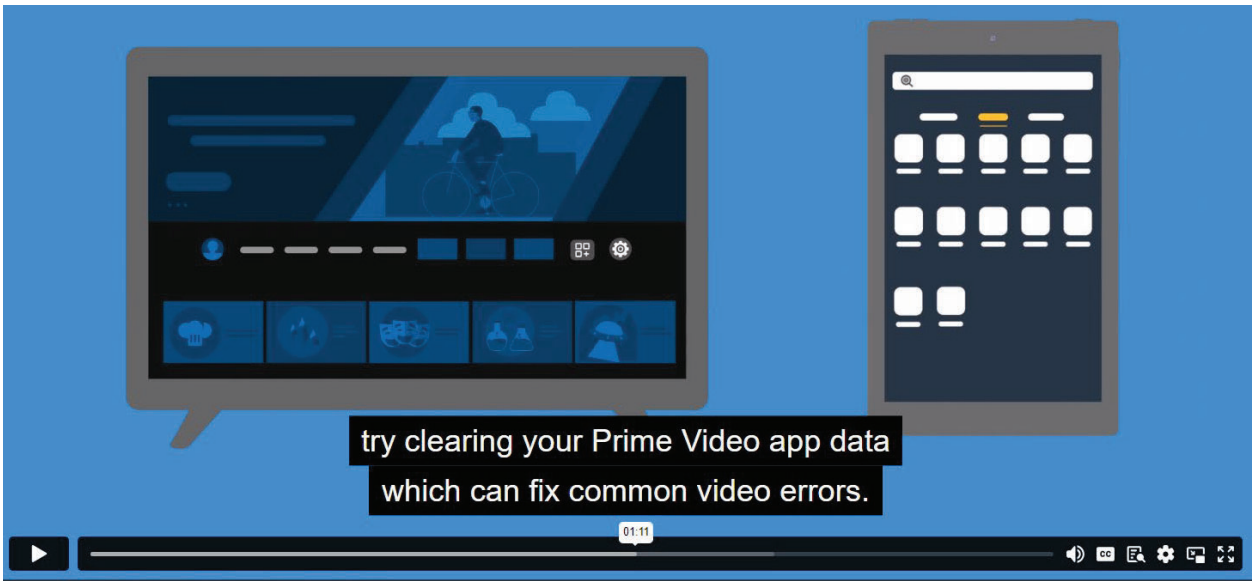
Note: If an app is unused for more than 30 days and the device is running low on storage, the storage optimizing feature will automatically clear the app cache. This feature runs in the background and can't be deactivated.

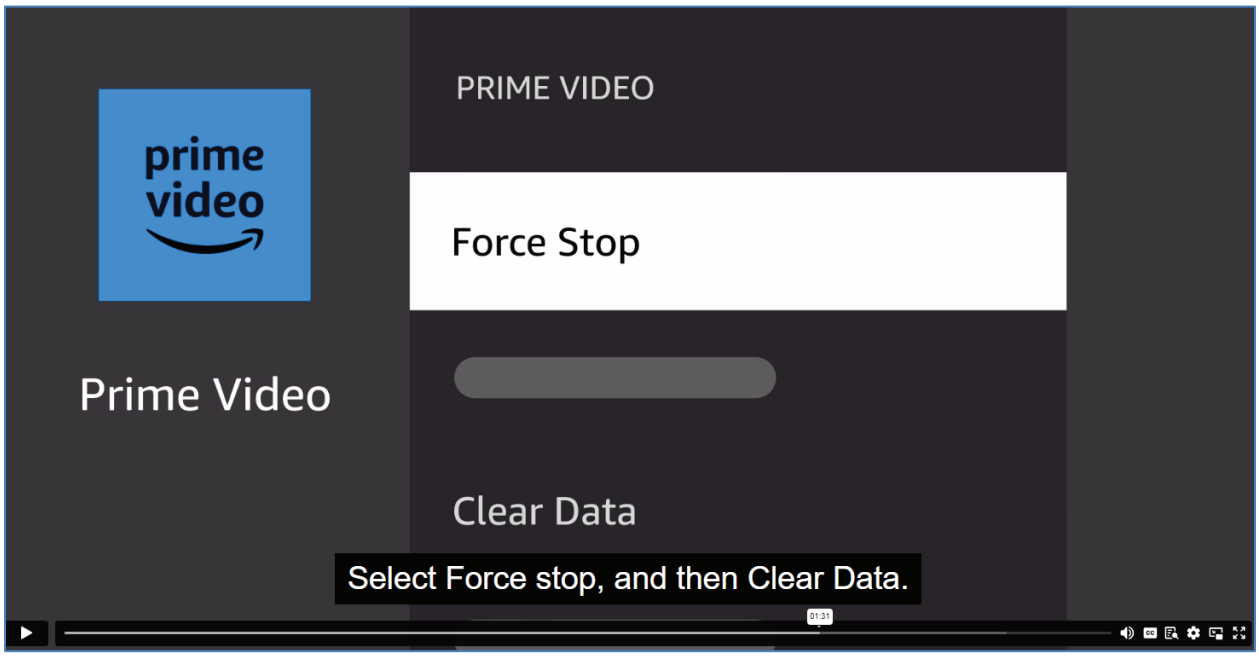
For more help, try our Amazon Fire TV forum.

See https://www.amazon.in/gp/help/customer/display.html?nodeId=GJZBJS5B8VBCGQ48.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.

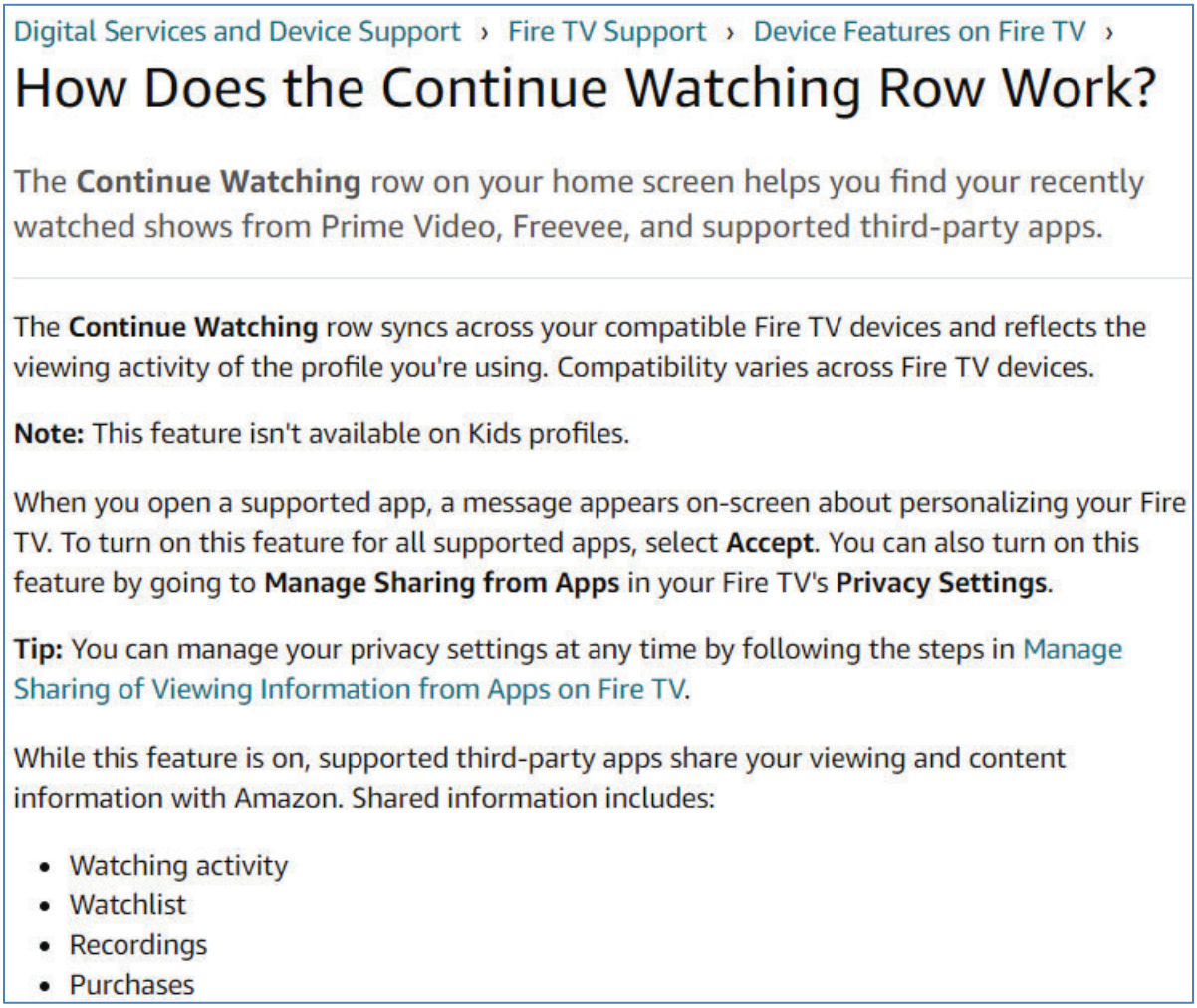




See https://www.amazon.com/gp/help/customer/display.html?ref_=hp_left_v4_sib&nodeId=G93JTA5QNSJT8WWP.

90. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising determining whether sufficient storage is available on the second client device to meet storage demand after releasing the storage resources allocated to the content of the second digital content that is not

identified as content to be retained (e.g., retrieving the maximum number of chunks of media content permitted by the available memory/cache after clearing App data, and/or the "storage optimizing feature").



Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

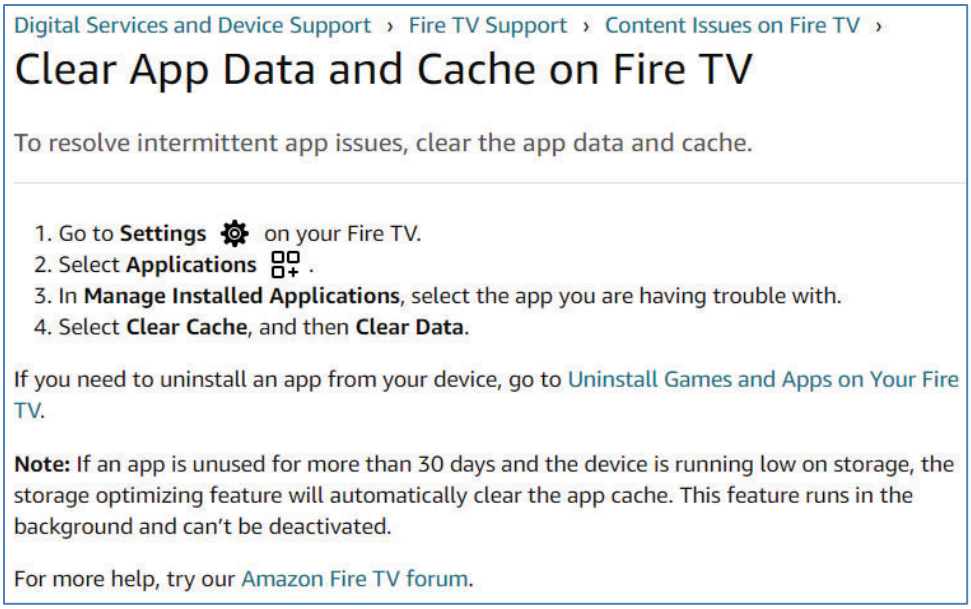
When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

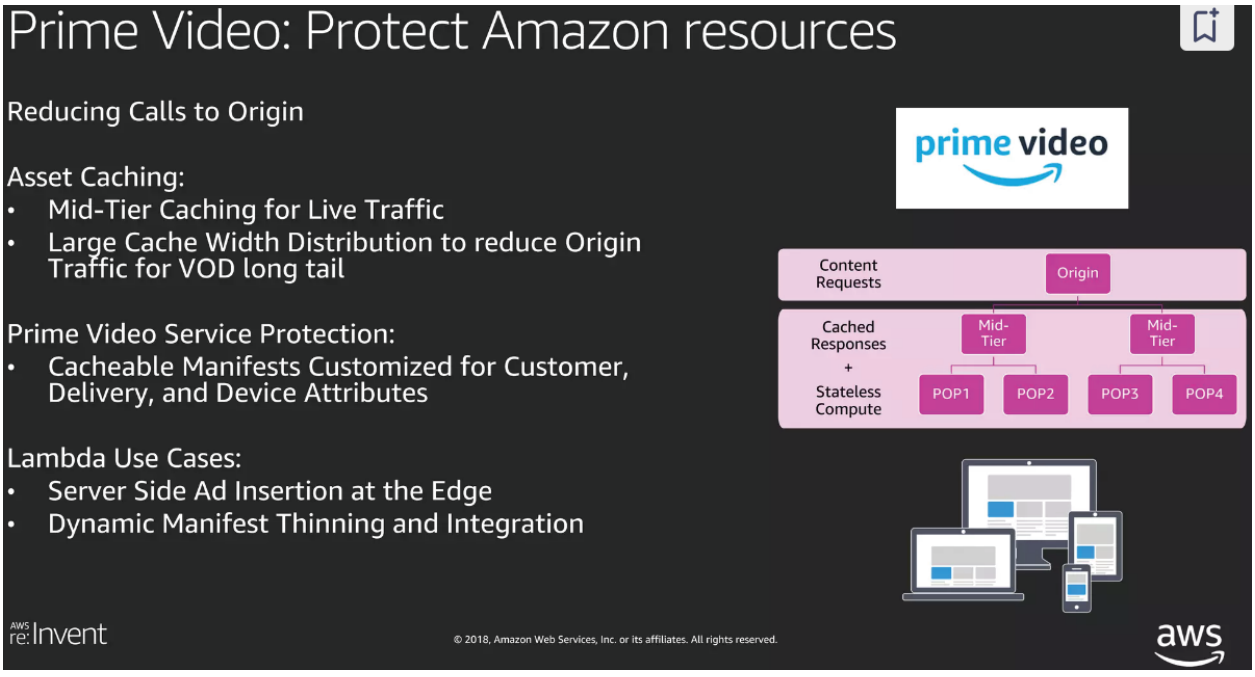
While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

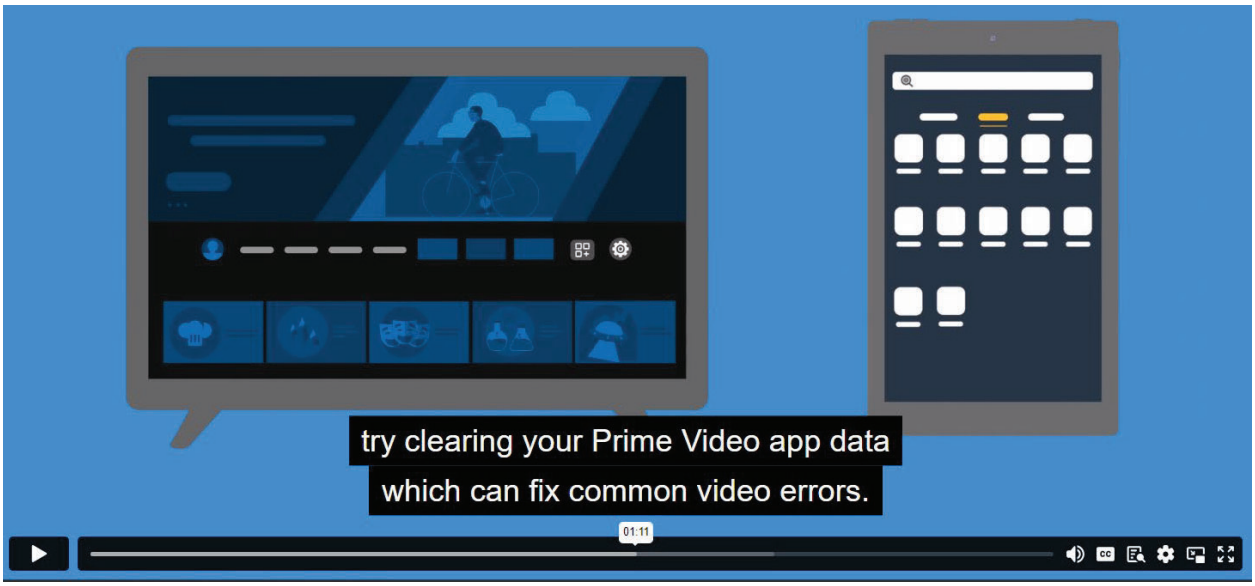
See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

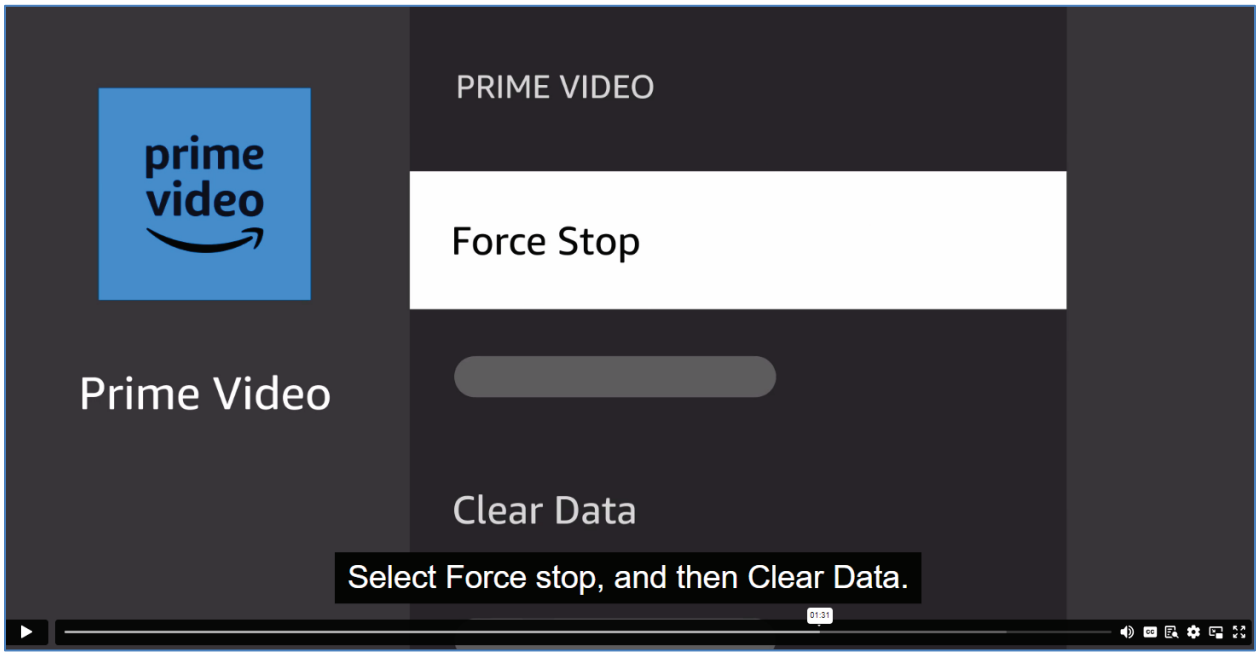


See https://www.amazon.in/gp/help/customer/display.html?nodeId=GJZBJS5B8VBCGQ48.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.

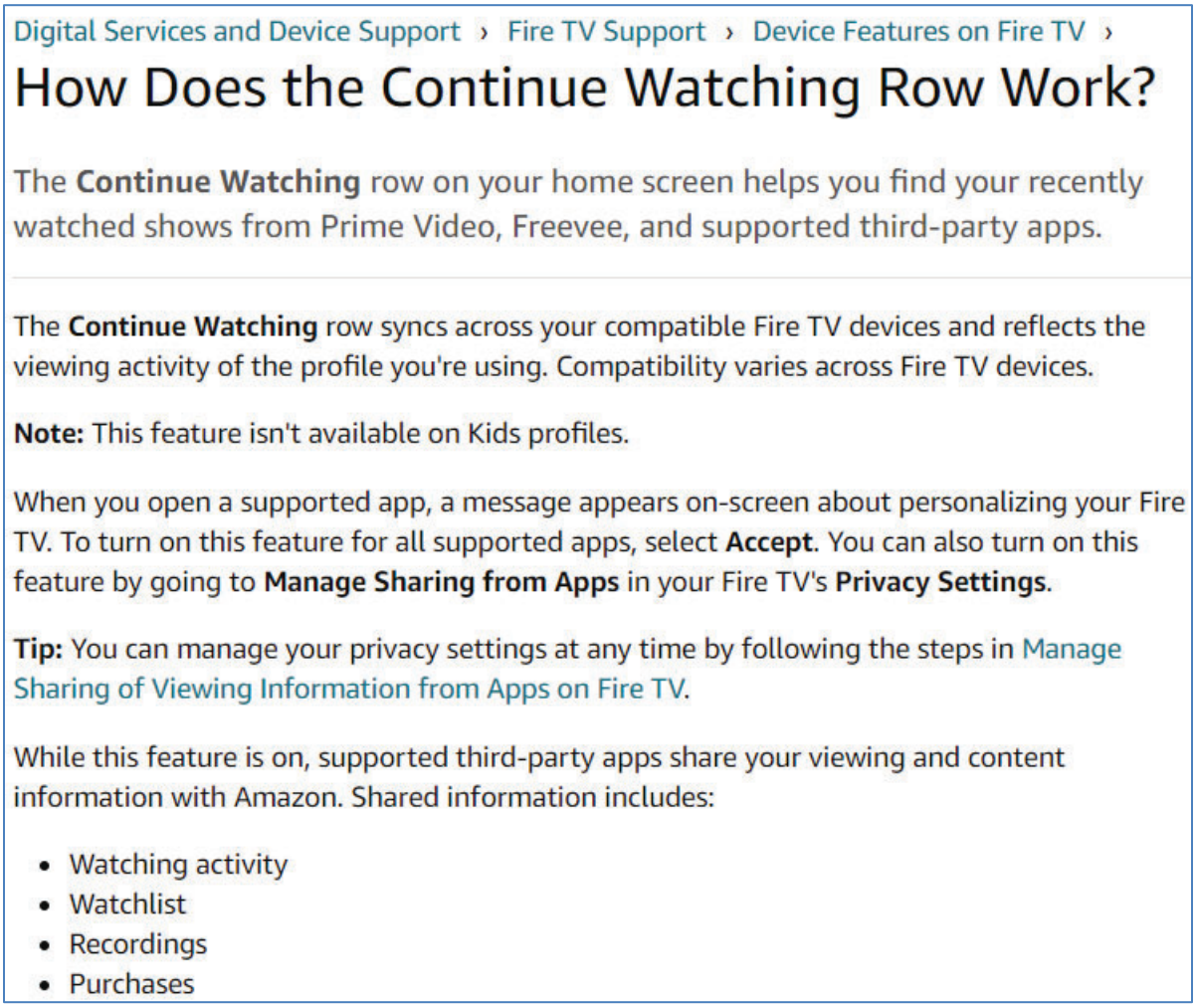




See https://www.amazon.com/gp/help/customer/display.html?ref_=hp_left_v4_sib&nodeId=G93JTA5QNSJT8WWP.

91. On information and belief, one or more components of Prime Video provide a method for rendering digital content across multiple client devices comprising if insufficient storage is available, narrowing the range of content surrounding the bookmarked position that is

identified as content to be retained (e.g., prefetching more or less chunks of media content based on the available memory/cache on the client device).



Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

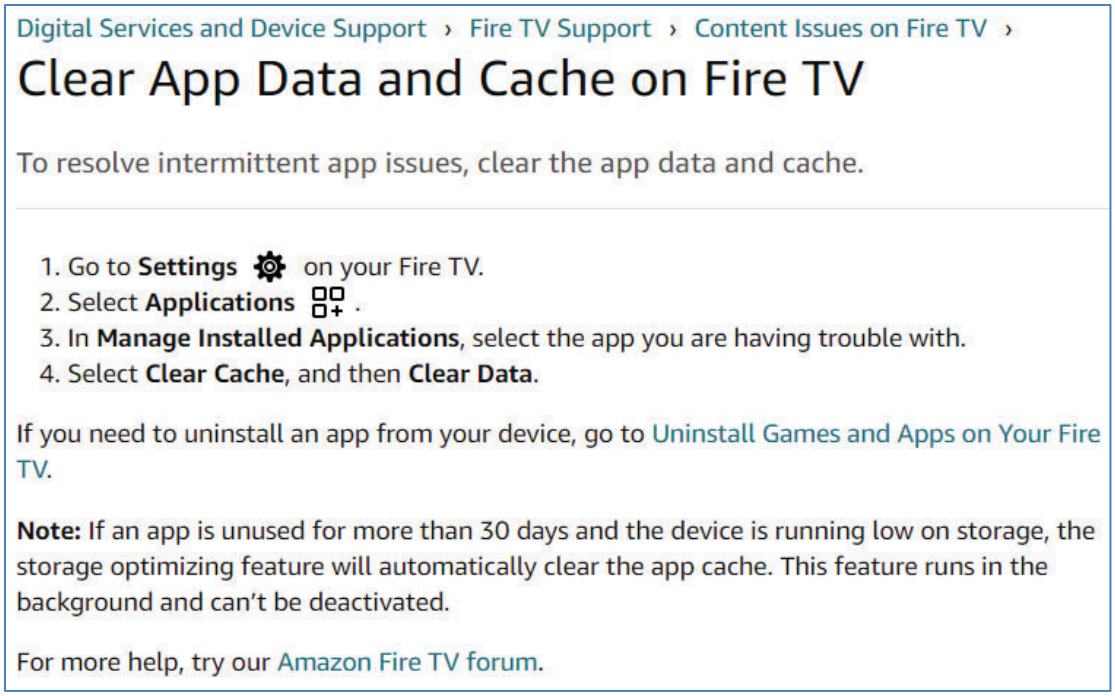
When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

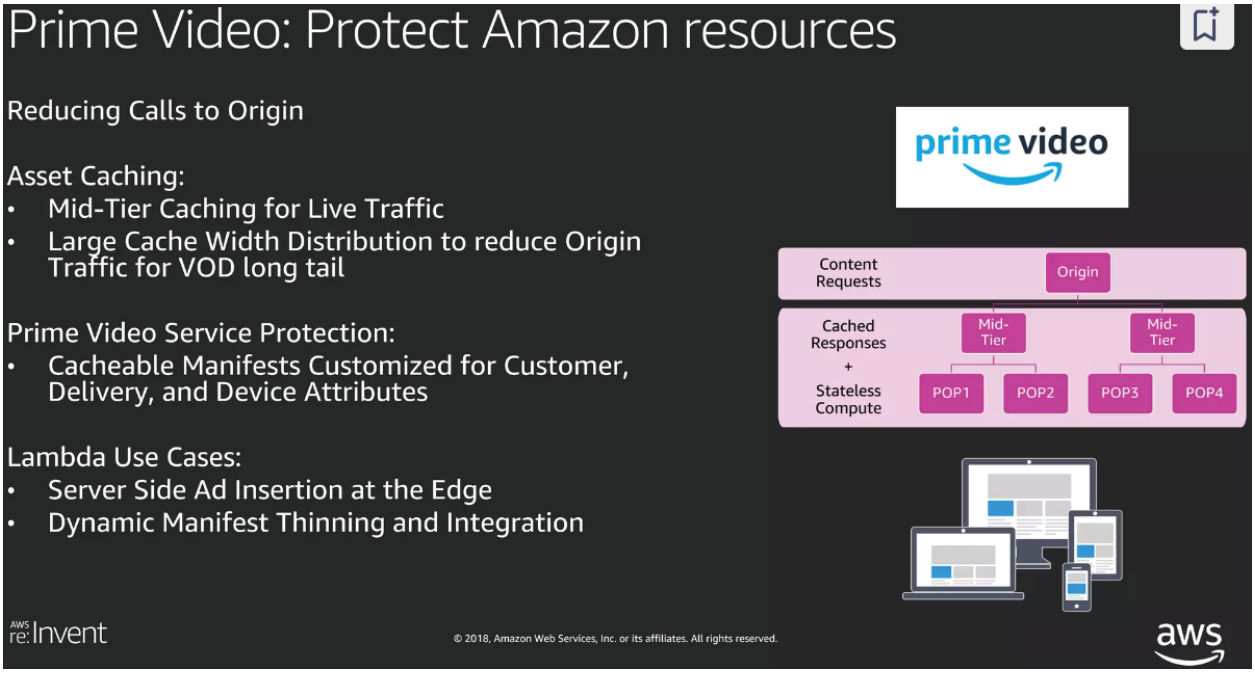
While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.



See https://www.amazon.in/gp/help/customer/display.html?nodeId=GJZBJS5B8VBCGQ48.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.

Once you decide on the relevant scoring formula for your business, you can apply it to your measurement dataset, by segments, to surface dimensions that affect the performance of video streaming. **The most important dimension is internet service provider (ISP)** which is represented by an autonomous system number (ASN). ISPs will typically have different performance characteristics because of their diverse capabilities in terms of network connectivity with various CDNs. Some customers like Amazon Prime Video go beyond a single dimension and add other dimensions like viewer location and device type for more granularity in evaluating the performance of CDNs. In fact, CDNs may optimize video delivery for some streaming formats or user devices. For example, when you use CloudFront's native smooth streaming feature to stream VoD from S3, CloudFront prefetches the next segments into the cache which improves performance for devices using this format like smart TVs. With dimensions now defined, **make sure that you have enough measurements in each segment** (ideally hundreds of thousands per day) for statistical relevance, otherwise, you can group segments. For example, in a country where you don't have enough audience to produce relevant statistics per ASN, you can group data points in one country segment.

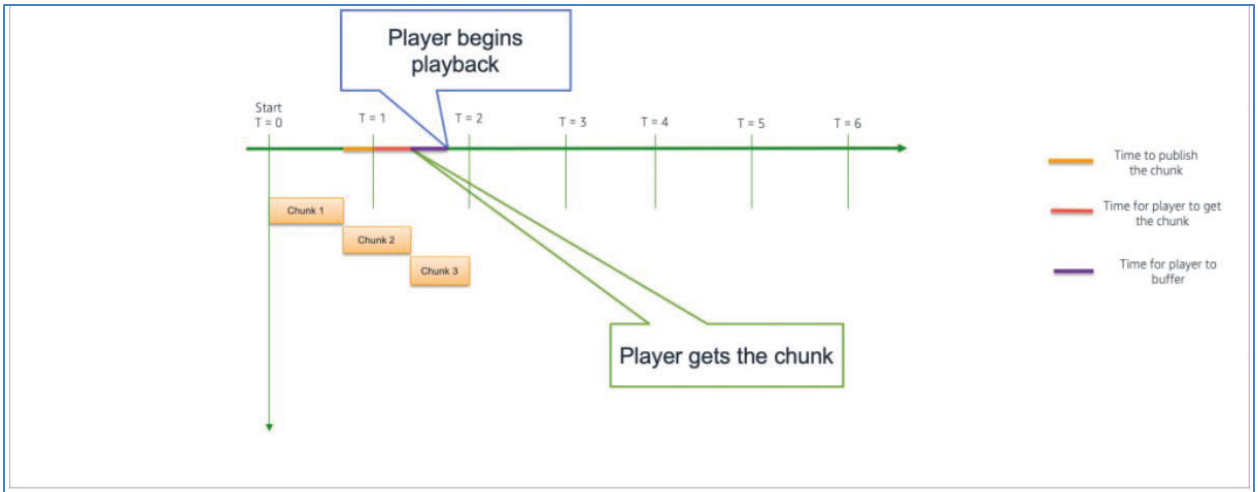
See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.



The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

92. On information and belief, Defendants directly infringe at least claim 1 of the '922 patent in violation of 35 U.S.C. § 271(a) by making, using, selling, selling access to, importing, offering for sale, and/or offering to sell access to Prime Video.

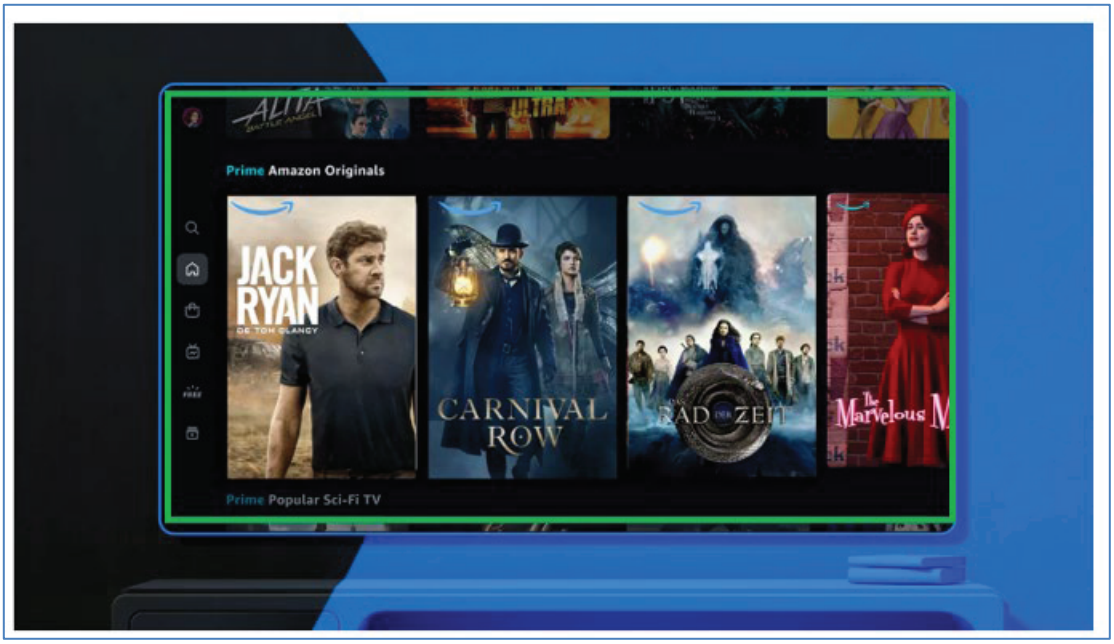
93. Defendants' infringement has damaged Audio Pod and caused/continues to cause it to suffer irreparable harm and damages.

COUNT III - Infringement of the '266 patent by Prime Video

94. Audio Pod repeats, realleges, and incorporates by reference, as if fully set forth here, the allegations of the preceding paragraphs above.

95. On information and belief, Defendants (or those acting on their behalf) make, use, sell, sell access to, import, offer to sell and/or offer to sell access to Prime Video in the United States that infringes (literally and/or under the doctrine of equivalents) at least claim 1 of the '266 patent.

96. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices (*e.g.*, desktops, laptops, smartphones, smart-TVs (e.g., Amazon Fire TV), tablets).



See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

How can I access Prime Video?

Prime Video is available on hundreds of compatible devices. Stream from the web or using the Prime Video
on your smartphone, tablet, set-top box, game console, or select smart TVs.

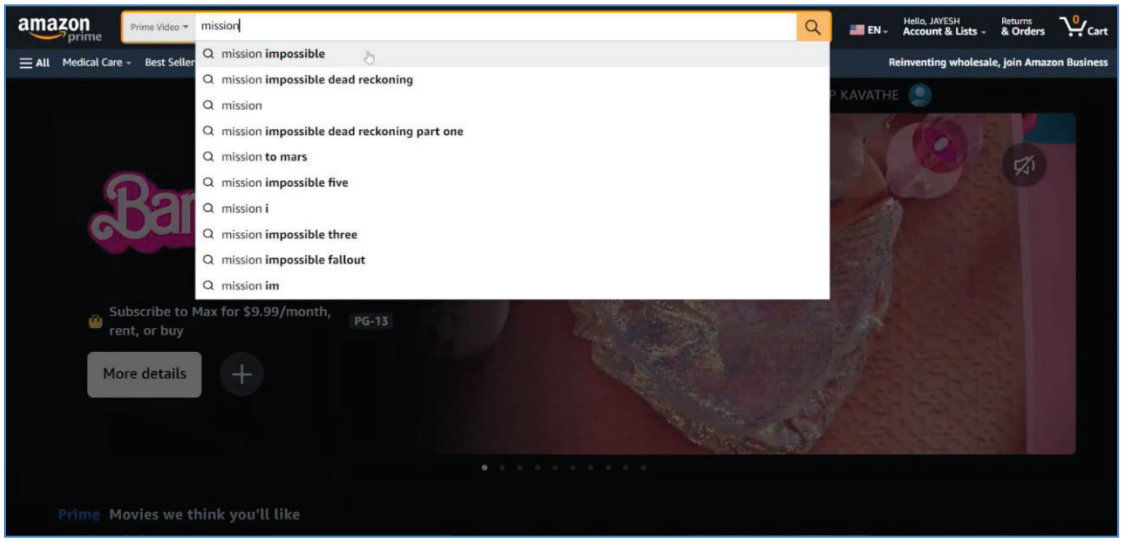
See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

Purchased Videos

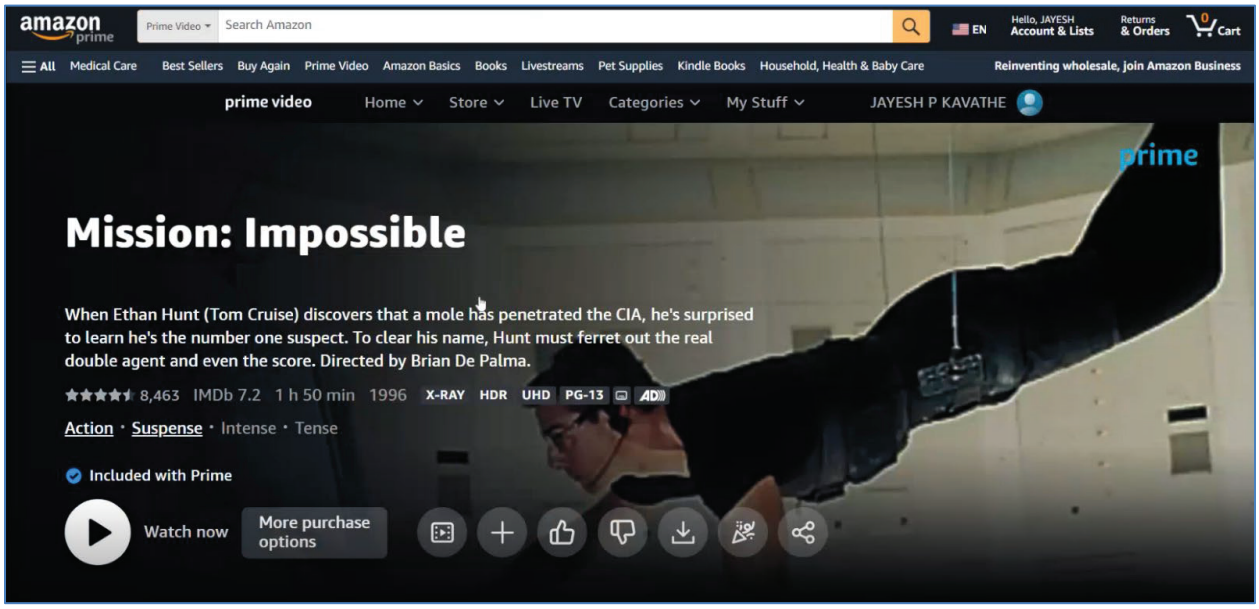
- **Ways to Watch:** When you purchase a video for on-demand viewing, we'll make it available to you to stream and, in most cases, download as follows:
  - **Streaming:** You may stream purchased videos online through your web browser and compatible Internet-connected TVs, Blu-ray players, set-top-boxes, Fire tablets, and other compatible devices. For a list of devices compatible with our service, visit the Compatible Devices page on the website you are using to access Amazon Prime Video at the following links, PrimeVideo.com, Amazon.com, Amazon.co.uk, Amazon.de or Amazon.co.jp (device compatibility may vary by location). You may stream up to three videos at the same time using the same Amazon account. You may stream the same video to no more than two devices at a time.

See https://www.primevideo.com/help?nodeId=202095500.

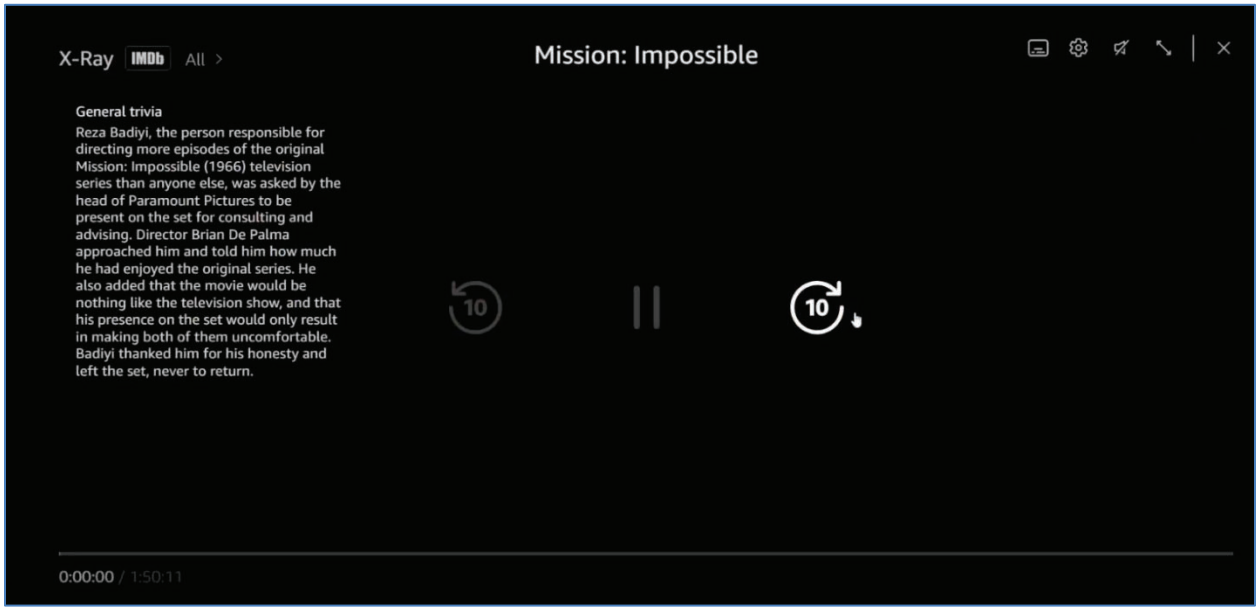
97. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising rendering (e.g., presenting the media content) on a first client device (*e.g.*, a desktop, laptop, smartphone, smart-TV (e.g., Amazon Fire TV), tablet) at least a portion (e.g., "video chunk") of primary digital content.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



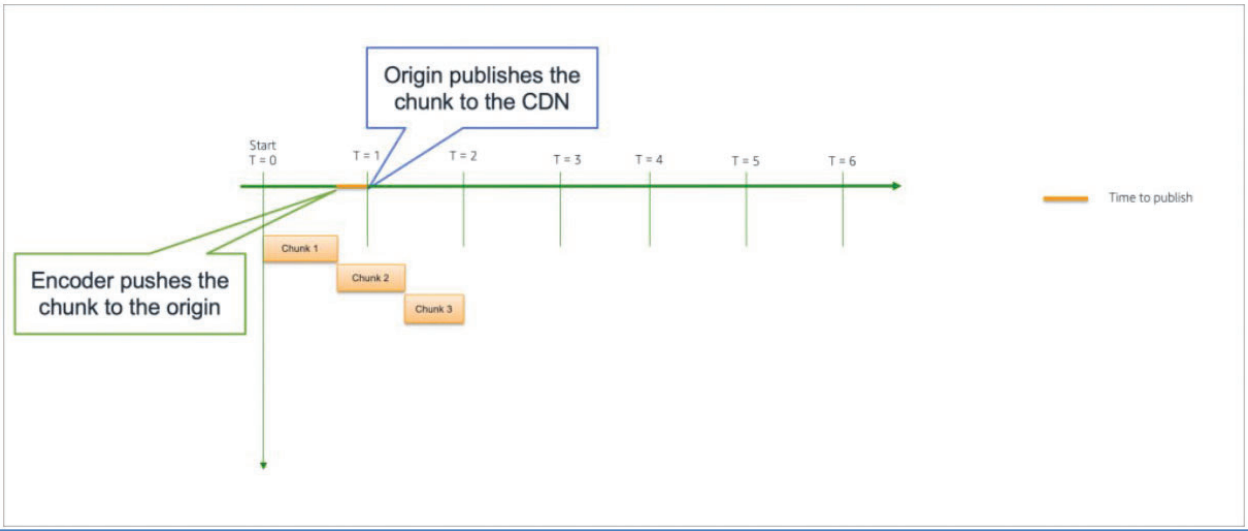
See Screenshot taken during product testing.

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

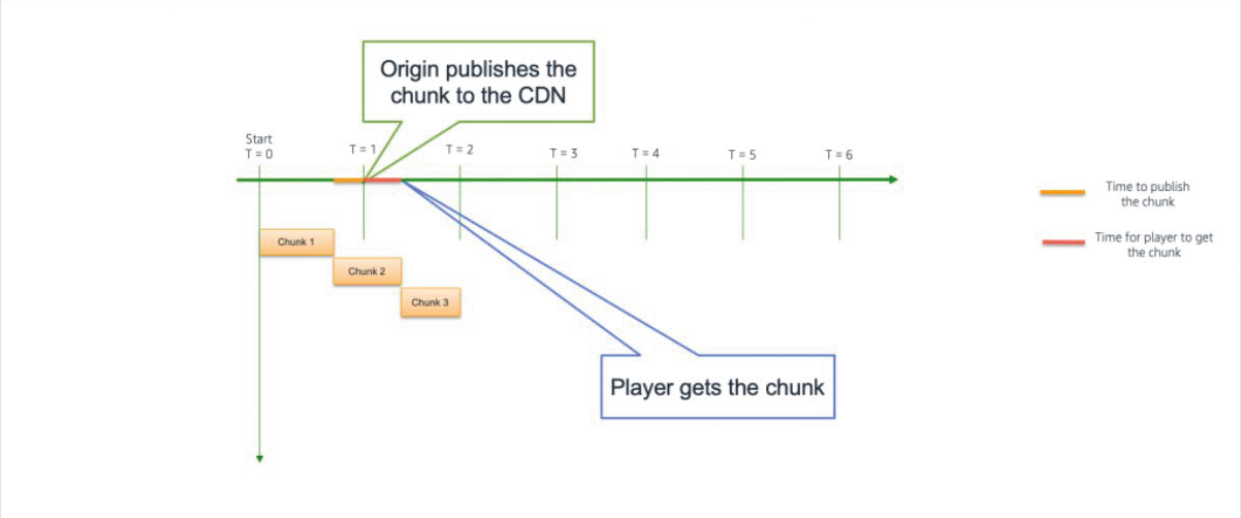
This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

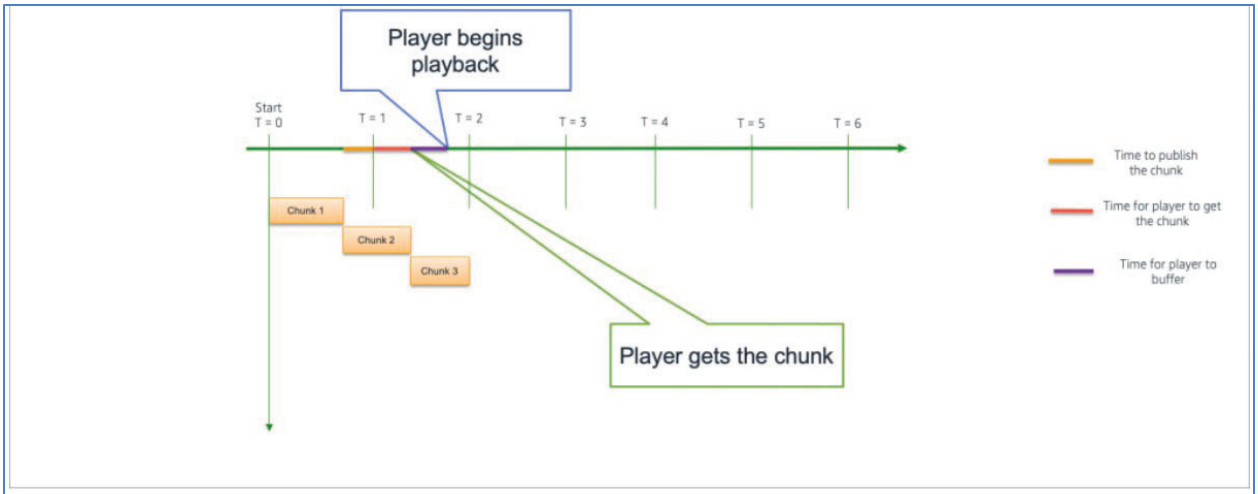
When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.

When using chunked transfer encoding, video delivery an start much sooner.



Once the chunk is pushed to MediaStore, it is available for delivery via a CDN like CloudFront. CloudFront will now be able to pull this chunk, and collapse all corresponding requests for this chunk.

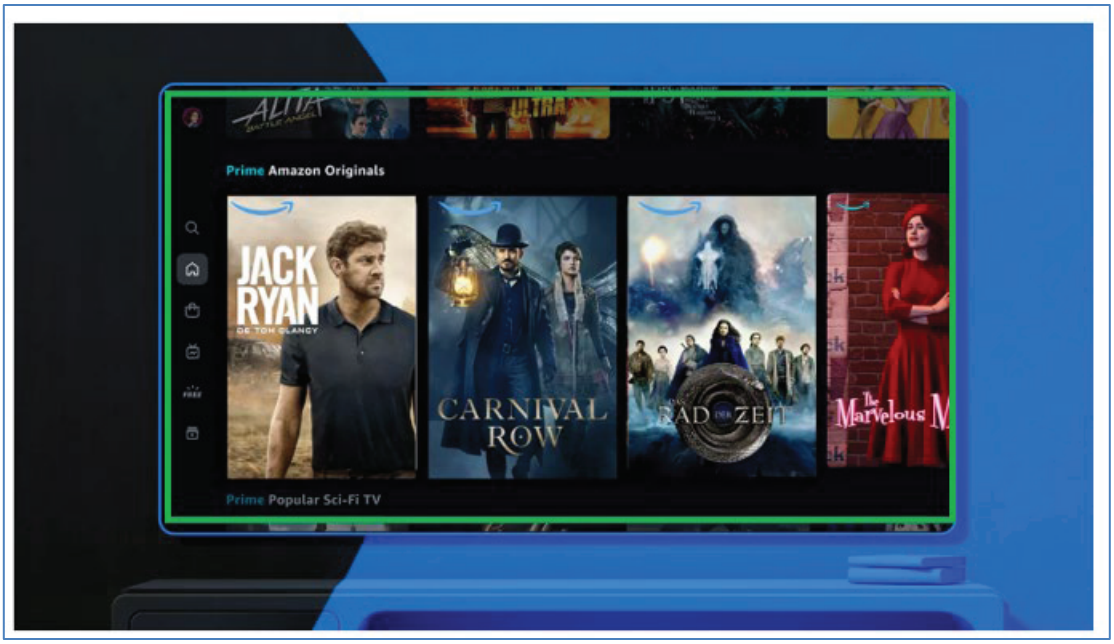




The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

98. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising determining on the first client device an identifier (e.g., a data element representing at least a particular item of media content (e.g., a movie, show, and/or episode)) corresponding to the primary digital content, wherein the identifier identifies a descriptor (e.g., Media Presentation Description ("MPD")) of the primary content.



See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

How can I access Prime Video?

Prime Video is available on hundreds of compatible devices. Stream from the web or using the Prime Video app on your smartphone, tablet, set-top box, game console, or select smart TVs.

See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

Purchased Videos

- **Ways to Watch:** When you purchase a video for on-demand viewing, we'll make it available to you to stream and, in most cases, download as follows:
  - **Streaming:** You may stream purchased videos online through your web browser and compatible Internet-connected TVs, Blu-ray players, set-top-boxes, Fire tablets, and other compatible devices. For a list of devices compatible with our service, visit the Compatible Devices page on the website you are using to access Amazon Prime Video at the following links, PrimeVideo.com, Amazon.com, Amazon.co.uk, Amazon.de or Amazon.co.jp (device compatibility may vary by location). You may stream up to three videos at the same time using the same Amazon account. You may stream the same video to no more than two devices at a time.

See https://www.primevideo.com/help?nodeId=202095500.

Scope of MPEG-DASH

Figure 2 illustrates a simple streaming scenario between an HTTP server and a DASH client. In this figure, the multimedia content is captured and stored on an HTTP server and is

delivered using HTTP. The content exists on the server in two parts: Media Presentation Description (MPD), which describes a manifest of the available content, its various alternatives, their URL addresses, and other characteristics; and segments, which contain the actual multimedia bitstreams in the form of chunks, in single or multiple files.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/T_MM1_TheMPEGDASHStandard.pdf.

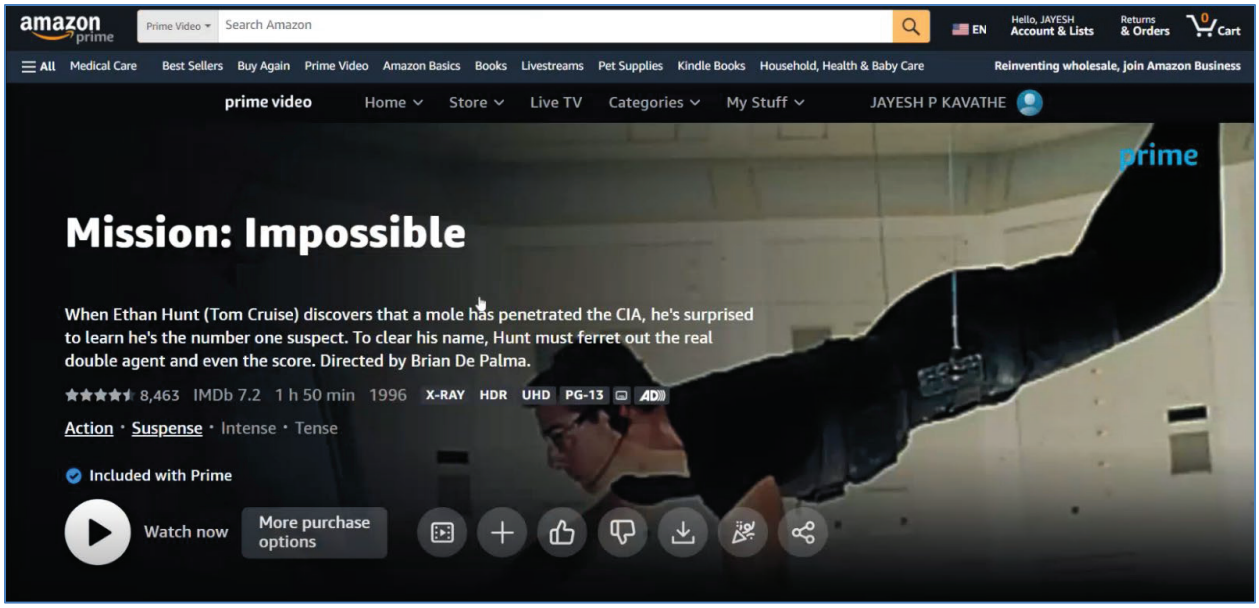
MPEG-DASH: The Standard for Multimedia Streaming Over Internet

In order to play the content, the DASH client first obtains the MPD. The MPD can be delivered using HTTP, email, thumb drive, broadcast or other transports. By parsing the MPD, the DASH client learns about the timing of the program, the availability of media content, the media types, resolutions, minimum and maximum bandwidths and the existence of various encoded alternatives of multimedia components, the accessibility features and the required digital right management (DRM), the location of each media component on the network and other characteristic of the content. Using this information, the DASH client selects the appropriate encoded alternative and starts streaming of the content by fetching the segments using HTTP GET requests.

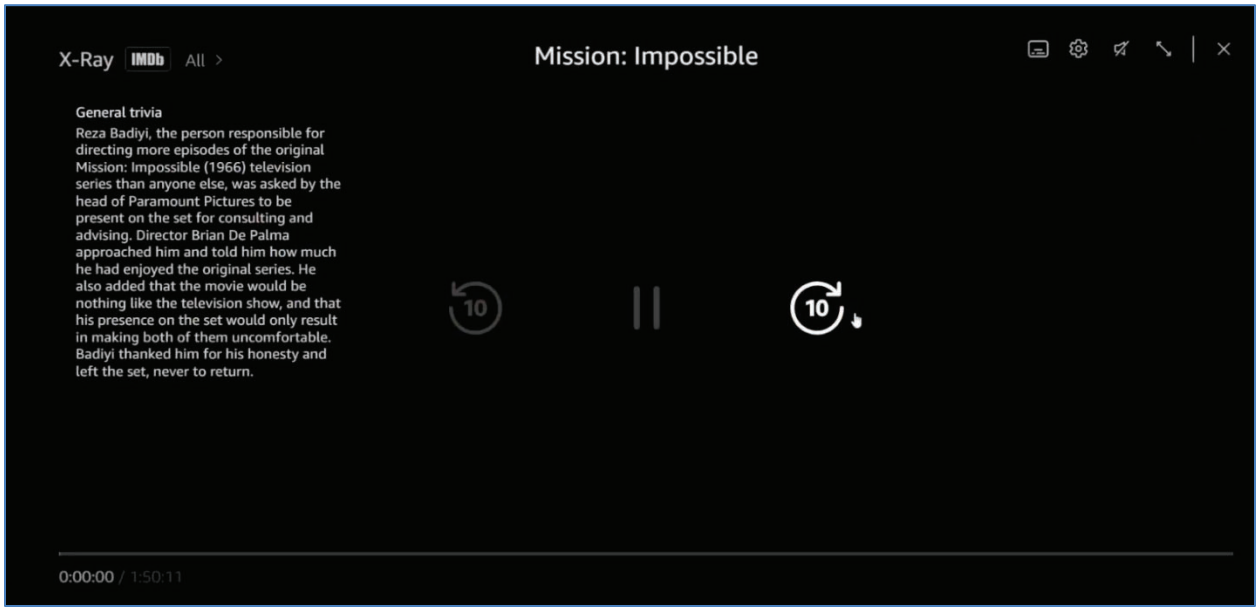
Dynamic HTTP streaming requires various bitrate alternatives of the multimedia content to be available at the server. In addition, the multimedia content may consist of several media components (e.g. audio, video, text), each of which may have different characteristics. In MPEG-DASH, these characteristics are described by MPD which is an XML document.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf.

99. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising determining on the first client device a first position (e.g., creating a data component that includes timing information to indicate the stop/pause position for the media stream, and/or a progress bar or cursor) in the primary digital content.



See Screenshot taken during product testing.



See Screenshot taken during product testing.

100. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising transferring the identifier and the first position from the first client device to a second client device via a network accessible library (e.g., the "Continue Watching" and/or "Multi-Device Sync" feature).

Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

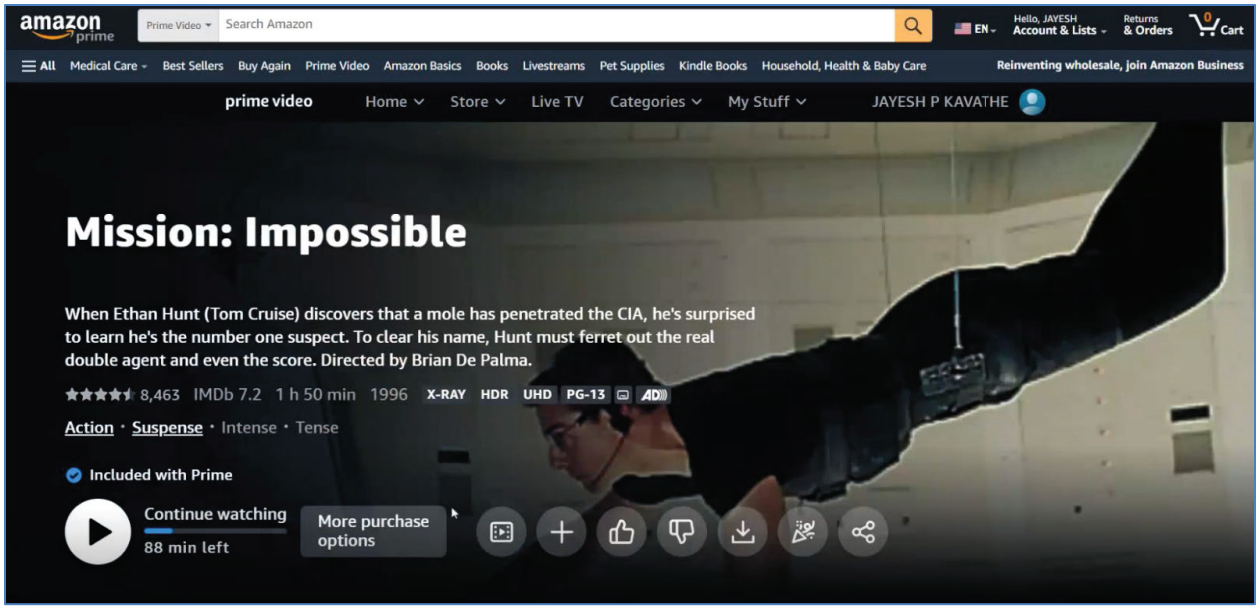
While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

7. Multi-Device Sync: Amazon Prime Video offers multi-device syncing, allowing you to start watching a TV show or movie on one device and continue on another. For example, if you start streaming on your smartphone, you can pause and resume playback seamlessly on your smart TV or tablet using the same Amazon account.

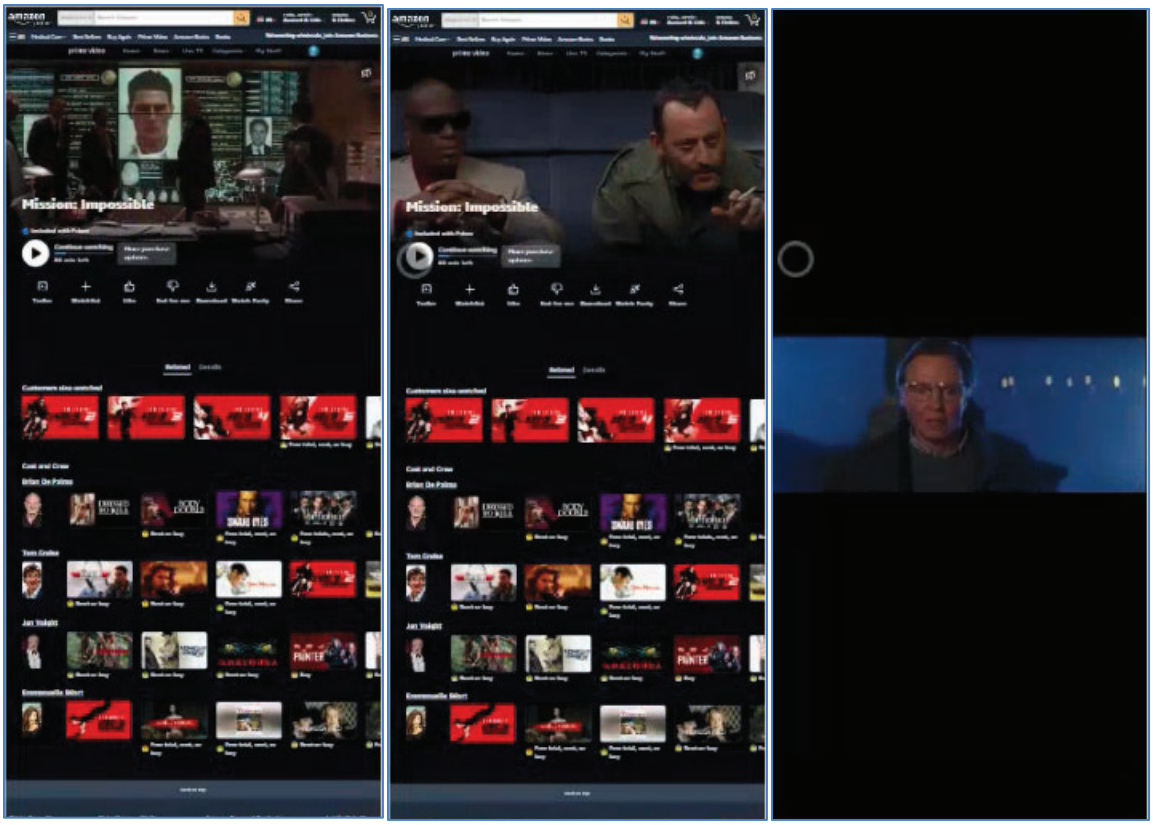
See https://citizenside.com/entertainment/how-to-watch-tv-with-amazon-prime/.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



See Screenshot taken during product testing.

101. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising downloading the descriptor from the network accessible library to the second client device by using the identifier (e.g., through the "Continue Watching" and/or "Multi-Device Sync" feature).

Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

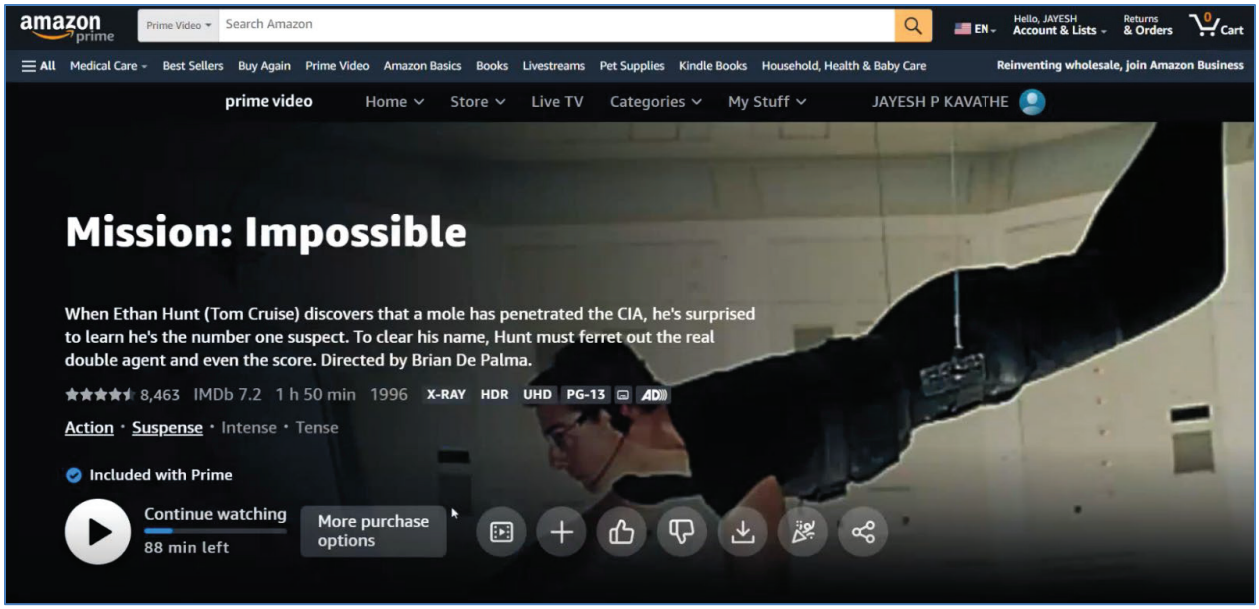
While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

7. Multi-Device Sync: Amazon Prime Video offers multi-device syncing, allowing you to start watching a TV show or movie on one device and continue on another. For example, if you start streaming on your smartphone, you can pause and resume playback seamlessly on your smart TV or tablet using the same Amazon account.

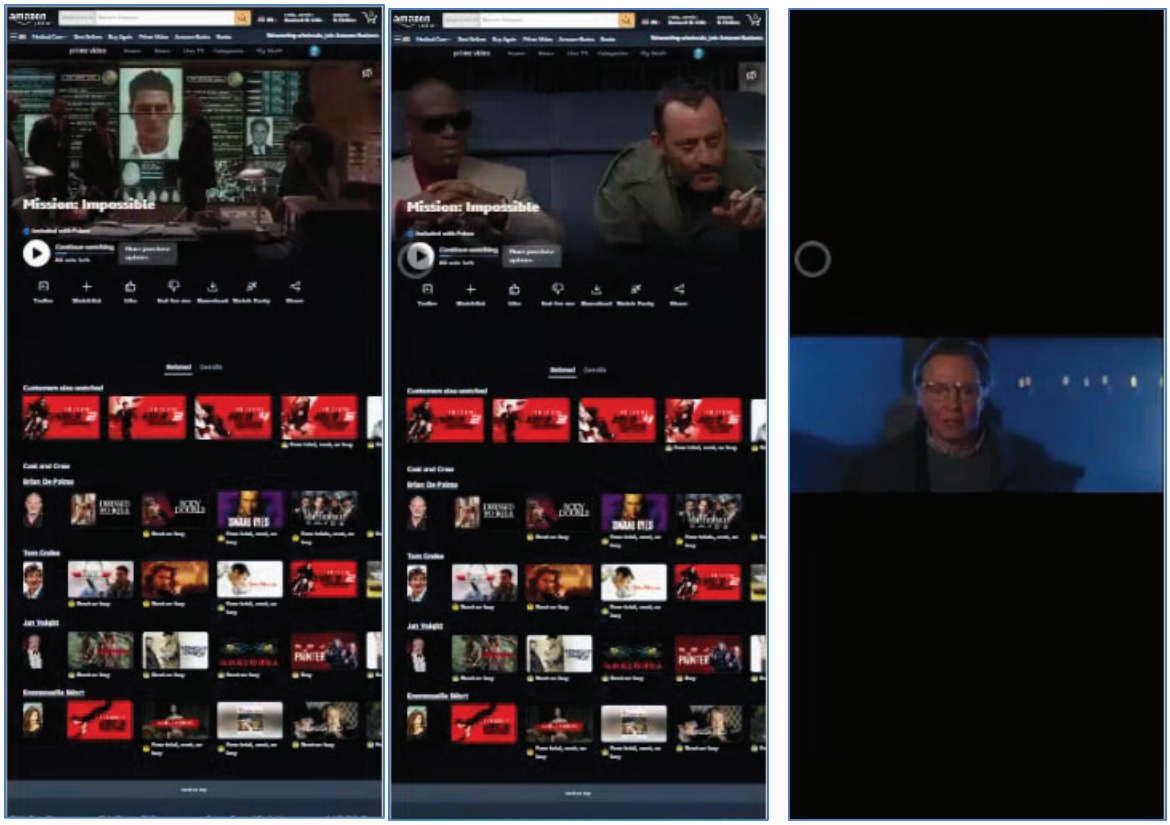
See https://citizenside.com/entertainment/how-to-watch-tv-with-amazon-prime/.



See Screenshot taken during product testing.

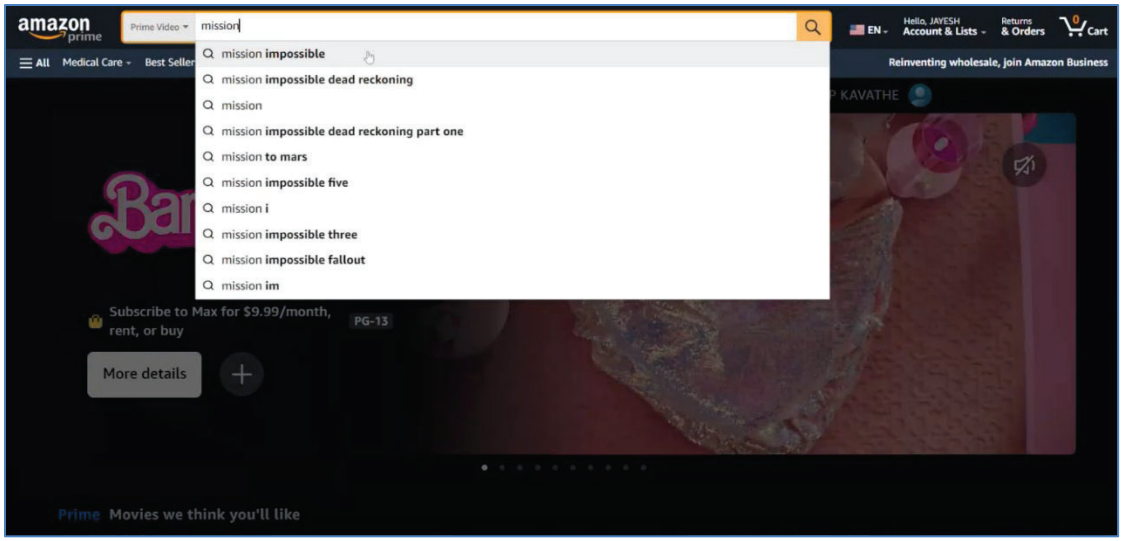


See Screenshot taken during product testing.

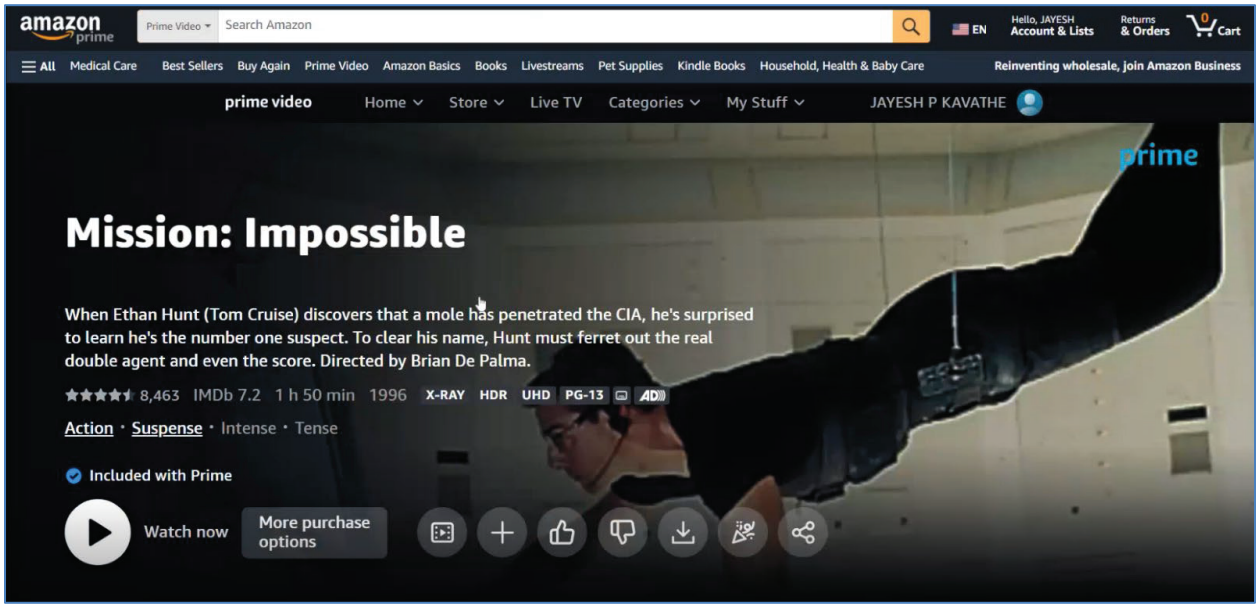


See Screenshot taken during product testing.

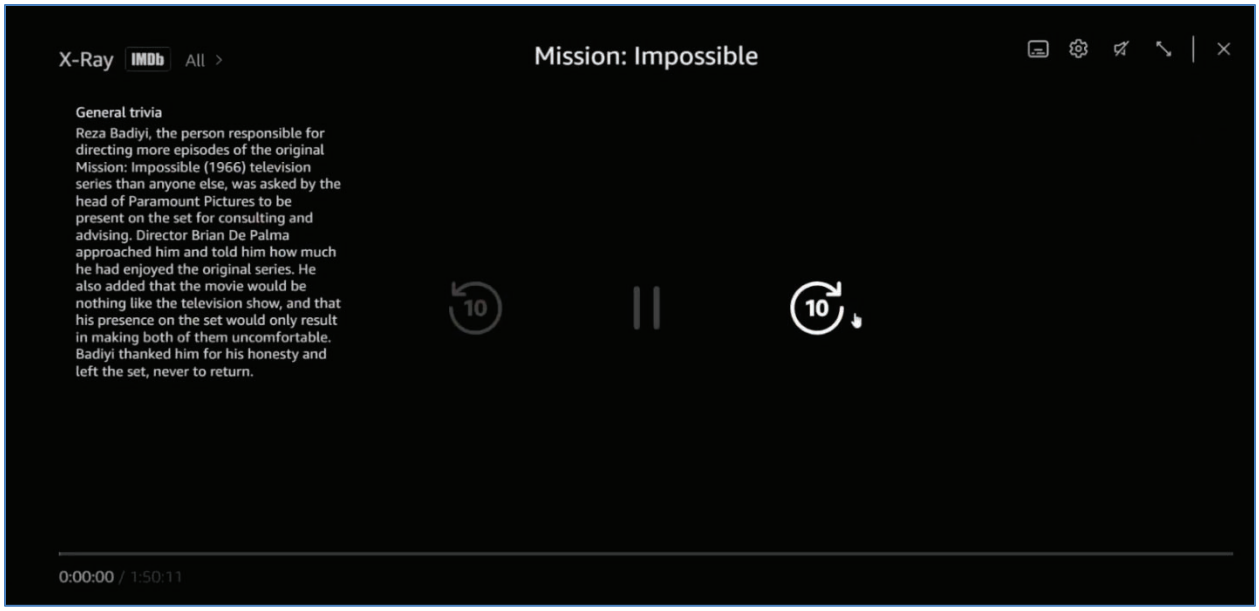
102. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising rendering on the second client device at least a portion (e.g., "video chunk") of secondary other digital content associated with the primary digital content (e.g., subtitle streams corresponding to a media work (e.g., a movie or visual work)) by using the descriptor and the first position, wherein the secondary digital content is ancillary to the primary digital content (e.g., subtitle streams are ancillary compared to the movie or visual work), and wherein the secondary digital content is rendered on the second client device simultaneously and in synchronization with the rendering of the primary digital content on the first client device (e.g., through the "Continue Watching" and/or "Multi-Device Sync" feature).



See Screenshot taken during product testing.



See Screenshot taken during product testing.



See Screenshot taken during product testing.



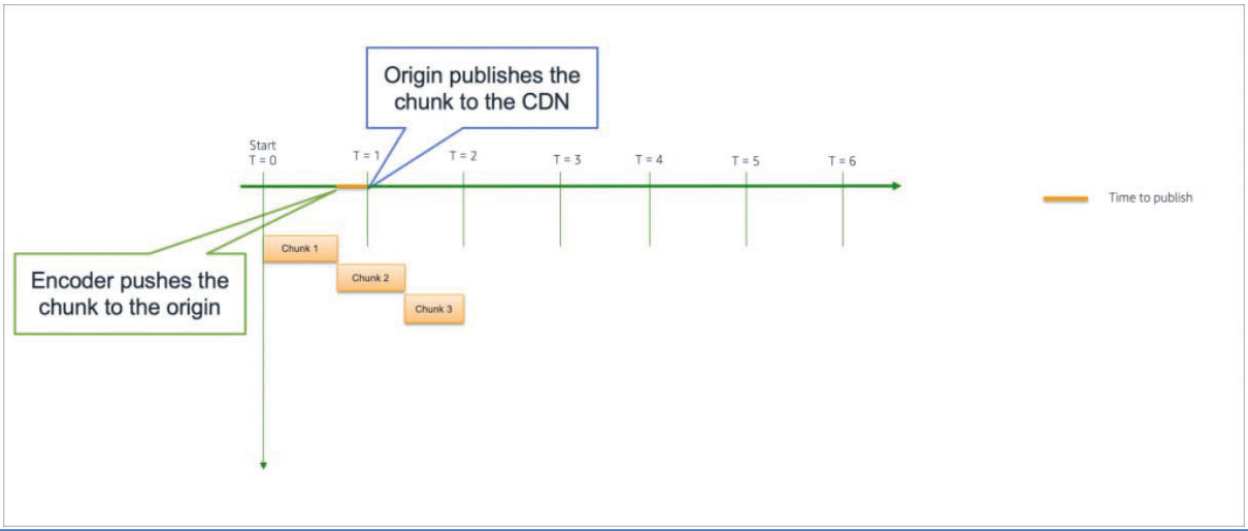
See Screenshot taken during product testing.

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

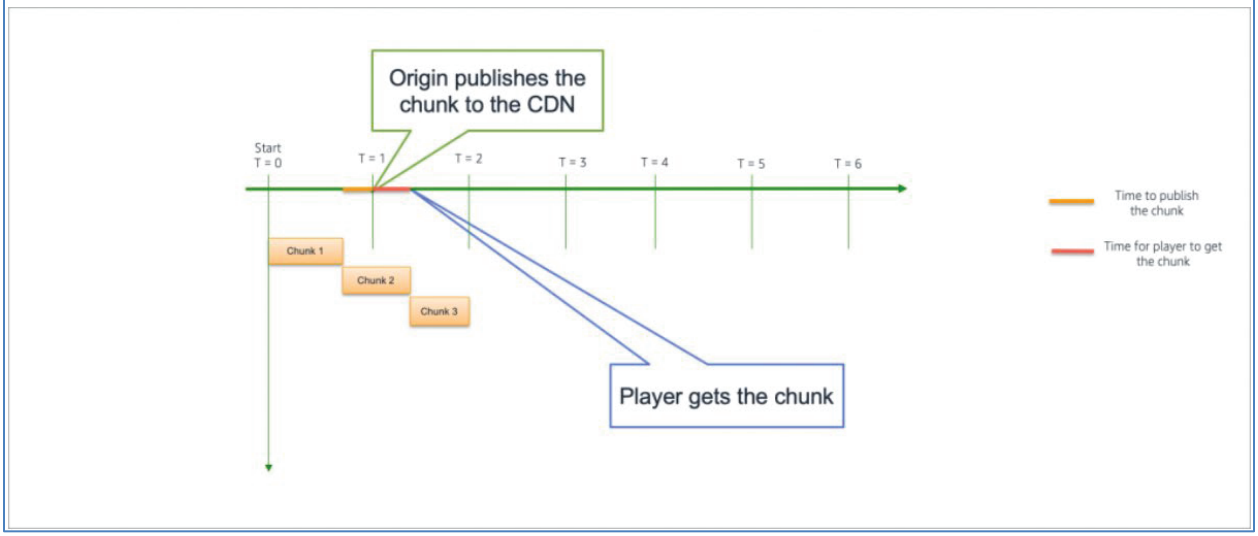
This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

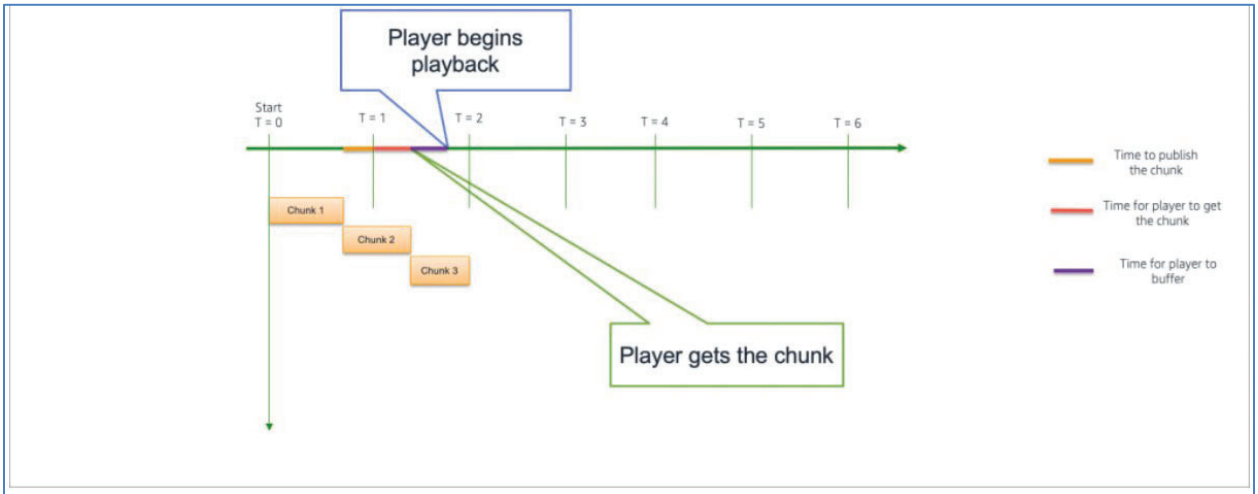
When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.

When using chunked transfer encoding, video delivery an start much sooner.



Once the chunk is pushed to MediaStore, it is available for delivery via a CDN like CloudFront. CloudFront will now be able to pull this chunk, and collapse all corresponding requests for this chunk.





The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.



A representation of how Multi-CDN switching for video streaming works

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

Scope of MPEG-DASH

Figure 2 illustrates a simple streaming scenario between an HTTP server and a DASH client. In this figure, the multimedia content is captured and stored on an HTTP server and is delivered using HTTP. The content exists on the server in two parts: 1) Media Presentation Description (MPD) which describes a manifest of the available content, its various alternatives, their URL addresses and other characteristics, and 2) Segments which contain the actual multimedia bitstreams in form of chunks, in single or multiple files.

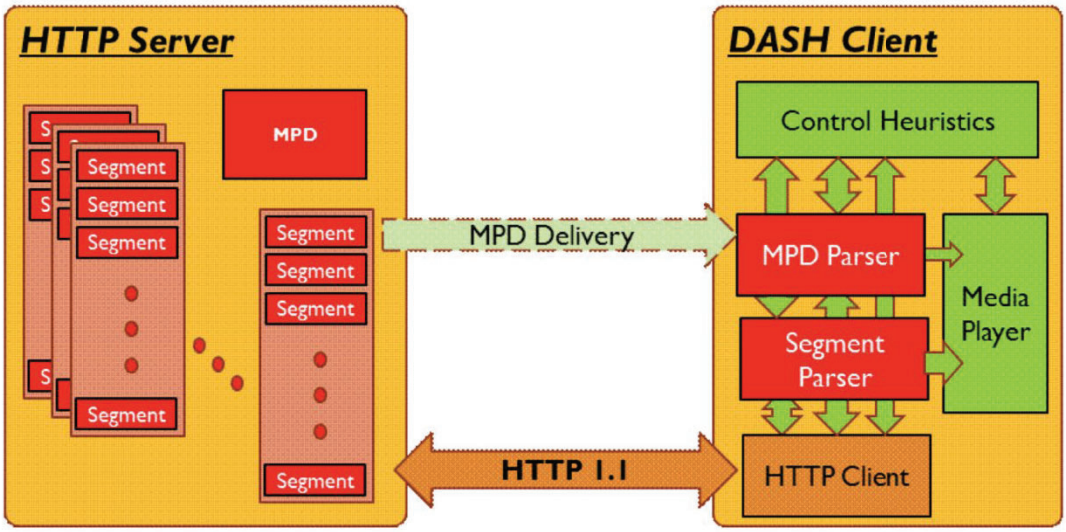


Figure 2. Scope of the MPEG-DASH Standard. The formats and the functionalities of the red blocks are defined by the specification. The clients control heuristics and media players are not within the scope of the standard.

Multimedia Presentation Description

Dynamic HTTP streaming requires various bitrate alternatives of the multimedia content to be available at the server. In addition, the multimedia content may consist of several media components (e.g. audio, video, text), each of which may have different characteristics. In MPEG-DASH, these characteristics are described by MPD which is an XML document.

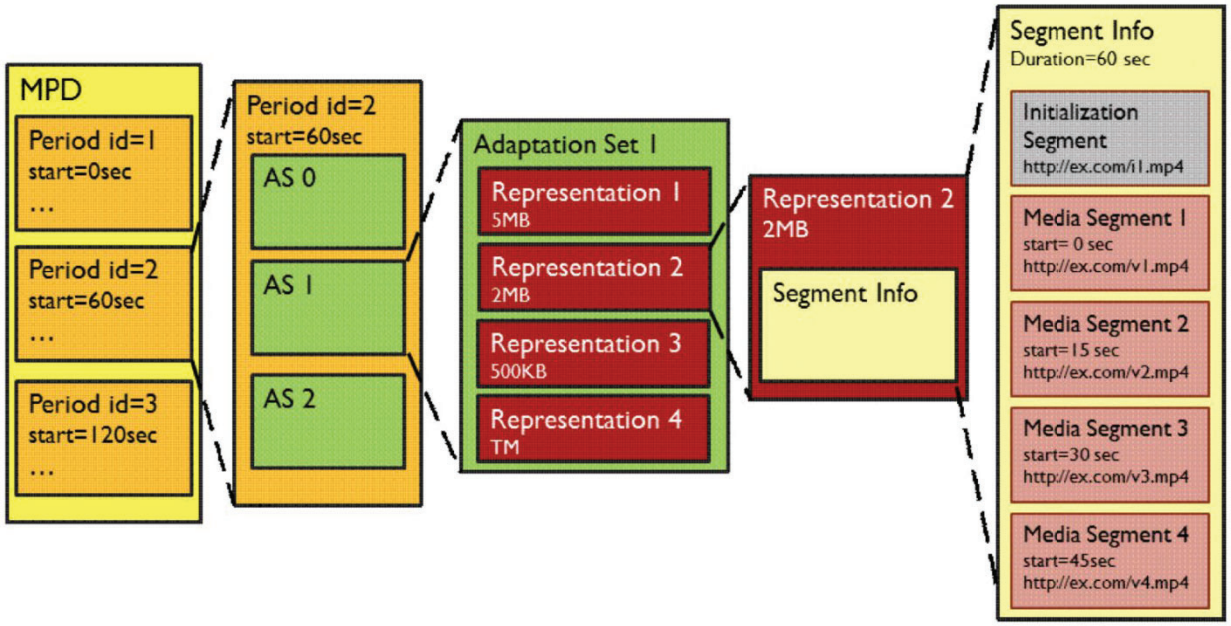


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf.

4.3 DASH data model overview

The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent or certain periods of a Media Presentation may belong to the same Asset, for example a Media Presentation is a collection of main program composed of multiple periods, each assigned to the same Asset, and interleaved with inserted advertisement periods. Each media content period is composed of one or multiple **media content components,** for example audio components in various languages, different video components providing different views of the same program, subtitles in different language, etc.. Each media content component has an assigned **media content component type**, for example audio or video.

DASH is based on a hierarchical data model aligned with the presentation in Figure 3. A DASH Media Presentation is described by a **Media Presentation Description** document. This describes the sequence of **Periods** (see 5.3.2) in time that make up the Media Presentation. A Period typically represents a media content period during which a consistent set of encoded versions of the media content is available i.e. the set of available bitrates, languages, captions, subtitles etc. does not change during a Period.

NOTE This is not strictly true, since the MPD may also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client could in principle construct a single request for multiple Segments, but this would not be the typical case.

DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components. This timeline is referred as Media Presentation timeline. The Media Segments themselves contain accurate Media Presentation timing information enabling synchronization of components and seamless switching.

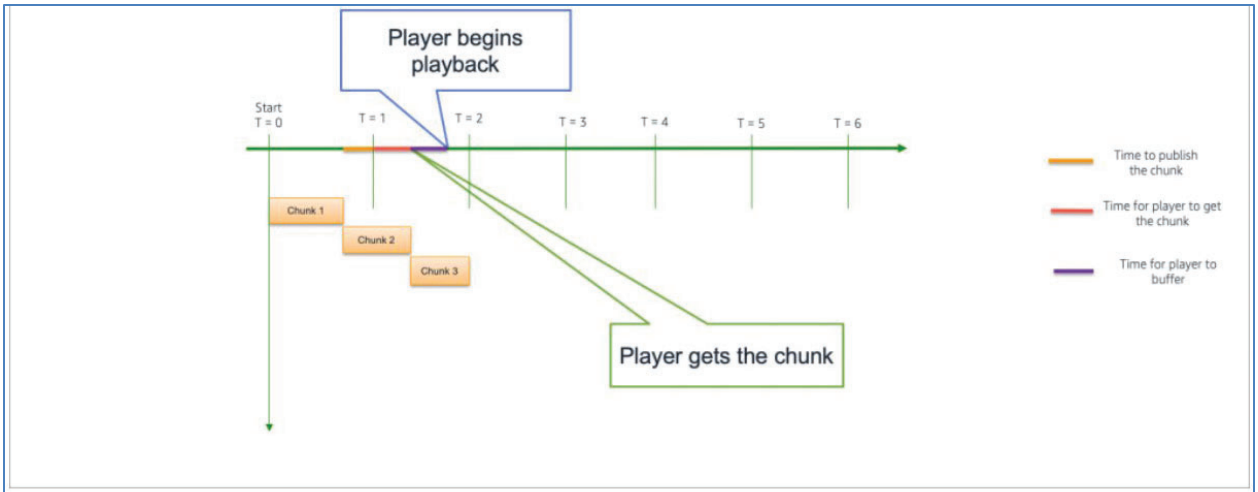
See https://github.com/liwf616/awesome-live-stream/raw/master/Ebook/ISO_IEC 23009-1_2014.pdf.

103. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising identifying a range of content surrounding the first position in the primary digital content as content to be retained (e.g., content of a next chunk overlapping a buffer time of a previous chunk, and/or the ability go forward and backward from a stop/pause/progress bar/cursor position).

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

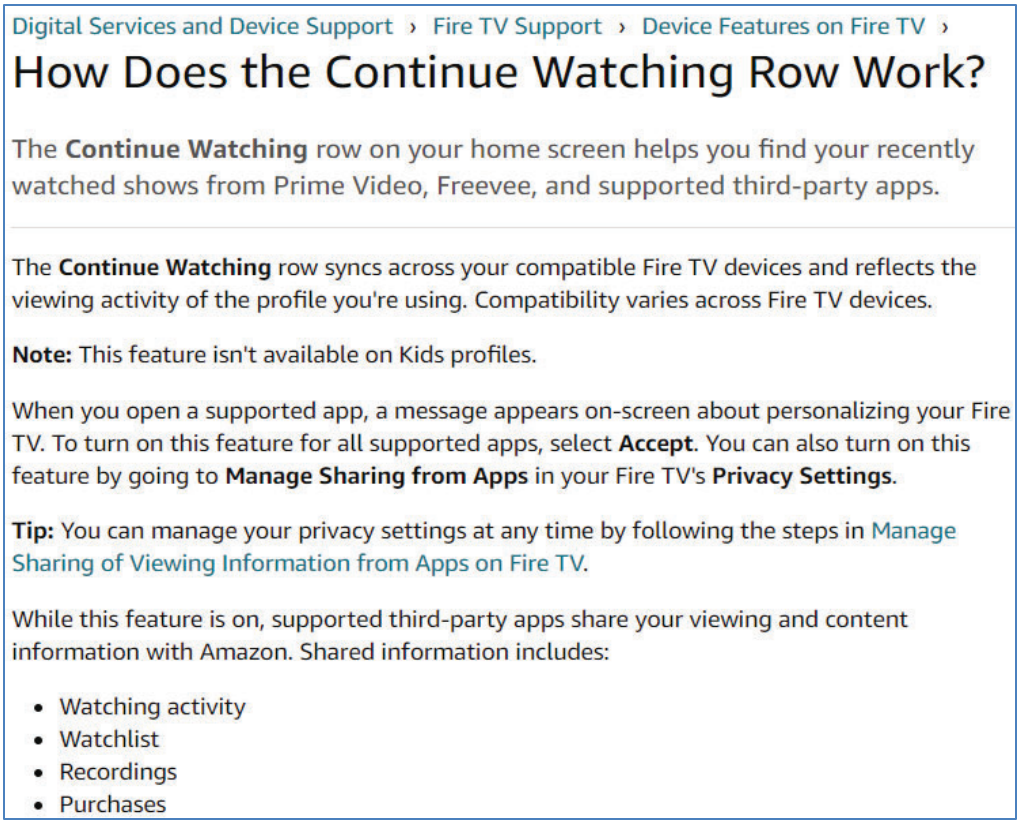
When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.



The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

104. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising releasing storage resources allocated to all content of the primary digital content that is not identified as content to be retained on the first client device (e.g., clearing App data, and/or the "storage optimizing feature").



Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

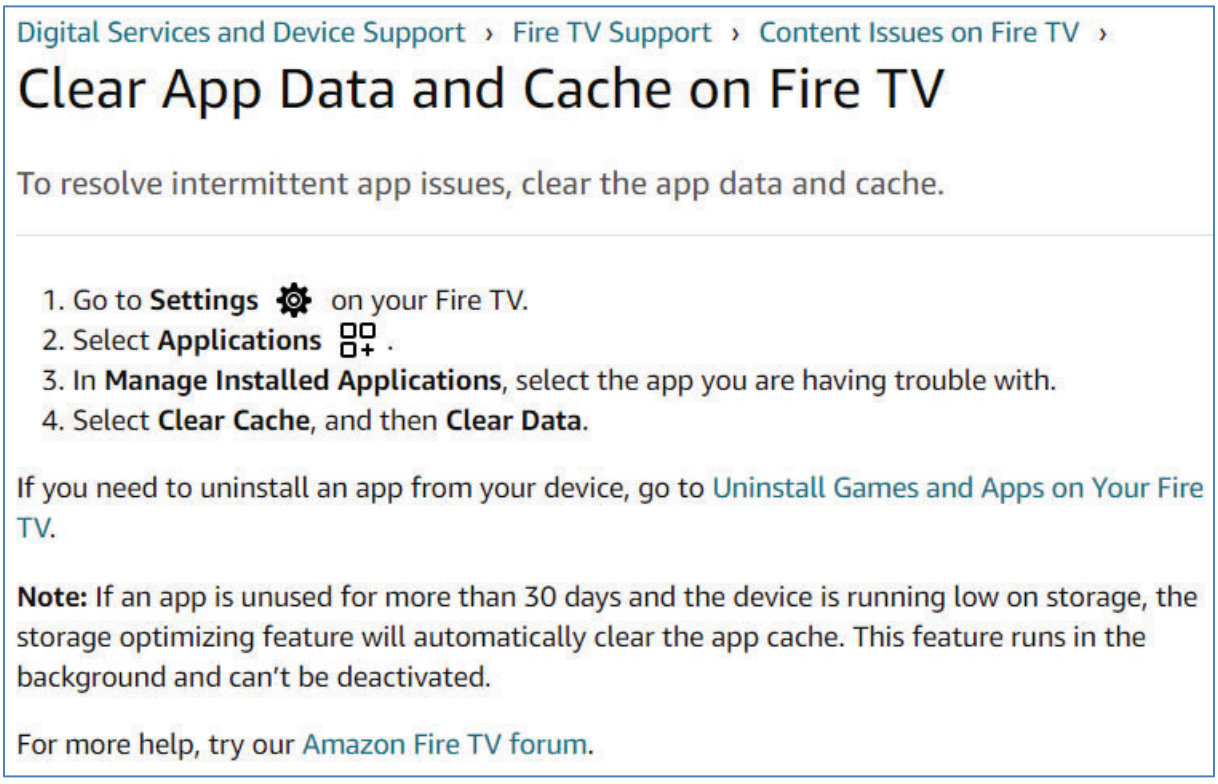
When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.



Digital Services and Device Support › Fire TV Support › Content Issues on Fire TV ›

Clear App Data and Cache on Fire TV

To resolve intermittent app issues, clear the app data and cache.

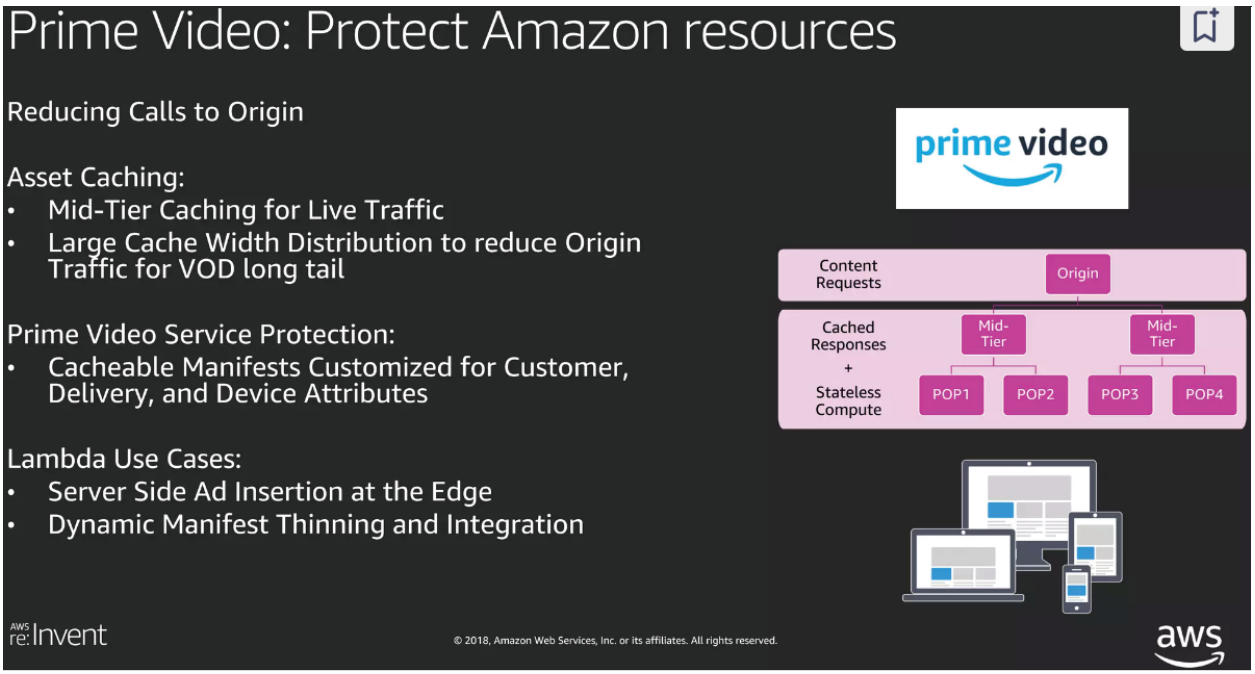
1. Go to **Settings** on your Fire TV.
2. Select **Applications** .
3. In **Manage Installed Applications**, select the app you are having trouble with.
4. Select **Clear Cache**, and then **Clear Data**.

If you need to uninstall an app from your device, go to Uninstall Games and Apps on Your Fire TV.

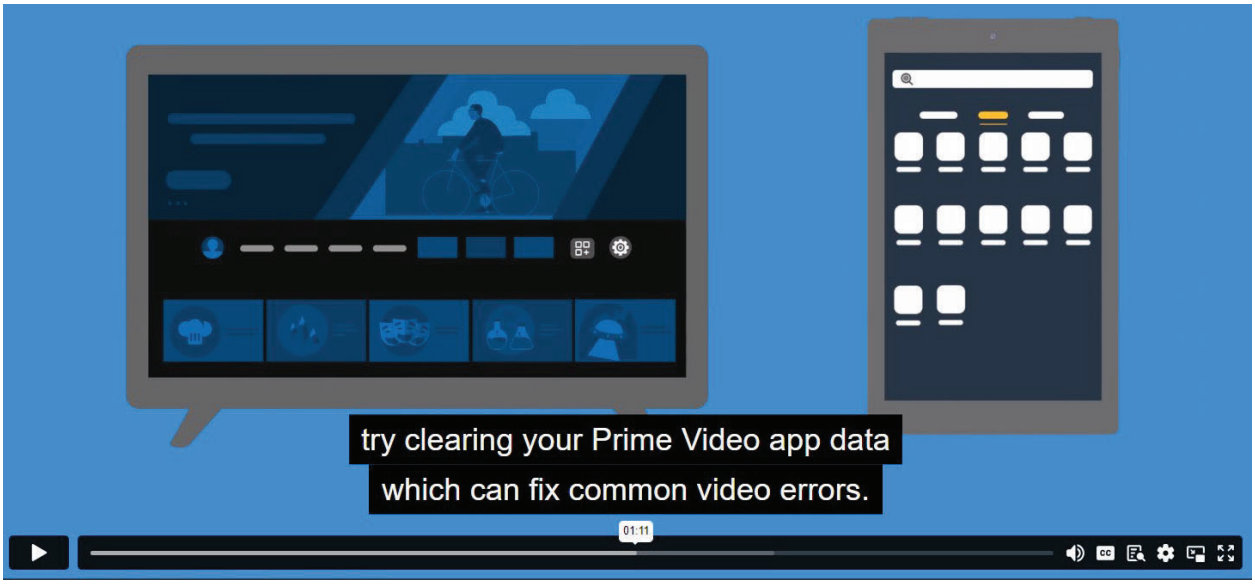
Note: If an app is unused for more than 30 days and the device is running low on storage, the storage optimizing feature will automatically clear the app cache. This feature runs in the background and can't be deactivated.

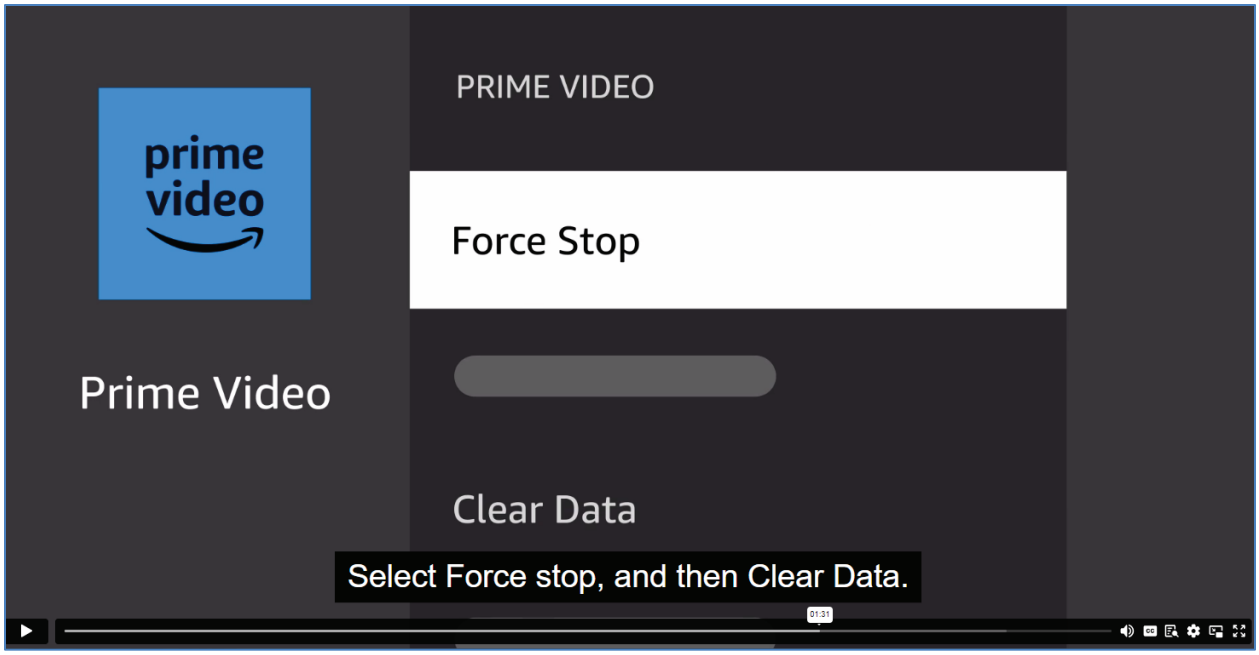
For more help, try our Amazon Fire TV forum.

See https://www.amazon.in/gp/help/customer/display.html?nodeId=GJZBJS5B8VBCGQ48.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.





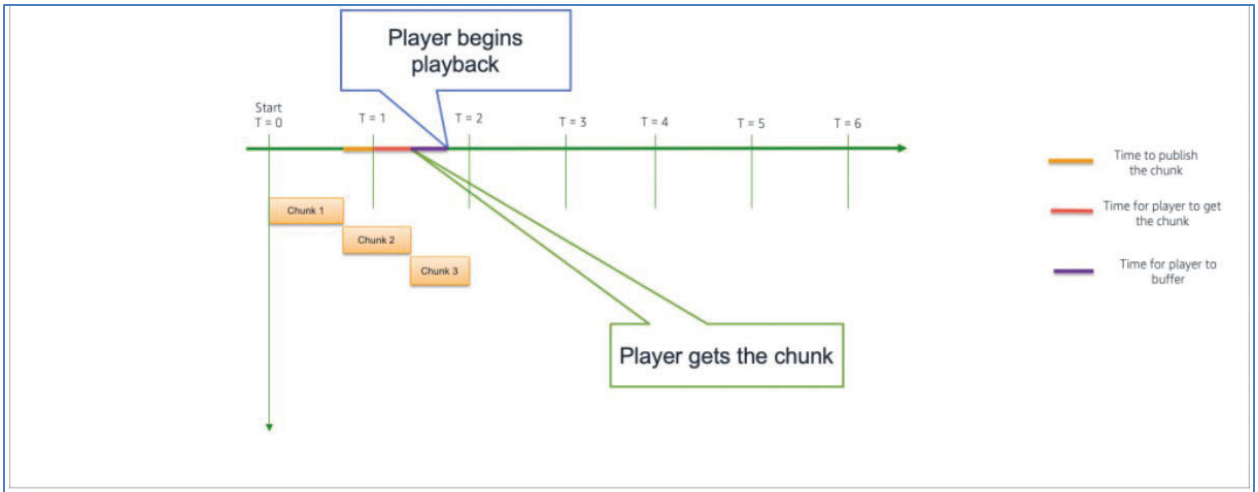
See https://www.amazon.com/gp/help/customer/display.html?ref_=hp_left_v4_sib&nodeId=G93JTA5QNSJT8WWP.

105. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising identifying content in the secondary digital content that is related to the range of content surrounding the first position in the primary digital content as content to be retained (e.g., through the "Continue Watching" and/or "Multi-Device Sync" feature).

In a previous series of posts, we detailed how to compete with broadcast latency using current adaptive bitrate (ABR) technologies. In April, AWS Elemental MediaStore announced support for workflows that use chunked object transfer to deliver objects; this means you can improve the end-to-end or "glass-to-glass" latency while also reducing the trade-offs that the current ABR technology usually demands for low-latency workflows.

This support enables ultra-low latency end-to-end workflows for over-the-top (OTT) video when using chunked CMAF or transport streams encoded for DASH or HLS distribution.

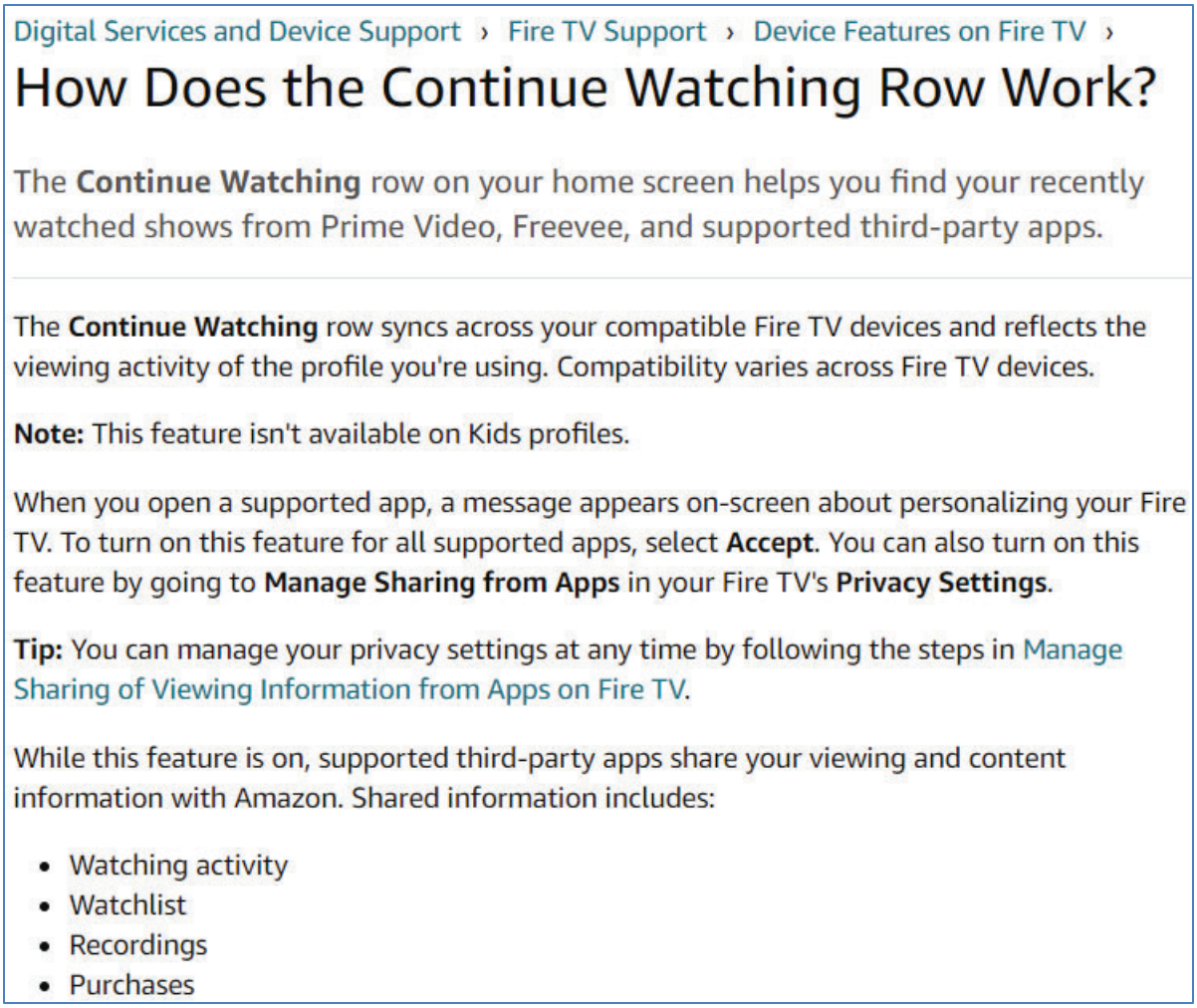
When using chunked object transfer to deliver segmented objects, your video segments are split into smaller chunks that can be played before the complete segment is delivered. Video players begin playback by requesting the video from a CDN, such as Amazon CloudFront, which in turn starts downloading the beginning of a video segment from MediaStore while the encoder is still in the process of writing the end of that same segment.



The player can then pull this chunk from CloudFront, create a small buffer to guard against network issues, and begin playback.

See https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/.

106. On information and belief, one or more components of Prime Video provide a method of rendering digital content across multiple client devices comprising the step of releasing storage resources allocated to all content of the secondary digital content that is not identified as content to be retained on the second client device (e.g., clearing App data, and/or the "storage optimizing feature").



Digital Services and Device Support › Fire TV Support › Device Features on Fire TV ›

How Does the Continue Watching Row Work?

The **Continue Watching** row on your home screen helps you find your recently watched shows from Prime Video, Freevee, and supported third-party apps.

The **Continue Watching** row syncs across your compatible Fire TV devices and reflects the viewing activity of the profile you're using. Compatibility varies across Fire TV devices.

Note: This feature isn't available on Kids profiles.

When you open a supported app, a message appears on-screen about personalizing your Fire TV. To turn on this feature for all supported apps, select **Accept**. You can also turn on this feature by going to **Manage Sharing from Apps** in your Fire TV's **Privacy Settings**.

Tip: You can manage your privacy settings at any time by following the steps in Manage Sharing of Viewing Information from Apps on Fire TV.

While this feature is on, supported third-party apps share your viewing and content information with Amazon. Shared information includes:

- Watching activity
- Watchlist
- Recordings
- Purchases

See https://www.amazon.com/gp/help/customer/display.html?nodeId=TEIq4vikEIpvh49FW1.

Digital Services and Device Support › Fire TV Support › Content Issues on Fire TV ›

Clear App Data and Cache on Fire TV

To resolve intermittent app issues, clear the app data and cache.

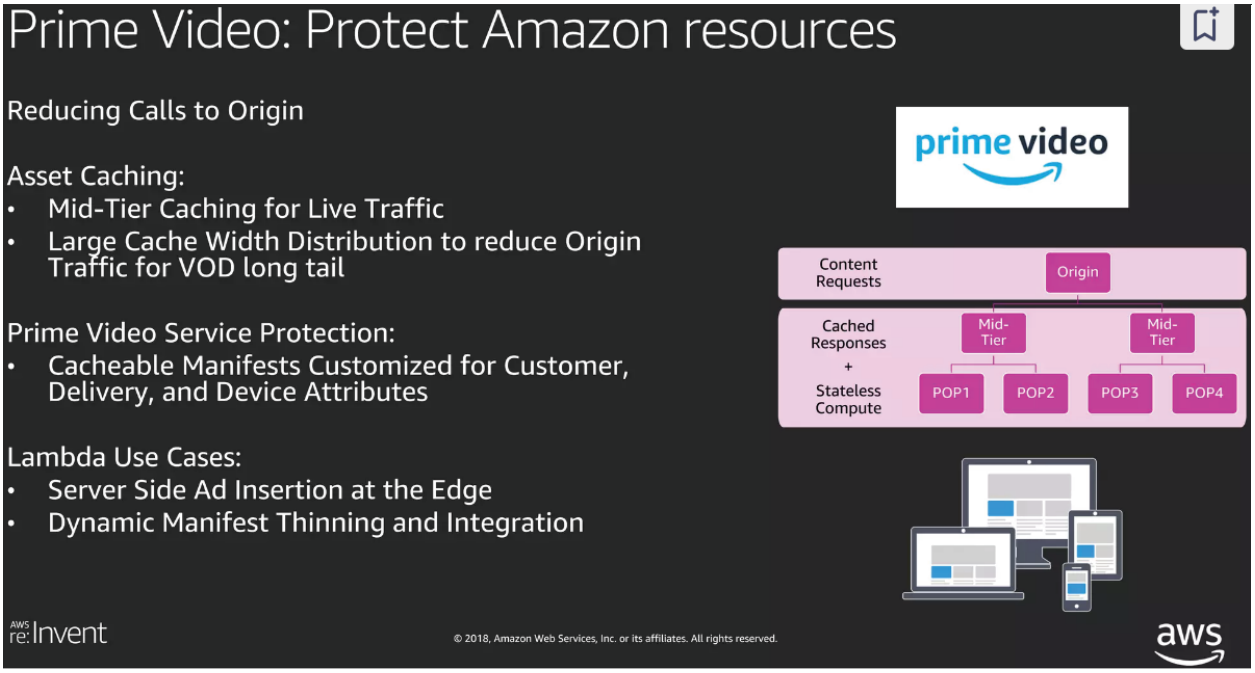
1. Go to **Settings** on your Fire TV.
2. Select **Applications** .
3. In **Manage Installed Applications**, select the app you are having trouble with.
4. Select **Clear Cache**, and then **Clear Data**.

If you need to uninstall an app from your device, go to Uninstall Games and Apps on Your Fire TV.

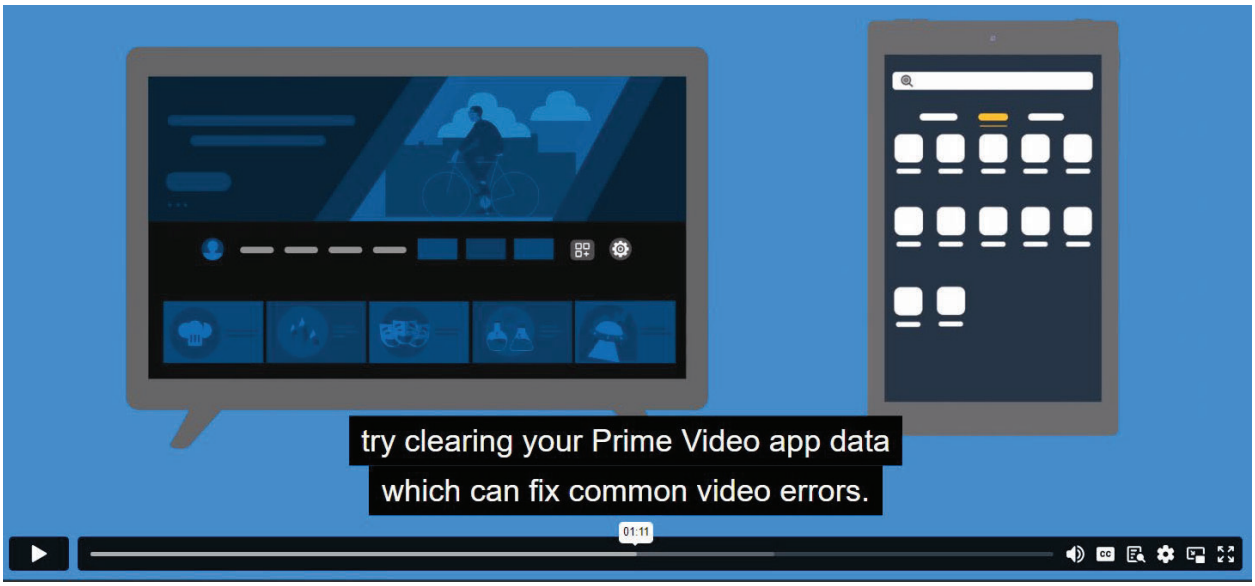
Note: If an app is unused for more than 30 days and the device is running low on storage, the storage optimizing feature will automatically clear the app cache. This feature runs in the background and can't be deactivated.

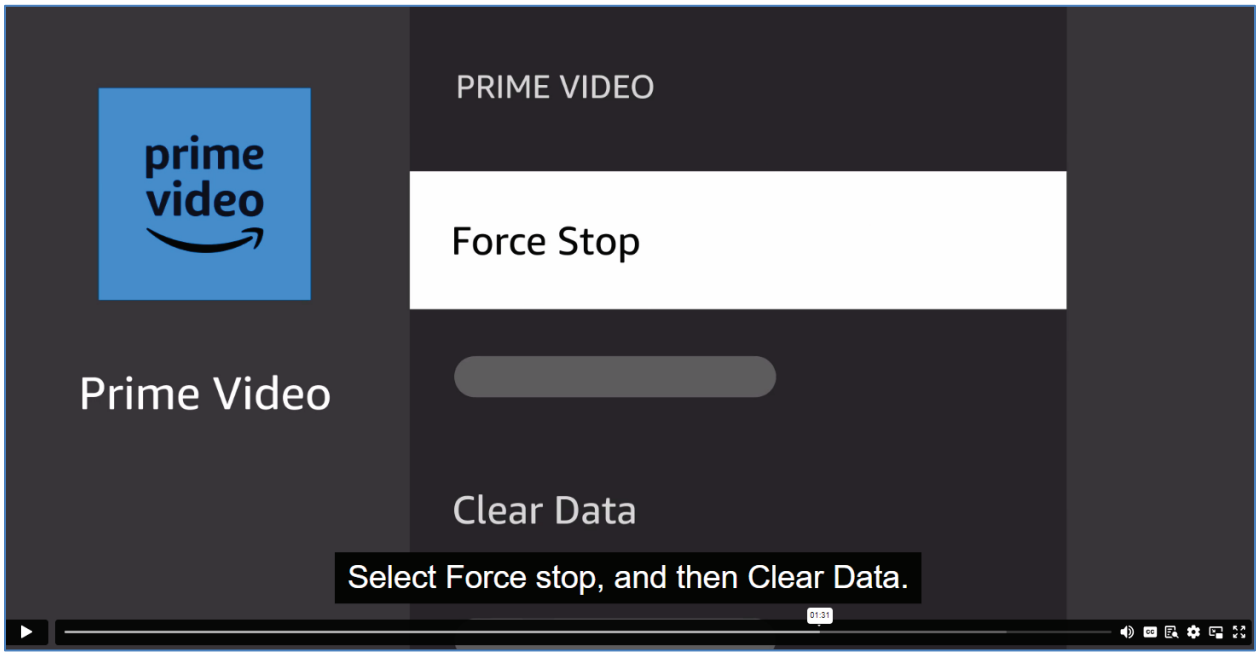
For more help, try our Amazon Fire TV forum.

See https://www.amazon.in/gp/help/customer/display.html?nodeId=GJZBJS5B8VBCGQ48.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.





See https://www.amazon.com/gp/help/customer/display.html?ref_=hp_left_v4_sib&nodeId=G93JTA5QNSJT8WWP.

107. On information and belief, Defendants directly infringe at least claim 1 of the '266 patent in violation of 35 U.S.C. § 271(a) by making, using, selling, selling access to, importing, offering for sale, and/or offering to sell access to Prime Video.

108. Defendants' infringement has damaged Audio Pod and caused/continues to cause it to suffer irreparable harm and damages.

COUNT IV - Infringement of the '488 patent by Prime Video

109. Audio Pod repeats, realleges, and incorporates by reference, as if fully set forth here, the allegations of the preceding paragraphs above.

110. On information and belief, Defendants (or those acting on their behalf) make, use, sell, sell access to, import, offer to sell and/or offer to sell access to Prime Video in the United States that infringe (literally and/or under the doctrine of equivalents) at least claim 1 of the '488 patent.

111. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered.

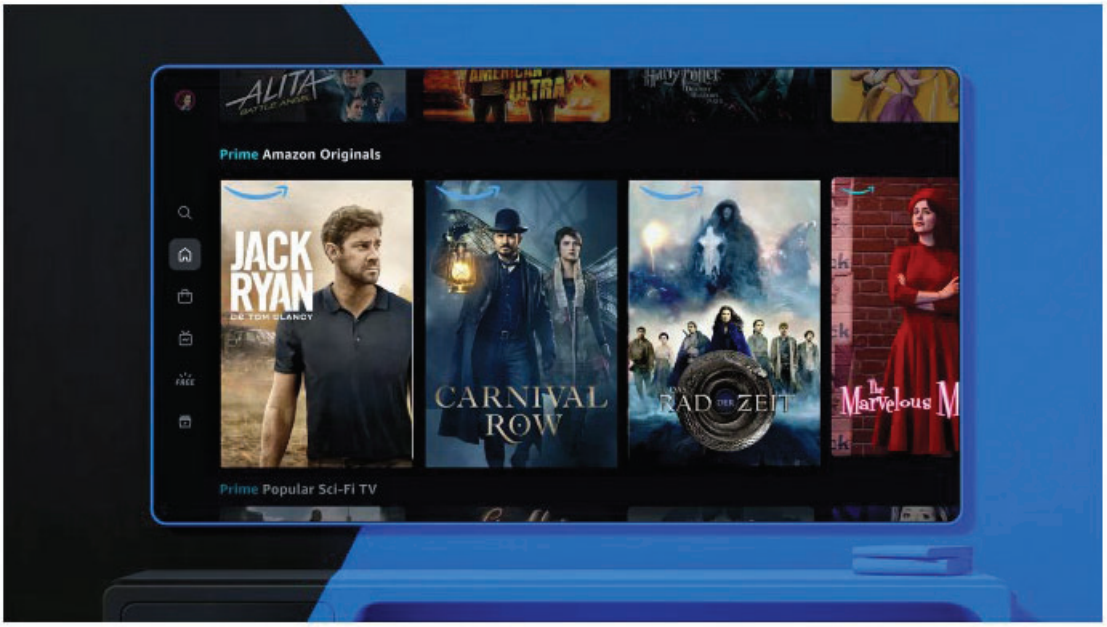
> Another case study is Amazon Prime Video who operates a video on demand service (transactional and subscription based) with a very large catalog where 12% of titles account for 90% of playbacks. For calculating performance scores of CDNs, they mostly index on the percentage of sessions without rebuffering, the percentage of sessions without fatal errors, bitrates, response times, and time to first frame. Watch this talk from re:Invent 2018, where Amazon Prime Video explains how they deliver their video using multiple CDNs, including CloudFront. You can also hear how CloudFront engineers work backwards with Amazon Prime Video to optimize performance and gain more traffic share.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

Prime Video offers a massive library of movies, series, and sports.

Read this article in Spanish.

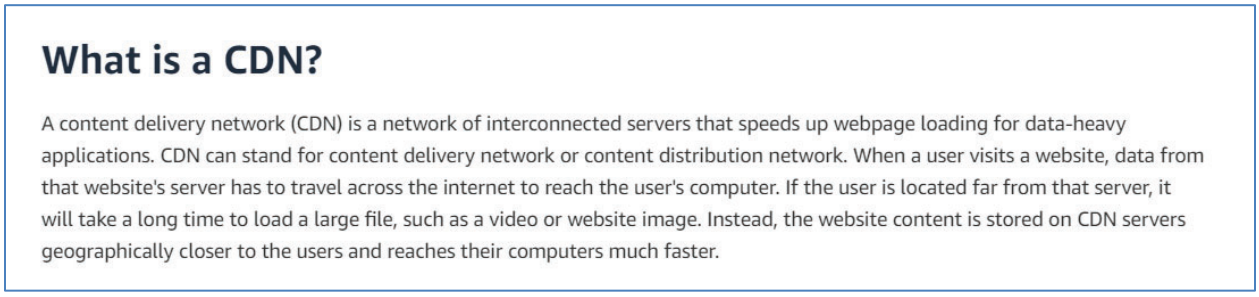
Prime Video is home to more content than you might realize. If you're ready to stream blockbuster movies, hit shows, award-winning Amazon Originals, premium channels, and sports, read on for everything you need to know about Prime Video.



See https://www.aboutamazon.com/news/entertainment/what-you-need-to-know-about-prime-video.

112. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising downloading from a network accessible server (*e.g.,* Prime Video's server) to a client device (*e.g.*, a desktop, laptop,

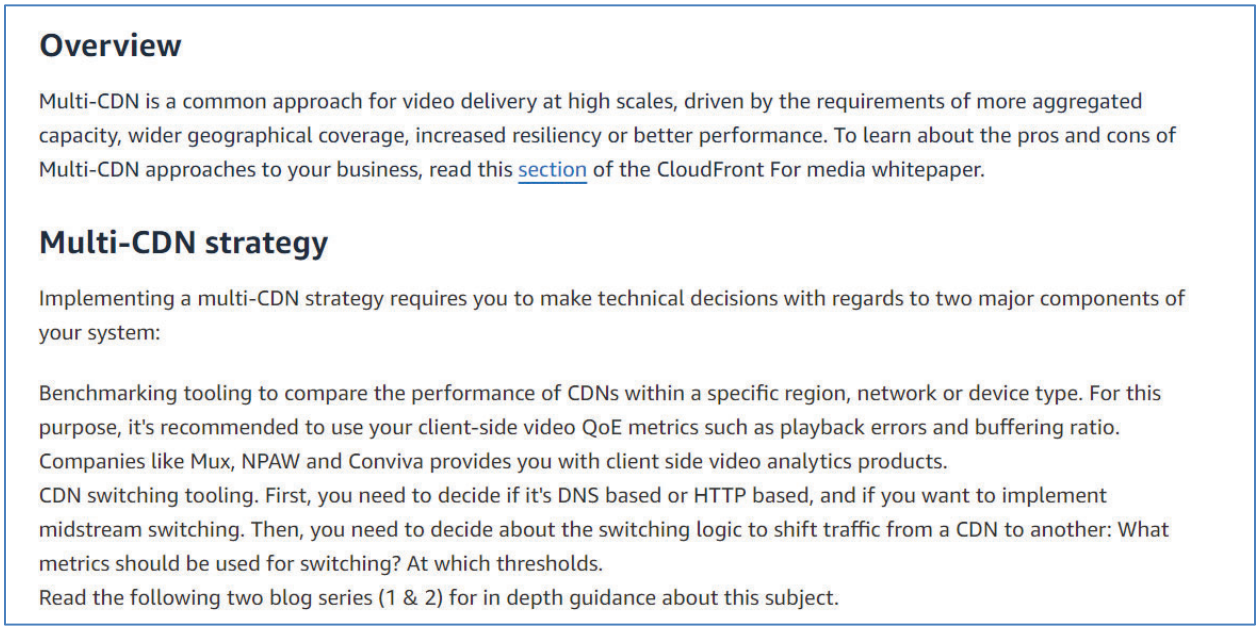
smartphone, smart-TV (e.g., Amazon Fire TV), tablets) a list of content servers (*e.g.*, Multiple CDN Servers) that are capable of serving requested digital content.



What is a CDN?

A content delivery network (CDN) is a network of interconnected servers that speeds up webpage loading for data-heavy applications. CDN can stand for content delivery network or content distribution network. When a user visits a website, data from that website's server has to travel across the internet to reach the user's computer. If the user is located far from that server, it will take a long time to load a large file, such as a video or website image. Instead, the website content is stored on CDN servers geographically closer to the users and reaches their computers much faster.

See https://aws.amazon.com/what-is/cdn/.



Overview

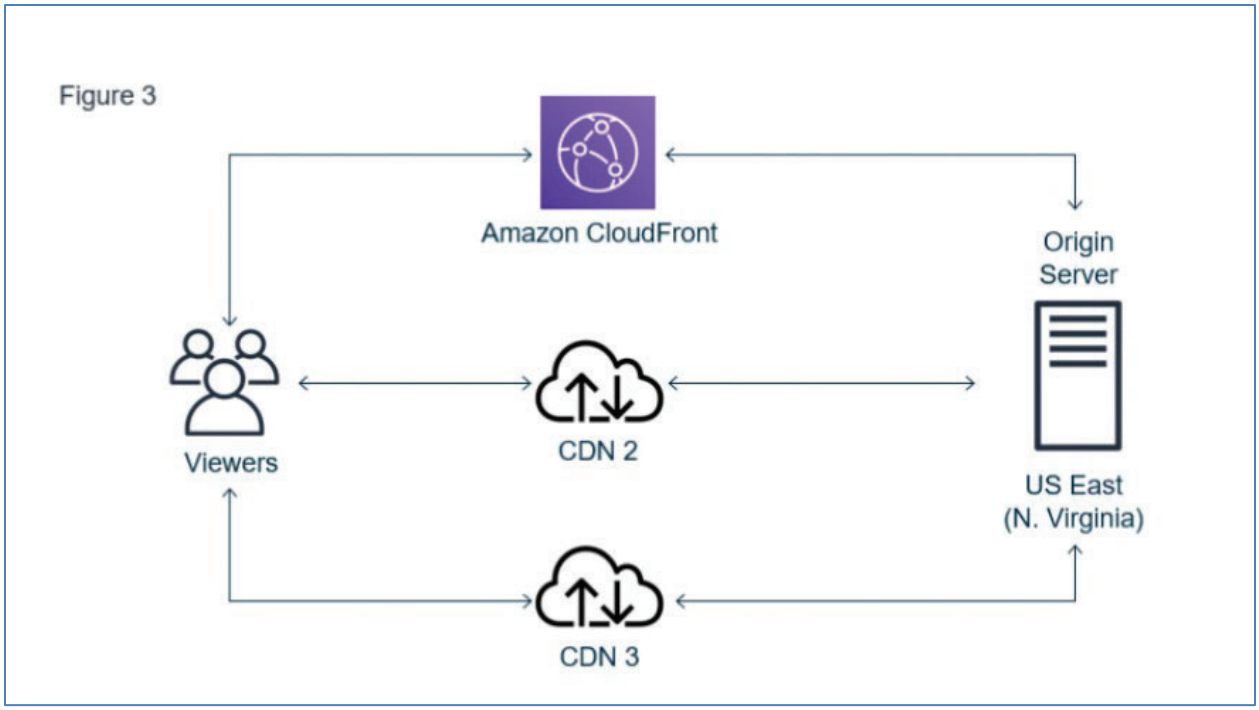
Multi-CDN is a common approach for video delivery at high scales, driven by the requirements of more aggregated capacity, wider geographical coverage, increased resiliency or better performance. To learn about the pros and cons of Multi-CDN approaches to your business, read this section of the CloudFront For media whitepaper.

Multi-CDN strategy

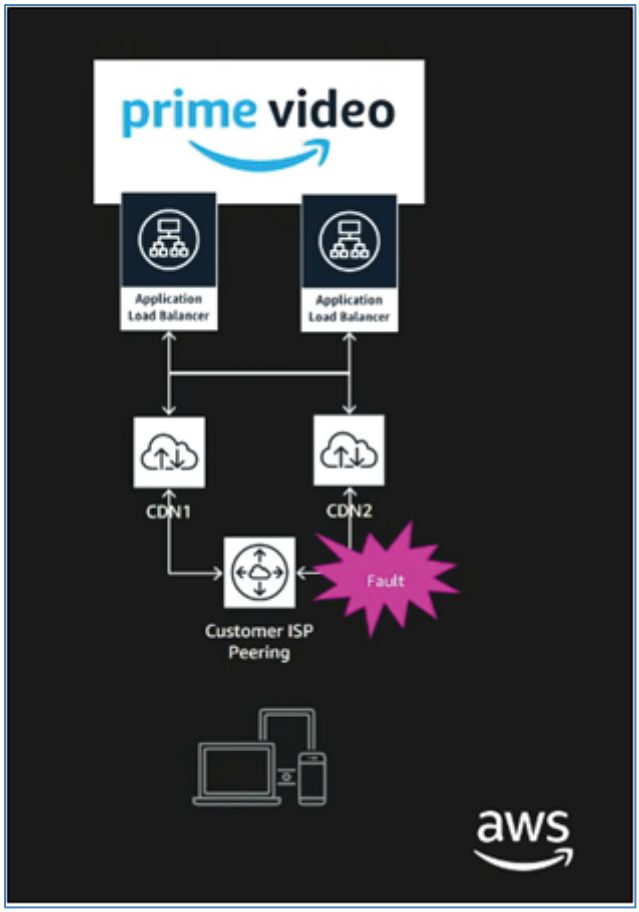
Implementing a multi-CDN strategy requires you to make technical decisions with regards to two major components of your system:

Benchmarking tooling to compare the performance of CDNs within a specific region, network or device type. For this purpose, it's recommended to use your client-side video QoE metrics such as playback errors and buffering ratio. Companies like Mux, NPAW and Conviva provides you with client side video analytics products.
CDN switching tooling. First, you need to decide if it's DNS based or HTTP based, and if you want to implement midstream switching. Then, you need to decide about the switching logic to shift traffic from a CDN to another: What metrics should be used for switching? At which thresholds.
Read the following two blog series (1 & 2) for in depth guidance about this subject.

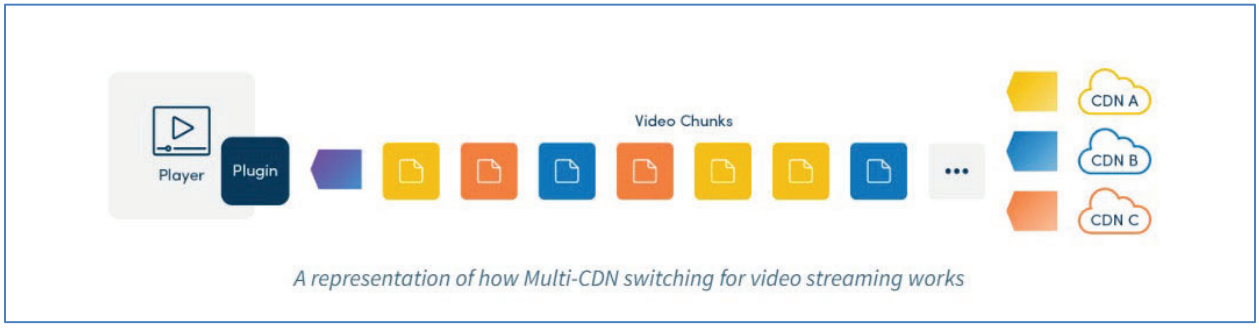
See https://aws.amazon.com/developer/application-security-performance/articles/multi-cdn-video-delivery/.



See https://aws.amazon.com/developer/application-security-performance/articles/multi-cdn-video-delivery/.



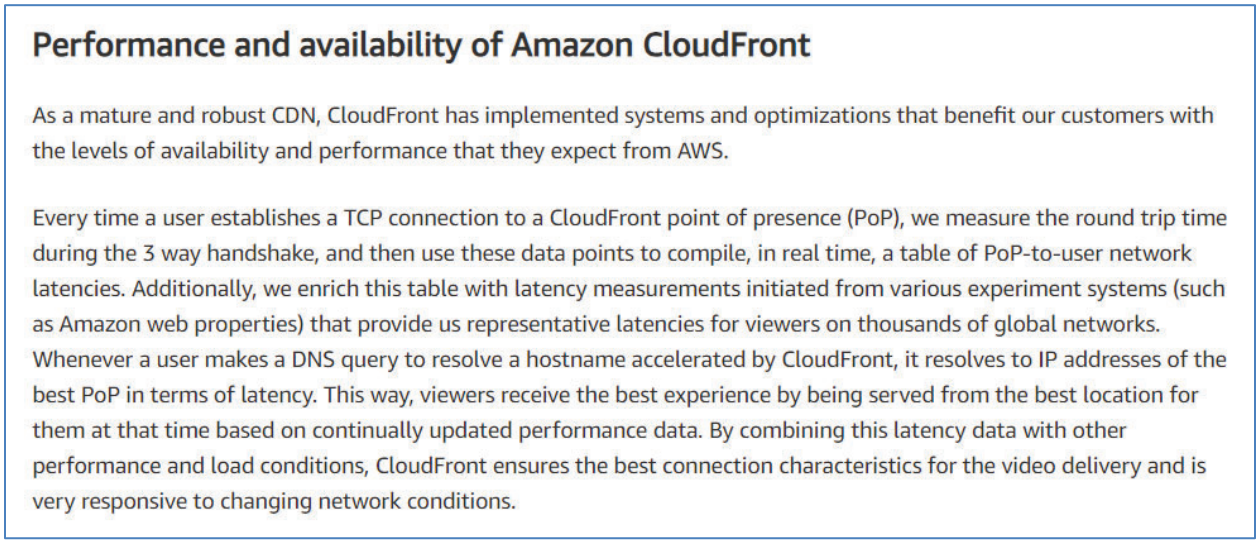
See e.g., https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.



See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

113. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising tracking service level statistics

(e.g., latency measurements, connection failures, link-level congestion) for the content servers in the list of content servers.

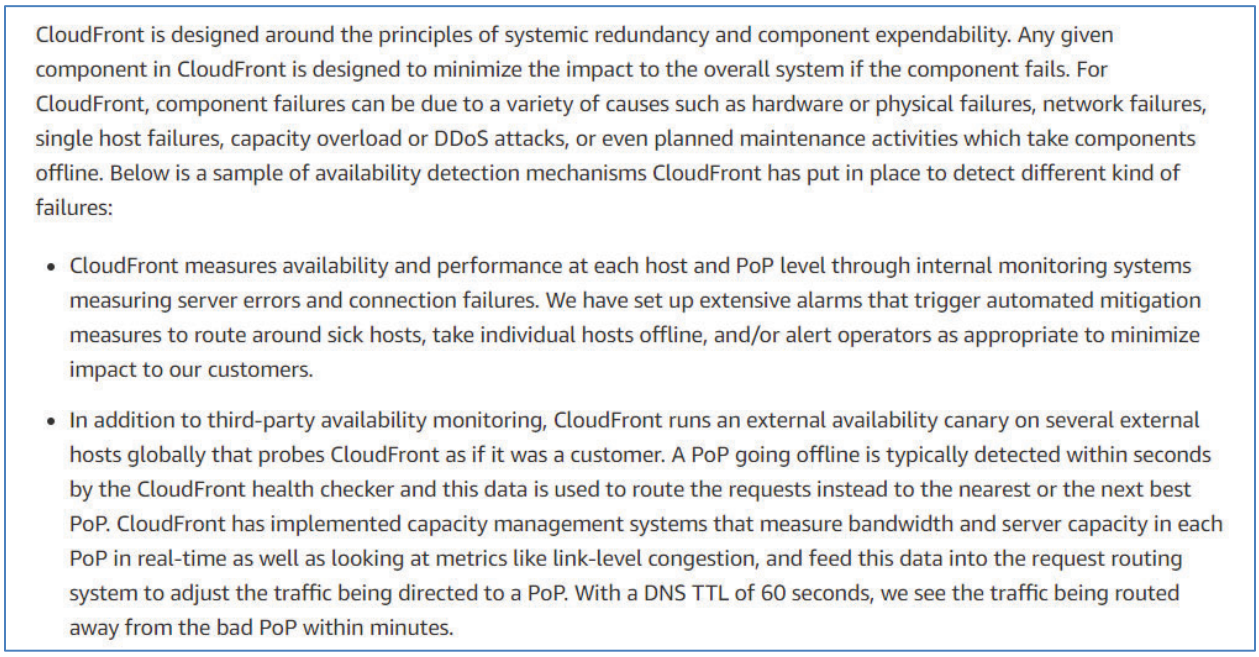


Performance and availability of Amazon CloudFront

As a mature and robust CDN, CloudFront has implemented systems and optimizations that benefit our customers with the levels of availability and performance that they expect from AWS.

Every time a user establishes a TCP connection to a CloudFront point of presence (PoP), we measure the round trip time during the 3 way handshake, and then use these data points to compile, in real time, a table of PoP-to-user network latencies. Additionally, we enrich this table with latency measurements initiated from various experiment systems (such as Amazon web properties) that provide us representative latencies for viewers on thousands of global networks. Whenever a user makes a DNS query to resolve a hostname accelerated by CloudFront, it resolves to IP addresses of the best PoP in terms of latency. This way, viewers receive the best experience by being served from the best location for them at that time based on continually updated performance data. By combining this latency data with other performance and load conditions, CloudFront ensures the best connection characteristics for the video delivery and is very responsive to changing network conditions.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-1/.



CloudFront is designed around the principles of systemic redundancy and component expendability. Any given component in CloudFront is designed to minimize the impact to the overall system if the component fails. For CloudFront, component failures can be due to a variety of causes such as hardware or physical failures, network failures, single host failures, capacity overload or DDoS attacks, or even planned maintenance activities which take components offline. Below is a sample of availability detection mechanisms CloudFront has put in place to detect different kind of failures:

- CloudFront measures availability and performance at each host and PoP level through internal monitoring systems measuring server errors and connection failures. We have set up extensive alarms that trigger automated mitigation measures to route around sick hosts, take individual hosts offline, and/or alert operators as appropriate to minimize impact to our customers.
- In addition to third-party availability monitoring, CloudFront runs an external availability canary on several external hosts globally that probes CloudFront as if it was a customer. A PoP going offline is typically detected within seconds by the CloudFront health checker and this data is used to route the requests instead to the nearest or the next best PoP. CloudFront has implemented capacity management systems that measure bandwidth and server capacity in each PoP in real-time as well as looking at metrics like link-level congestion, and feed this data into the request routing system to adjust the traffic being directed to a PoP. With a DNS TTL of 60 seconds, we see the traffic being routed away from the bad PoP within minutes.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-1/.

Measuring the performance of CDNs

Regardless of your CDN strategy, you should measure a CDN in a relevant way. This is an important step, because relying on inaccurate data leads to suboptimal decisions whether it is for benchmarking CDNs during procurement or for dynamic routing across multiple CDNs. Today, there are two common approaches for this.

The first one is consuming third party real user measurements (RUM) data like Radar community measurements provided by Cedexis. This approach has the benefit of being simple to implement, and it is quite reliable for monitoring the performance trend of a single CDN. However, it has some undesirable biases when used to compare CDNs:

- **In methodology**: It measures the performance of a CDN for hot objects with 100Kb size, while your video content is a mix of hot, warm, and cold objects with larger object sizes.
- **In configuration:** Your CDN setup might not be the same as the one used by Cedexis for measurement, for example in terms of CDN maps or optimizations.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-1/.

Calculating performance score

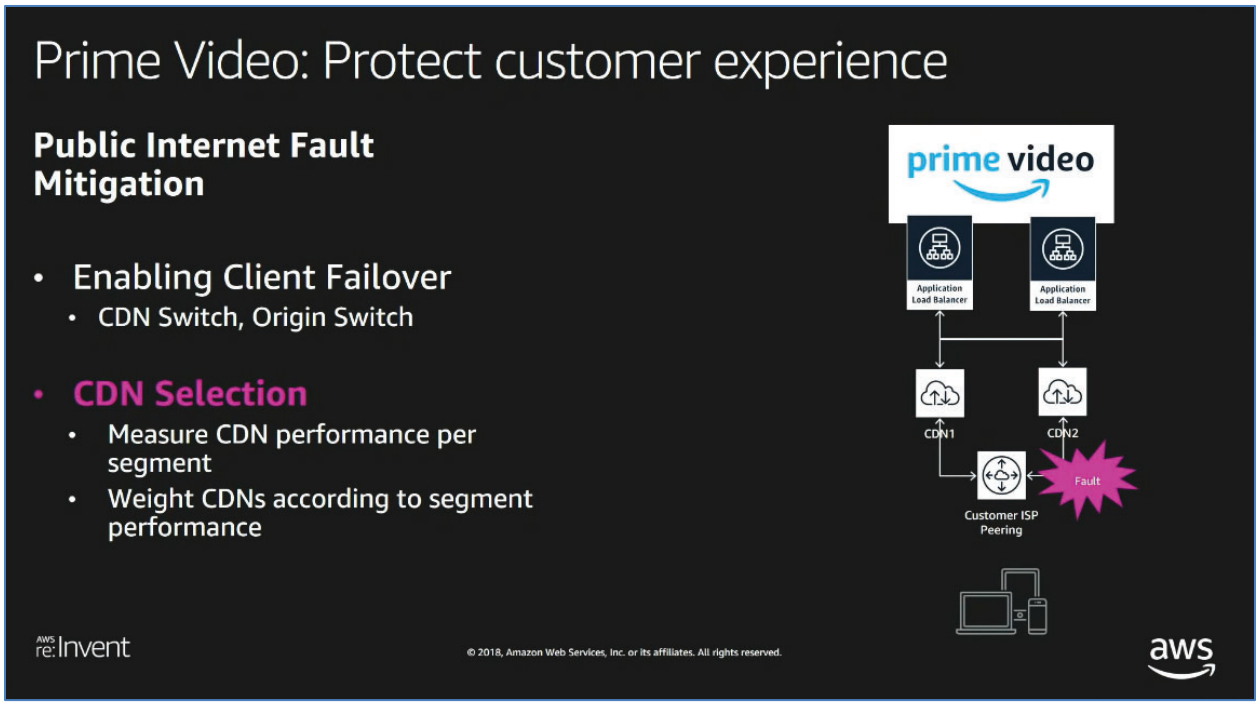
In part 1, I discussed how to measure the performance of a CDN. With measurement data available to you now, how do you combine different metrics to score CDNs and rank them?

You can calculate CDN performance scores using mathematical formulas based on commonly accepted metrics like buffering, play failures, bitrates, and startup times. **However, the optimal formula is unique according to the specifics of your business**. For example, if you deliver short video clips with ads, you would give more weight to startup time in your formula. On the other hand, for long-form video on demand (VoD) like films, you would give more weight to buffering and bitrates. As an example of scoring viewer experience, this article explains the formula used by Mux, a video technology company that provides client-side quality of experience (QoE) analytics as shown in the following screenshot. Regardless of whether you buy commercial solutions like Mux, or build your own system, make sure that you understand well the formula that is used, and then **optimize it for your workload through multiple iterations and feedback loops** based on the engagement of your audience.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

Once you decide on the relevant scoring formula for your business, you can apply it to your measurement dataset, by segments, to surface dimensions that affect the performance of video streaming. **The most important dimension is internet service provider (ISP)** which is represented by an autonomous system number (ASN). ISPs will typically have different performance characteristics because of their diverse capabilities in terms of network connectivity with various CDNs. Some customers like Amazon Prime Video go beyond a single dimension and add other dimensions like viewer location and device type for more granularity in evaluating the performance of CDNs. In fact, CDNs may optimize video delivery for some streaming formats or user devices. For example, when you use CloudFront's native smooth streaming feature to stream VoD from S3, CloudFront prefetches the next segments into the cache which improves performance for devices using this format like smart TVs. With dimensions now defined, **make sure that you have enough measurements in each segment** (ideally hundreds of thousands per day) for statistical relevance, otherwise, you can group segments. For example, in a country where you don't have enough audience to produce relevant statistics per ASN, you can group data points in one country segment.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.

114. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising selecting a first content server (e.g., a server in a first CDN) to serve the requested digital content from the list of content servers in dependence upon the service level statistics.

Allocating traffic to CDNs

At this stage, you have a performance score per segment and for each CDN. As a next step, you need to apply your traffic allocation logic to produce a routing table that will be consumed by your CDN switching system. If you do not want to build these components yourself, some commercial solutions like Cedexis or NS1 with Mux, SmartSwitch from Nice People At Work, and Conviva Precision provide them, but you can still go through the next two paragraphs to better evaluate the capabilities of your selected provider.

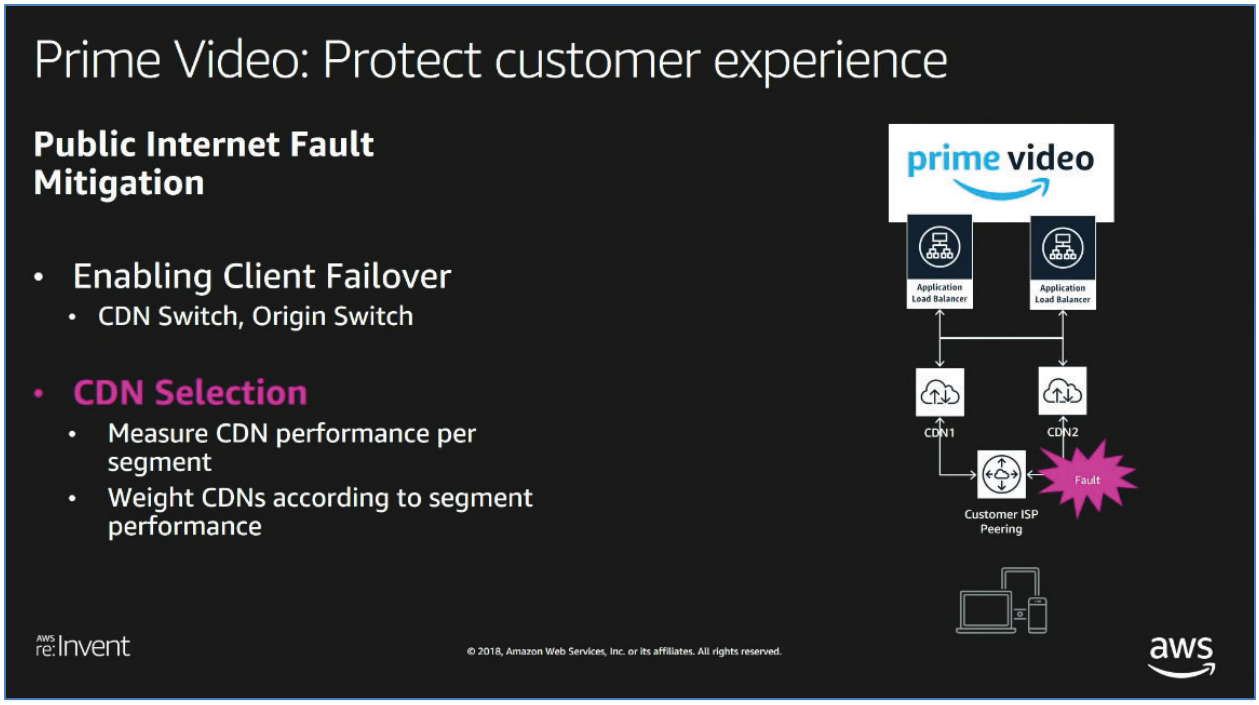
See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

That is all well and good, but how exactly is the switching between CDNs done? Multi-CDN switching is a complex process involving a load balancing algorithm that analyzes CDN performance and assigns each request to one CDN or the other based on a series of predefined rules.

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

The second approach is **HTTP based switching**. With this method, CDN switching decision are made when the video player requests the playback URL from your application. With HTTP based switching, a streaming session will stick to one CDN and benefit from reusing the same TCP/TLS connection, which makes it easier for some features like prefetching and token protection. However, you can still **failover to the second best CDN** if an issue occurs with the primary CDN allocated to a streaming session, by sending to the player a list of playback URLs using different CDNs, and ordered by their scores. You can implement failover logic on the client side by leveraging native failover mechanisms in streaming formats, such as multiple base URLs with MPEG-DASH, and redundant variant streams with HLS. For more sophisticated failover, you can implement the logic in the player code. Finally, to **improve response times of returning playback URLs**, you can use a CDN in front of your switching endpoint, and if your audience is spread across the world, you can replicate your switching endpoint to multiple geographies to route a user request to the nearest endpoint. A simple implementation using AWS services is illustrated in the following diagram.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.



See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018.

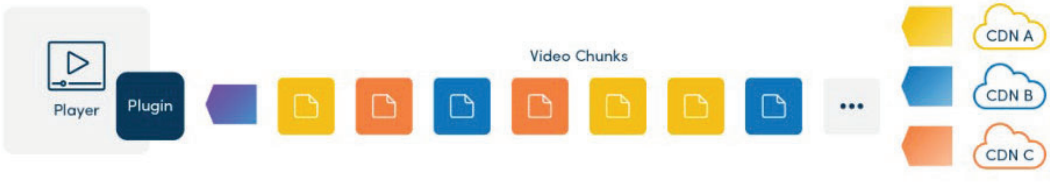
115. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising downloading a first segment (e.g., "video chunk") of the requested digital content from the first content server to the client device for rendering.

> When a user requests a specific piece of content, the request is sent directly to the CDN, which re-routes it to the closest server within the network. This process enables users to access content in near-real time, reducing latency and video playback issues like buffering or drops in video quality. That is why most streaming services today rely on a CDN to distribute their content more effectively.

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

The second approach is **HTTP based switching**. With this method, CDN switching decision are made when the video player requests the playback URL from your application. With HTTP based switching, a streaming session will stick to one CDN and benefit from reusing the same TCP/TLS connection, which makes it easier for some features like prefetching and token protection. However, you can still **failover to the second best CDN** if an issue occurs with the primary CDN allocated to a streaming session, by sending to the player a list of playback URLs using different CDNs, and ordered by their scores. You can implement failover logic on the client side by leveraging native failover mechanisms in streaming formats, such as multiple base URLs with MPEG-DASH, and redundant variant streams with HLS. For more sophisticated failover, you can implement the logic in the player code. Finally, to **improve response times of returning playback URLs**, you can use a CDN in front of your switching endpoint, and if your audience is spread across the world, you can replicate your switching endpoint to multiple geographies to route a user request to the nearest endpoint. A simple implementation using AWS services is illustrated in the following diagram.

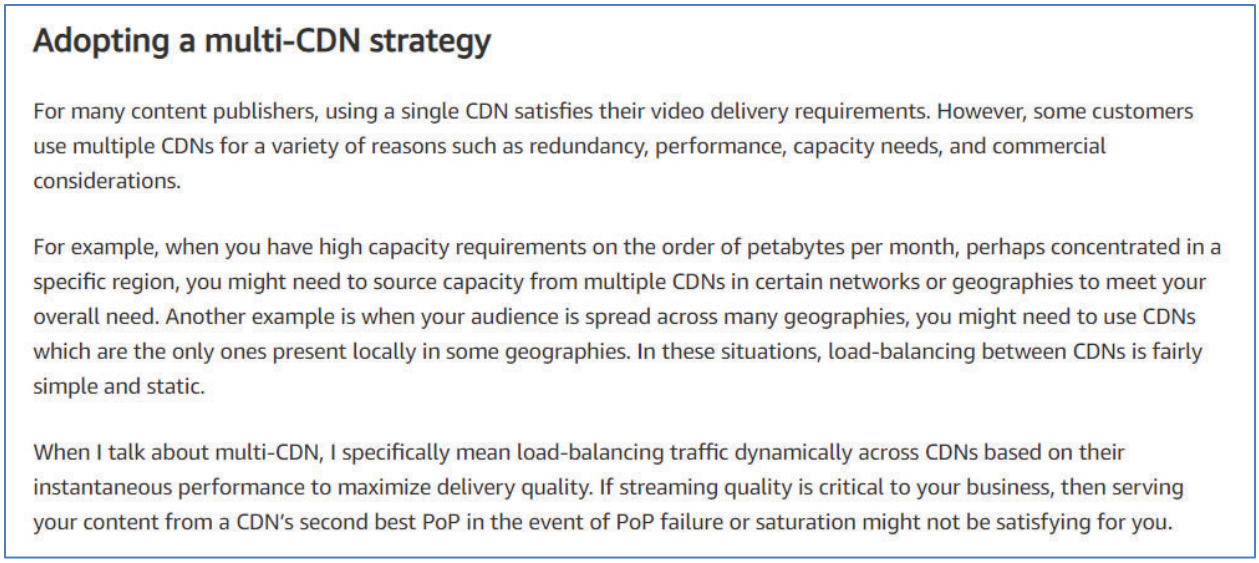
See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.



A representation of how Multi-CDN switching for video streaming works

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

116. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising in the event of a degradation in service from the first content server, selecting a second content server (e.g., a server in a second CDN) to replace the first content server in serving the requested digital content from the list of content servers in dependence upon the service level statistics, wherein the server replacement is imperceptible to a user of the client device (e.g., "uninterrupted service").



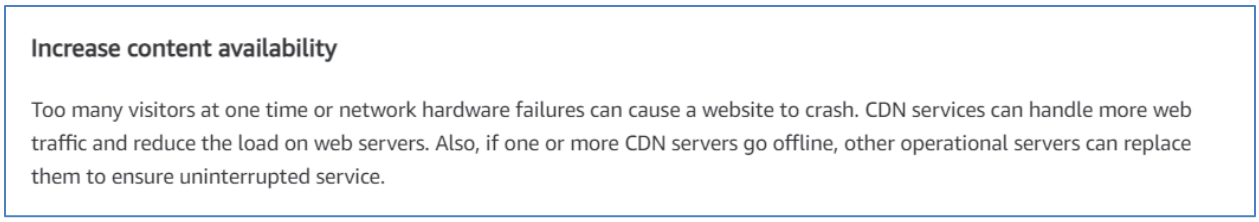
Adopting a multi-CDN strategy

For many content publishers, using a single CDN satisfies their video delivery requirements. However, some customers use multiple CDNs for a variety of reasons such as redundancy, performance, capacity needs, and commercial considerations.

For example, when you have high capacity requirements on the order of petabytes per month, perhaps concentrated in a specific region, you might need to source capacity from multiple CDNs in certain networks or geographies to meet your overall need. Another example is when your audience is spread across many geographies, you might need to use CDNs which are the only ones present locally in some geographies. In these situations, load-balancing between CDNs is fairly simple and static.

When I talk about multi-CDN, I specifically mean load-balancing traffic dynamically across CDNs based on their instantaneous performance to maximize delivery quality. If streaming quality is critical to your business, then serving your content from a CDN's second best PoP in the event of PoP failure or saturation might not be satisfying for you.

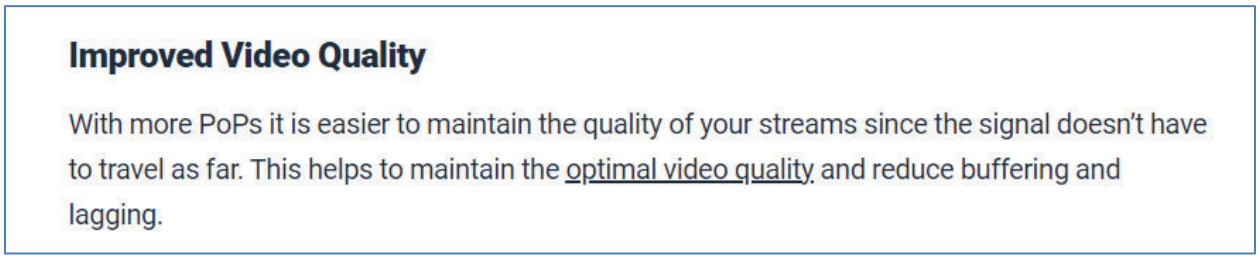
See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-1/.



Increase content availability

Too many visitors at one time or network hardware failures can cause a website to crash. CDN services can handle more web traffic and reduce the load on web servers. Also, if one or more CDN servers go offline, other operational servers can replace them to ensure uninterrupted service.

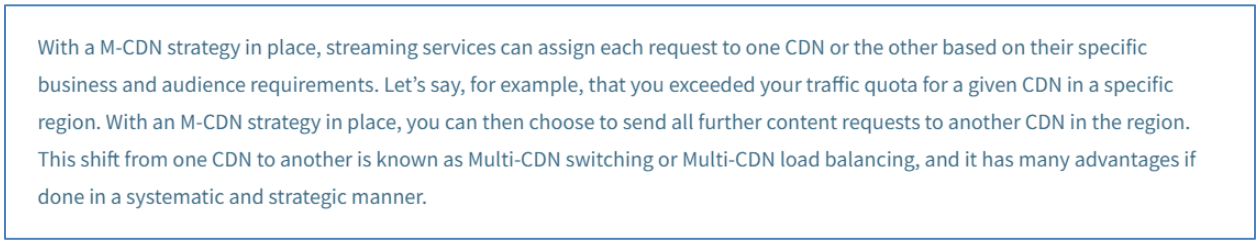
See https://aws.amazon.com/what-is/cdn/.



Improved Video Quality

With more PoPs it is easier to maintain the quality of your streams since the signal doesn't have to travel as far. This helps to maintain the optimal video quality and reduce buffering and lagging.

See https://www.dacast.com/blog/multi-cdn-solutions/.



With a M-CDN strategy in place, streaming services can assign each request to one CDN or the other based on their specific business and audience requirements. Let's say, for example, that you exceeded your traffic quota for a given CDN in a specific region. With an M-CDN strategy in place, you can then choose to send all further content requests to another CDN in the region. This shift from one CDN to another is known as Multi-CDN switching or Multi-CDN load balancing, and it has many advantages if done in a systematic and strategic manner.

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

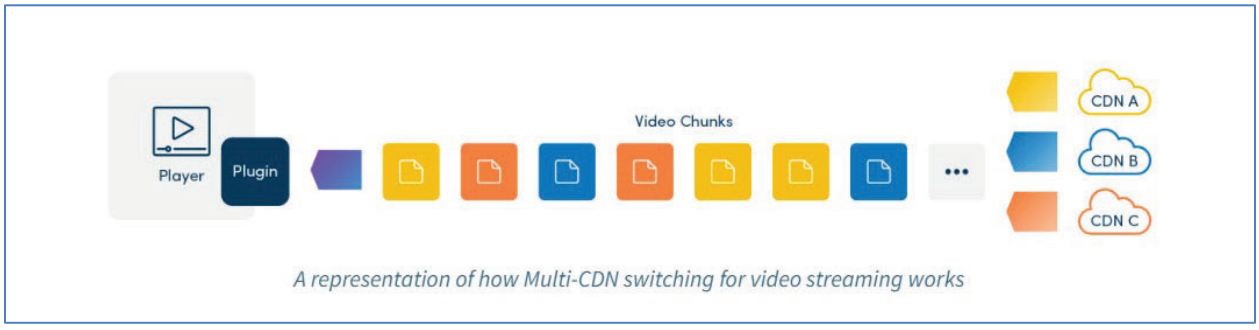
117. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising downloading a second segment (e.g., another "video chunk") of the requested digital content from the second content server to the client device for rendering.

> When a user requests a specific piece of content, the request is sent directly to the CDN, which re-routes it to the closest server within the network. This process enables users to access content in near-real time, reducing latency and video playback issues like buffering or drops in video quality. That is why most streaming services today rely on a CDN to distribute their content more effectively.

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

> The second approach is **HTTP based switching**. With this method, CDN switching decision are made when the video player requests the playback URL from your application. With HTTP based switching, a streaming session will stick to one CDN and benefit from reusing the same TCP/TLS connection, which makes it easier for some features like prefetching and token protection. However, you can still **failover to the second best CDN** if an issue occurs with the primary CDN allocated to a streaming session, by sending to the player a list of playback URLs using different CDNs, and ordered by their scores. You can implement failover logic on the client side by leveraging native failover mechanisms in streaming formats, such as multiple base URLs with MPEG-DASH, and redundant variant streams with HLS. For more sophisticated failover, you can implement the logic in the player code. Finally, to **improve response times of returning playback URLs**, you can use a CDN in front of your switching endpoint, and if your audience is spread across the world, you can replicate your switching endpoint to multiple geographies to route a user request to the nearest endpoint. A simple implementation using AWS services is illustrated in the following diagram.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

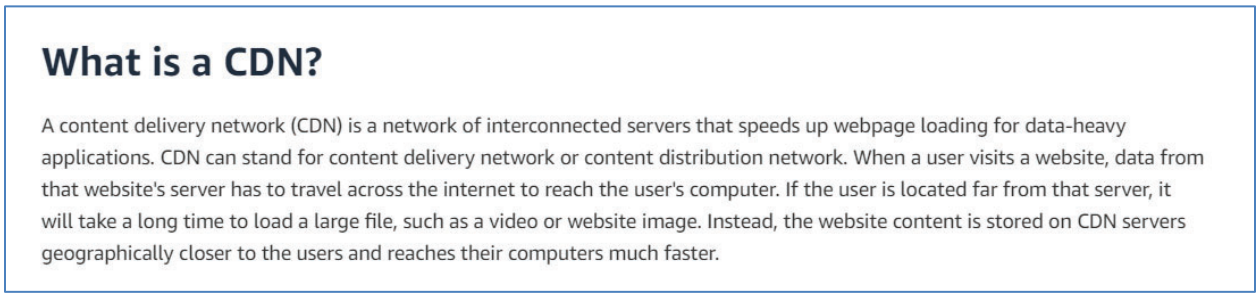


A representation of how Multi-CDN switching for video streaming works

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

118. On information and belief, one or more components of Prime Video provide a method of downloading digital content to be rendered comprising wherein the requested digital

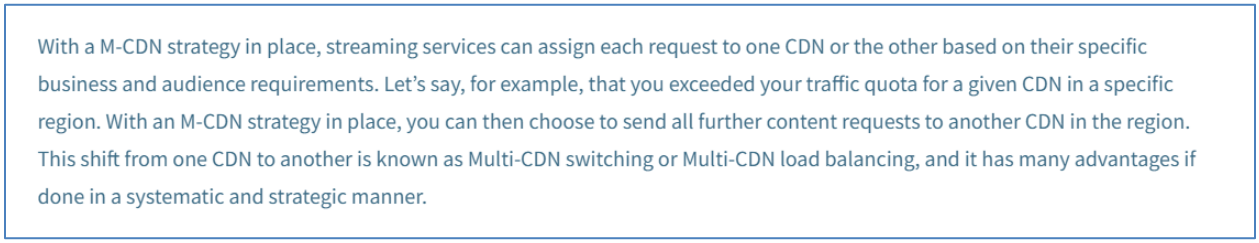
content is a digital media stream that includes the first and second segments (e.g., video chunks), wherein the first content server is a first library server (e.g., a server in a first CDN) having the digital media stream stored thereon, and wherein the second content server is a second library server (e.g., a server in a second CDN) having a copy of the digital media stream stored thereon.



What is a CDN?

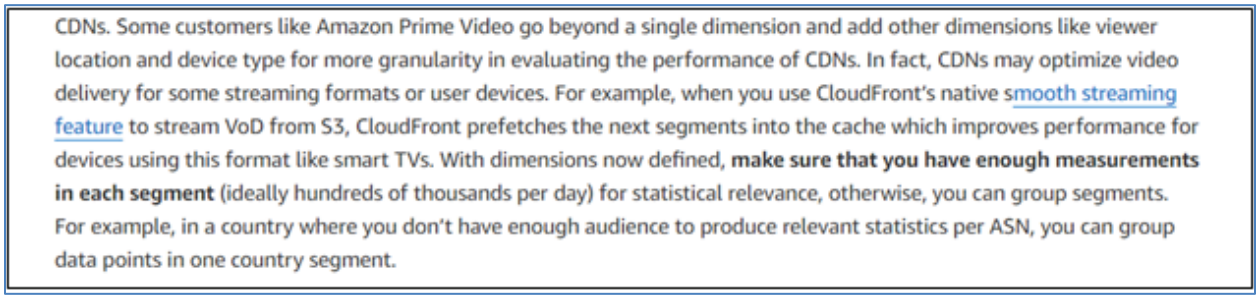
A content delivery network (CDN) is a network of interconnected servers that speeds up webpage loading for data-heavy applications. CDN can stand for content delivery network or content distribution network. When a user visits a website, data from that website's server has to travel across the internet to reach the user's computer. If the user is located far from that server, it will take a long time to load a large file, such as a video or website image. Instead, the website content is stored on CDN servers geographically closer to the users and reaches their computers much faster.

See https://aws.amazon.com/what-is/cdn/.



With a M-CDN strategy in place, streaming services can assign each request to one CDN or the other based on their specific business and audience requirements. Let's say, for example, that you exceeded your traffic quota for a given CDN in a specific region. With an M-CDN strategy in place, you can then choose to send all further content requests to another CDN in the region. This shift from one CDN to another is known as Multi-CDN switching or Multi-CDN load balancing, and it has many advantages if done in a systematic and strategic manner.

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.

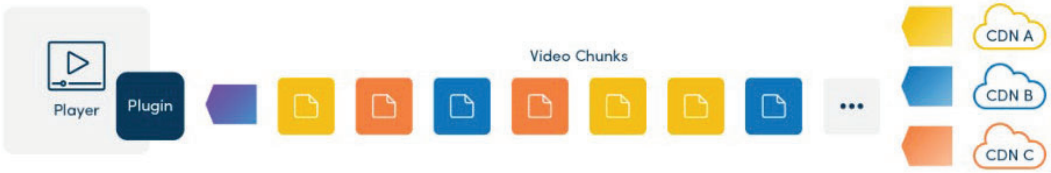


CDNs. Some customers like Amazon Prime Video go beyond a single dimension and add other dimensions like viewer location and device type for more granularity in evaluating the performance of CDNs. In fact, CDNs may optimize video delivery for some streaming formats or user devices. For example, when you use CloudFront's native smooth streaming feature to stream VoD from S3, CloudFront prefetches the next segments into the cache which improves performance for devices using this format like smart TVs. With dimensions now defined, **make sure that you have enough measurements in each segment** (ideally hundreds of thousands per day) for statistical relevance, otherwise, you can group segments. For example, in a country where you don't have enough audience to produce relevant statistics per ASN, you can group data points in one country segment.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

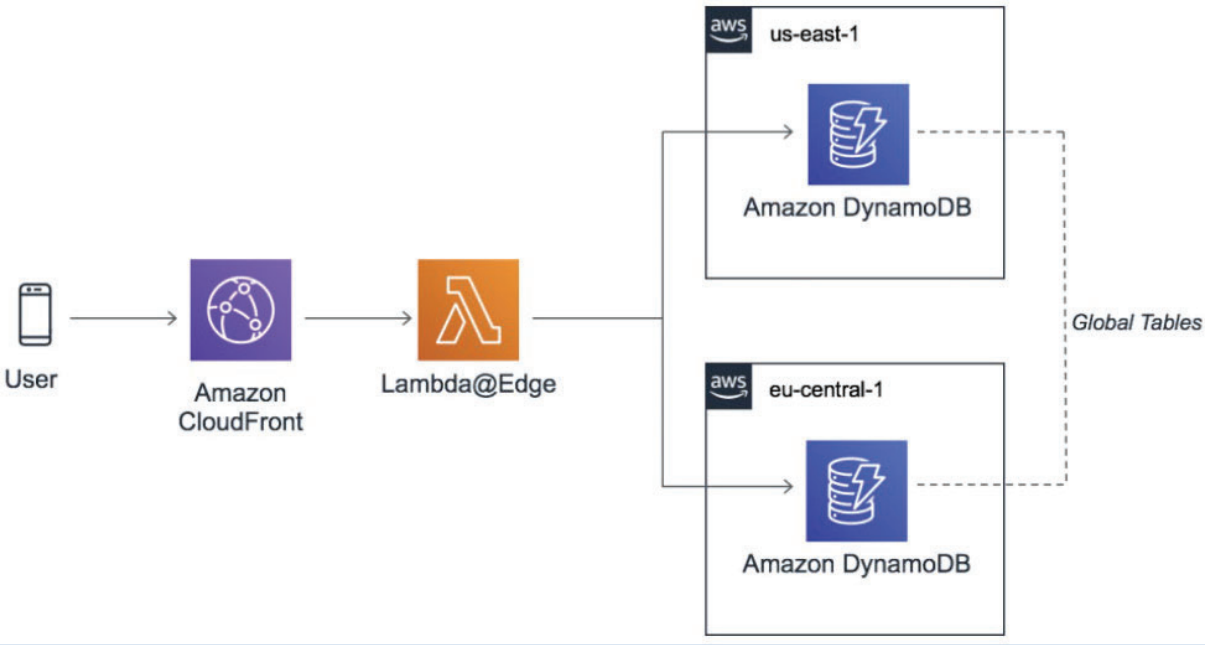
The second approach is **HTTP based switching**. With this method, CDN switching decision are made when the video player requests the playback URL from your application. With HTTP based switching, a streaming session will stick to one CDN and benefit from reusing the same TCP/TLS connection, which makes it easier for some features like prefetching and token protection. However, you can still **failover to the second best CDN** if an issue occurs with the primary CDN allocated to a streaming session, by sending to the player a list of playback URLs using different CDNs, and ordered by their scores. You can implement failover logic on the client side by leveraging native failover mechanisms in streaming formats, such as multiple base URLs with MPEG-DASH, and redundant variant streams with HLS. For more sophisticated failover, you can implement the logic in the player code. Finally, to **improve response times of returning playback URLs**, you can use a CDN in front of your switching endpoint, and if your audience is spread across the world, you can replicate your switching endpoint to multiple geographies to route a user request to the nearest endpoint. A simple implementation using AWS services is illustrated in the following diagram.

See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.



A representation of how Multi-CDN switching for video streaming works

See https://npaw.com/blog/what-is-multi-cdn-for-video-streaming/.



See https://aws.amazon.com/blogs/networking-and-content-delivery/using-multiple-content-delivery-networks-for-video-streaming-part-2/.

119. On information and belief, Defendants directly infringe at least claim 1 of the '488 patent in violation of 35 U.S.C. § 271(a) by making, using, selling, selling access to, importing, offering for sale, and/or offering to sell access to Prime Video.

120. Defendants' infringement has damaged Audio Pod and caused/continues to cause it to suffer irreparable harm and damages.

COUNT V - Infringement of the '922 patent by Prime Music

121. Audio Pod repeats, realleges, and incorporates by reference, as if fully set forth here, the allegations of the preceding paragraphs above.

122. On information and belief, Defendants (or those acting on their behalf) make, use, sell, sell access to, import, offer to sell and/or offer to sell access to Prime Music in the United States that infringes (literally and/or under the doctrine of equivalents) at least claim 1 of the '922 patent.

123. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices (e.g., desktops, laptops, smartphones, tablets, smart TVs, Echo devices, Alexa-enabled devices, Fire TV, etc.).

Amazon Music Unlimited Streaming Limits on Multiple Devices

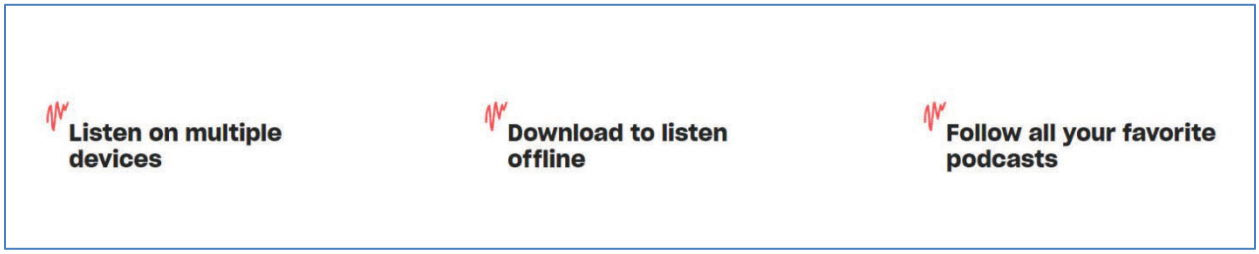
Each plan has different streaming limitations.

The Amazon Music Unlimited Family Plan allows you to stream up to six devices at the same time.

The Amazon Music Unlimited Individual Plan allows you to listen to Amazon Music Unlimited titles on all your devices. Streaming is limited to one device at a time.

The Amazon Music Unlimited Single Device Plan only allows for streaming Amazon Music Unlimited titles from the device you started your subscription on.

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GA3WMK9TWFN8PEK8.



See https://www.amazon.com/music/lp/podcasts.

What is Amazon Music?

Amazon Music is a music streaming service. It delivers a library of more than 100 million songs, as well as a huge catalog of podcasts for streaming and offline listening. Exactly what your listening experience sounds like depends on the plan you choose. Amazon offers several tiers of service, and you can pick the one that suits your needs and budget.

See https://www.aboutamazon.com/news/entertainment/what-is-amazon-music.

Feature	Amazon Music Free	Amazon Music Prime	Amazon Music Unlimited
Amazon Prime Membership Needed	No	Yes	No
Available Titles	Discover new music and podcasts based on your likes. Curated Playlists on demand for Android and iOS. Thousands of Stations and top Playlists. Millions of podcast episodes	All the music ad-free. The most ad-free top podcasts **All-Access Playlists** – pick and play any song on-demand with no skip limits, or download them to listen offline. **Note:** Except when listening to All-Access Playlists, skip limits are applicable in most other scenarios. User-created Playlists have skip limits unless they solely include "purchased songs"	Pick and play any song, ad-free. The most ad-free top podcasts Personalized Stations and thousands of Playlists SD, HD, Ultra HD, and Spatial Audio Listen offline Unlimited skips **Note:** Amazon Music Unlimited Single Device Plan doesn't have access to HD, Ultra HD, or Spatial Audio.

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GW3PHAUCZM8L7W9L.

Supported Devices	Supported Echo devices. A growing list of Alexa-enabled devices. iOS Android Fire TV devices Fire Tablet Amazon Music for Web	All Amazon Music supported devices. **Note:** All-Access Playlists requested on Echo devices play in shuffle mode. On Fire TV, you can only listen to music. There are no podcasts available. On Fire Tablet, you don't have a full music catalog or ad-free podcasts availability.	All Amazon Music supported devices (HD, Ultra HD, and Spatial Audio are available on select devices). **Amazon Music Unlimited Single Device Plan**: Echo devices Fire TV devices

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GW3PHAUCZM8L7W9L.

124. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising downloading first digital content corresponding to a media work (*e.g.*, Prime Music uses MPEG-DASH for adaptive streaming, which can include, e.g., audio streams, and subtitle streams corresponding to a media work (e.g., an audio work (e.g., a song, or podcast))) from a network accessible library (e.g., on an "Amazon server") to a first client device (e.g., a desktop, laptop, smartphone, tablet, smart TV, Echo device, Alexa-enabled device, Fire TV) via a network, the first digital content including at least a first portion (e.g., "chunk") of a first media stream.

What is Amazon Music?

Amazon Music is a music streaming service. It delivers a library of more than 100 million songs, as well as a huge catalog of podcasts for streaming and offline listening. Exactly what your listening experience sounds like depends on the plan you choose. Amazon offers several tiers of service, and you can pick the one that suits your needs and budget.

See https://www.aboutamazon.com/news/entertainment/what-is-amazon-music.

Streaming services, such as Amazon Music, have revolutionized the way we listen to music. Unlike traditional methods of purchasing and downloading individual songs or albums, streaming platforms allow users to access an extensive catalog of music for a monthly subscription fee or through free ad-supported options. This vast library of songs is stored on remote servers, and users can stream them directly to their devices over the internet.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

Streaming services like Amazon Music rely on internet connectivity to deliver music to your devices. Each time you play a song, it is streamed from Amazon's servers to your device in real-time. This constant streaming requires a significant amount of data, which can quickly add up, especially for those with limited data plans or slower internet connections.

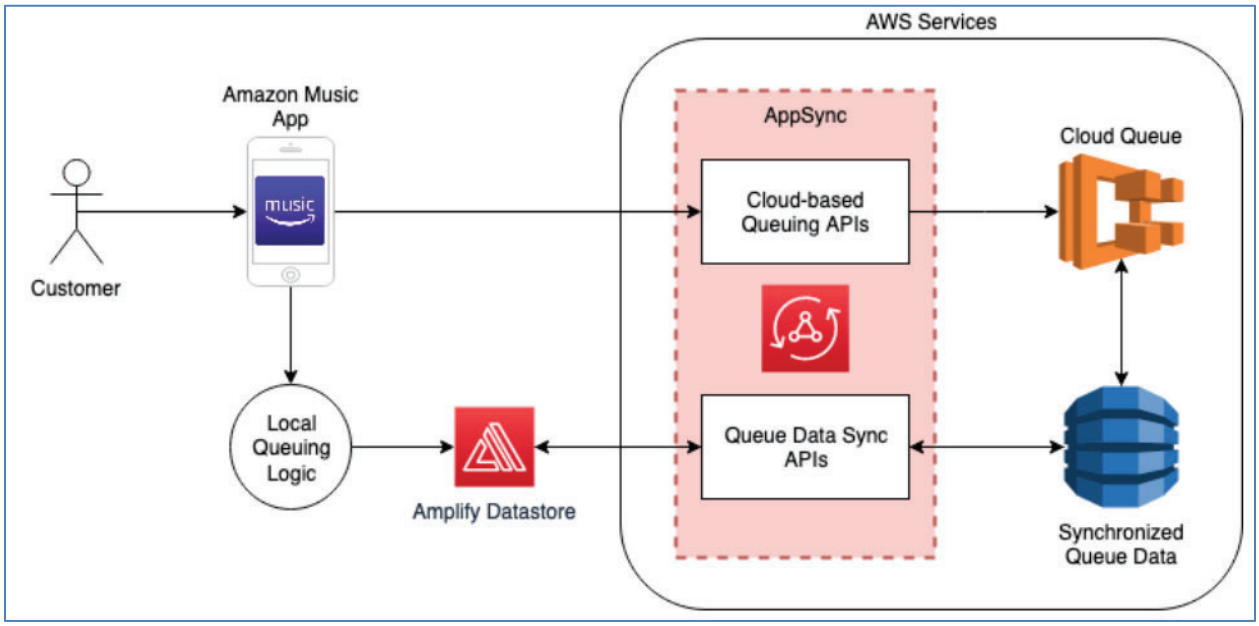
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

To stream music without issues, you need a strong Internet connection. To troubleshoot streaming issues:

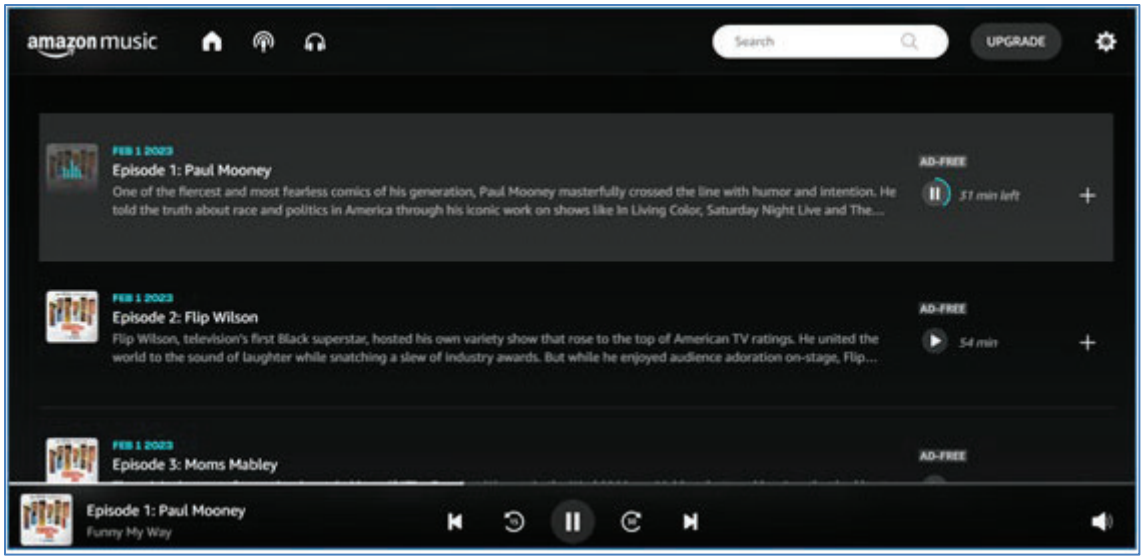
- Confirm that your device is connected to Wi-Fi or a mobile network.
- If using a mobile network, confirm that the Amazon Music app settings allow for Cellular.
- Force stop and reopen the app.

To stream HD and Ultra HD music with Amazon Music Unlimited, you need a strong Internet connection. To troubleshoot streaming issues:

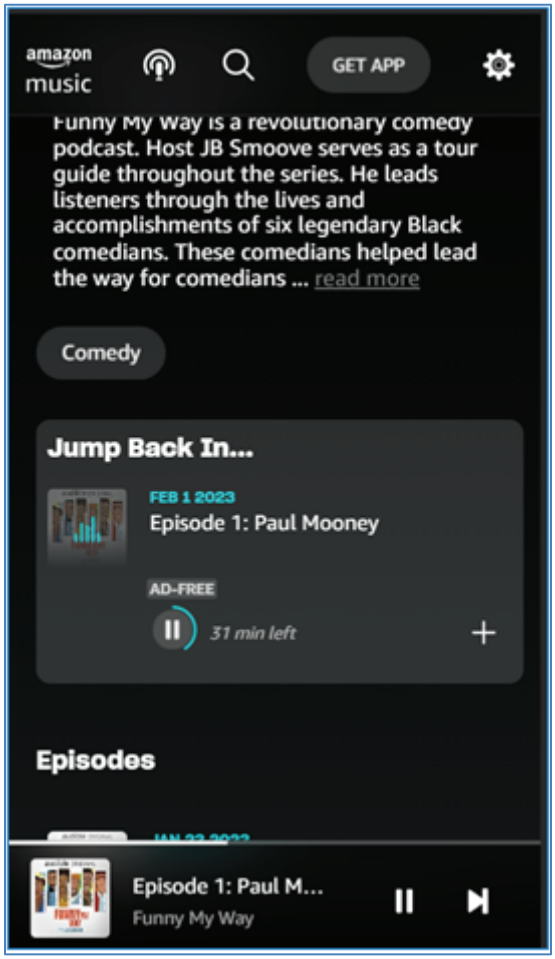
See https://www.amazon.com/gp/help/customer/display.html?nodeId=GPTQFUDBXQY85W6M.



See https://aws.amazon.com/blogs/mobile/amazon-music-unifies-music-queuing-at-scale-using-aws-appsync-and-aws-amplify/.



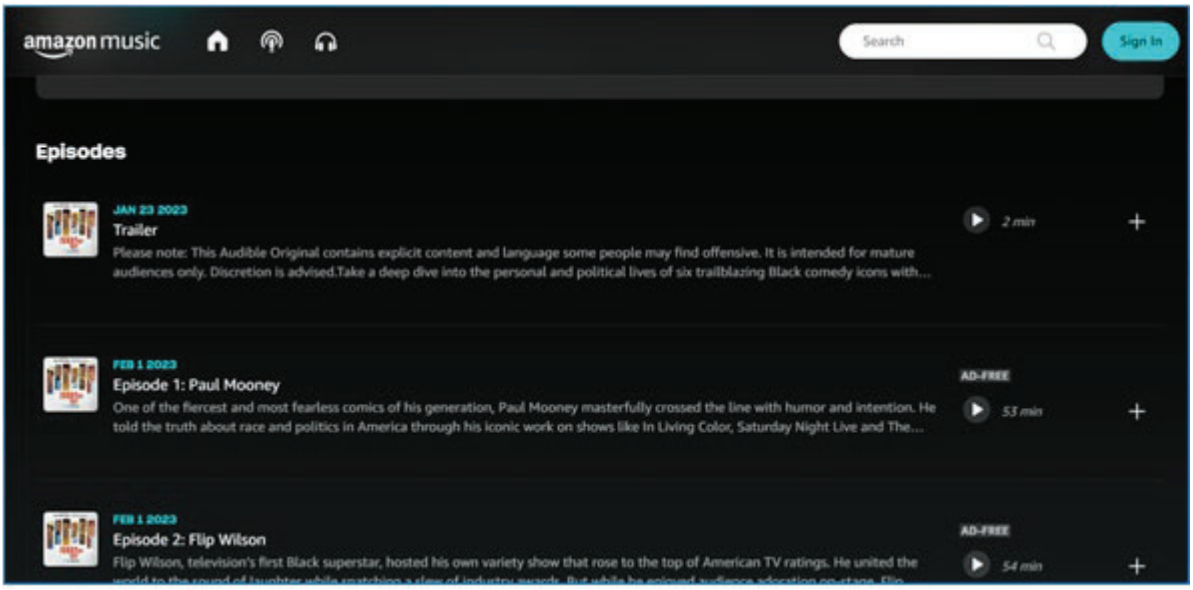
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



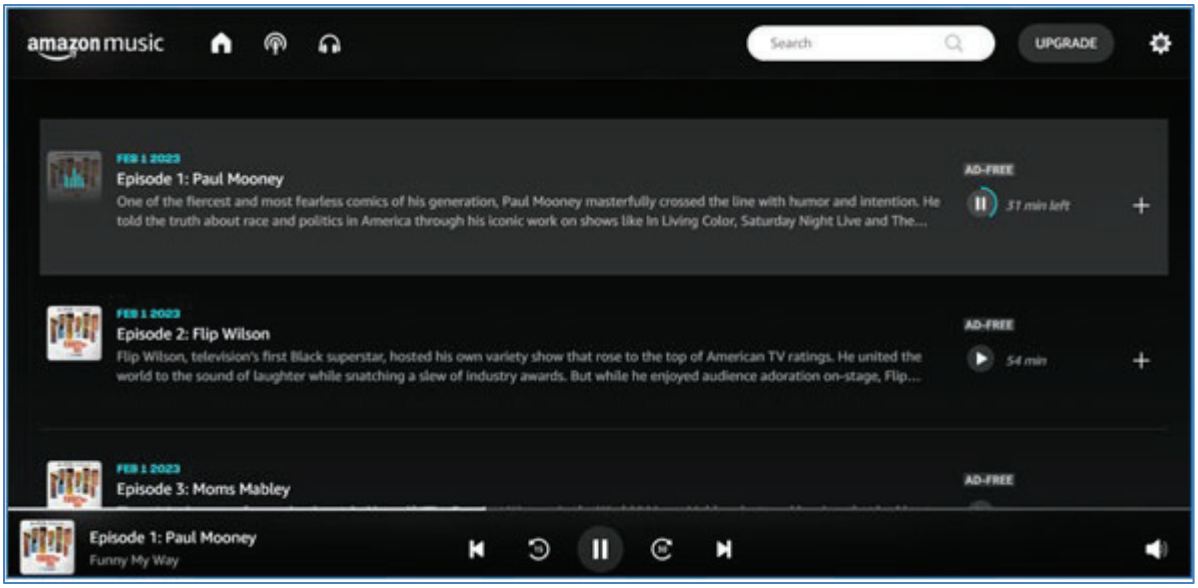
See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

> In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

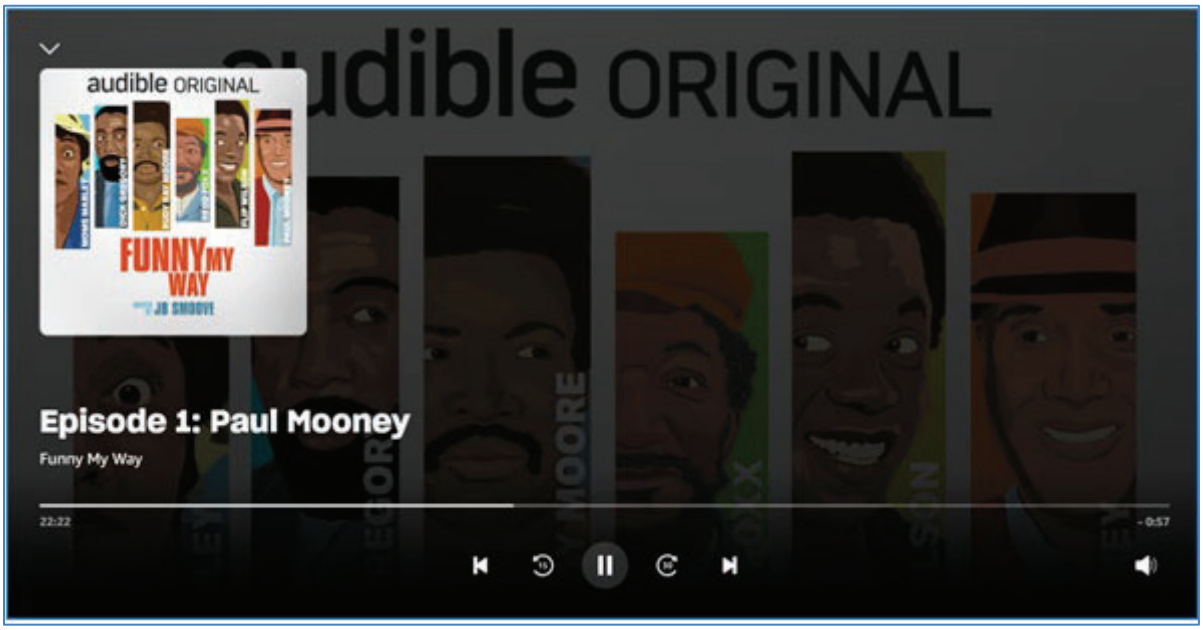
See https://developer.amazon.com/docs/music/playback_overview.html.



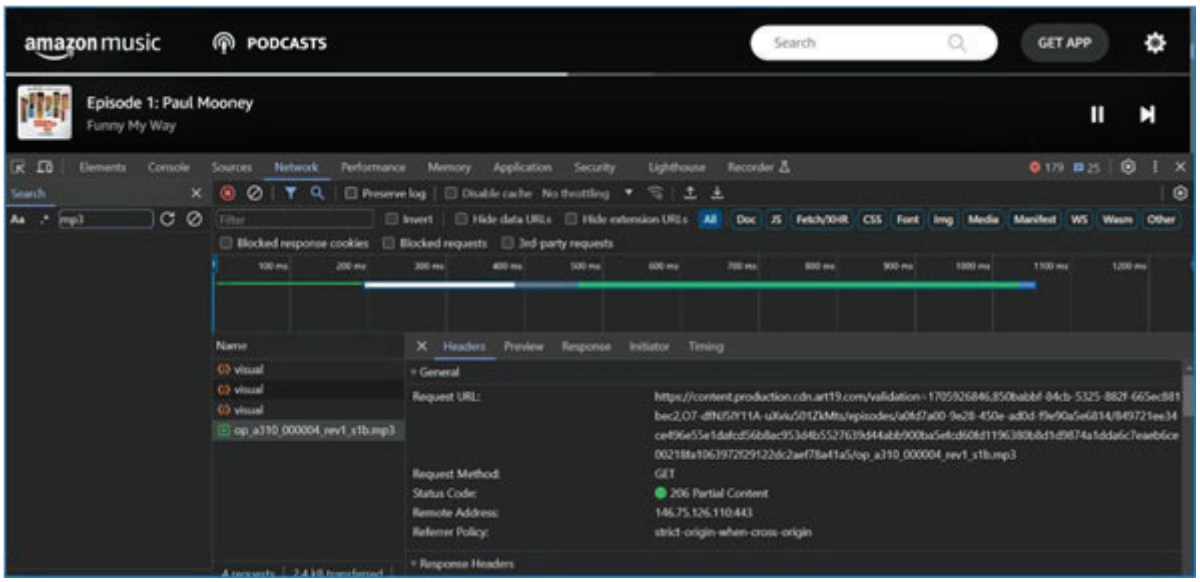
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



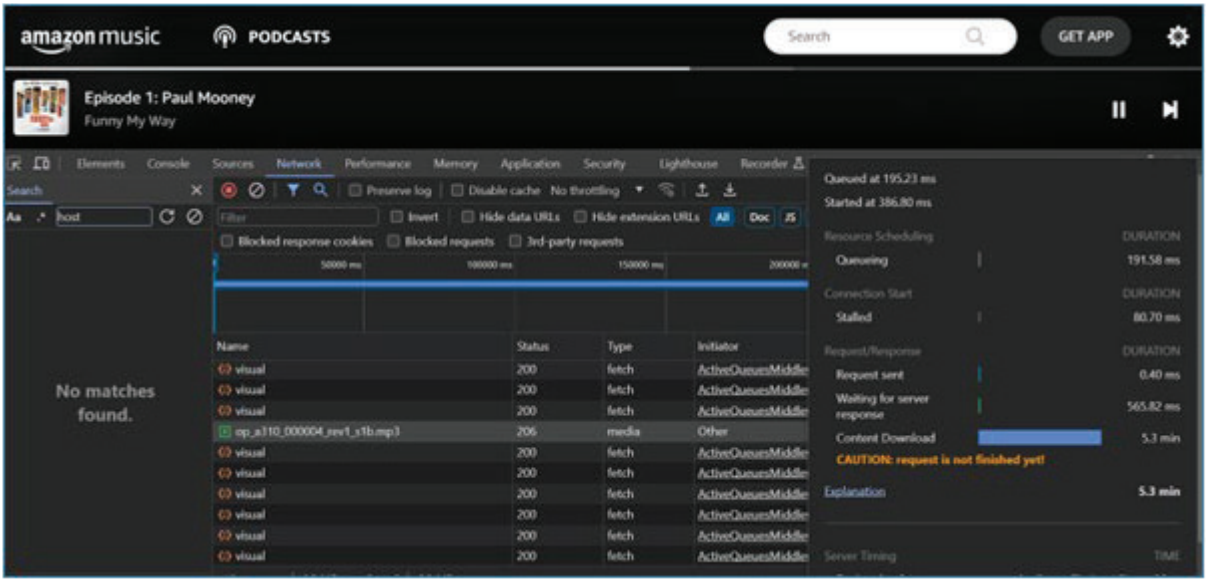
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

Caching

A "'cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.

125. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising storing (e.g., buffering chunks of audio stream data on a client-side device) the first digital content on the first client device.

Streaming services, such as Amazon Music, have revolutionized the way we listen to music. Unlike traditional methods of purchasing and downloading individual songs or albums, streaming platforms allow users to access an extensive catalog of music for a monthly subscription fee or through free ad-supported options. This vast library of songs is stored on remote servers, and users can stream them directly to their devices over the internet.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

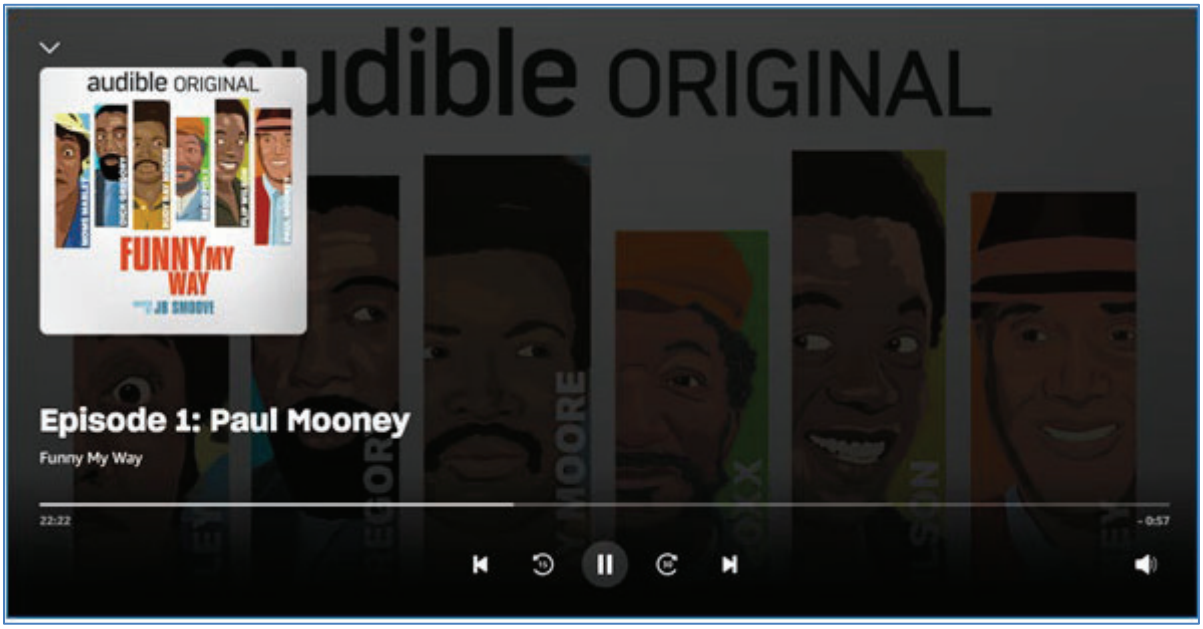
See https://developer.amazon.com/docs/music/playback_overview.html.

Caching

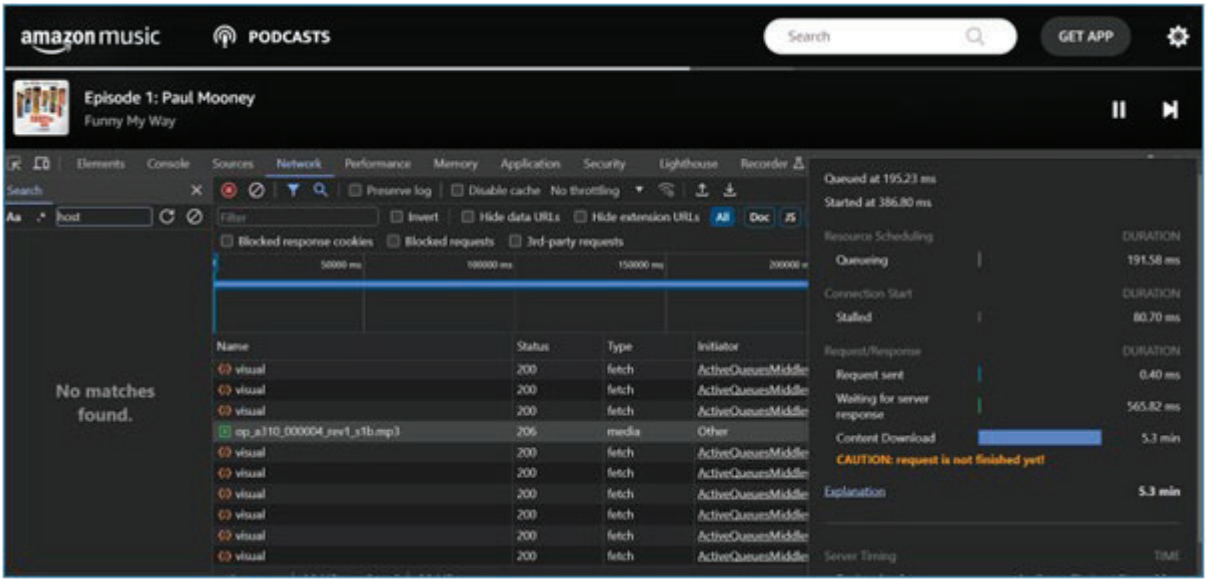
A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

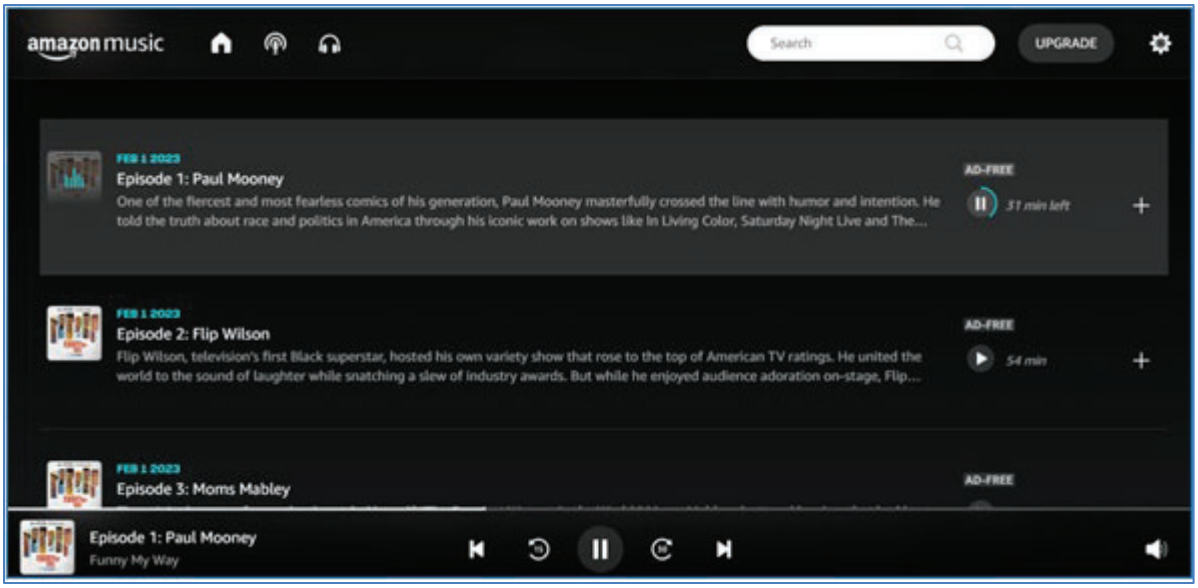


See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

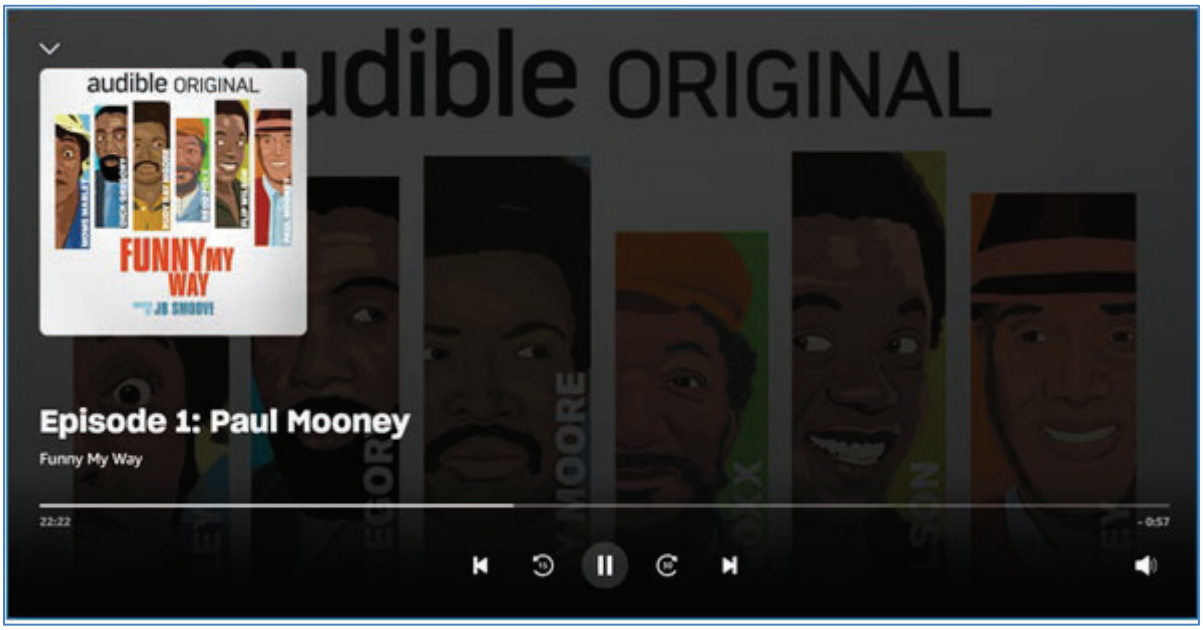
126. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising rendering (e.g., presenting the media content) at least a portion (e.g., chunk) of the first digital content on the client device.

> When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

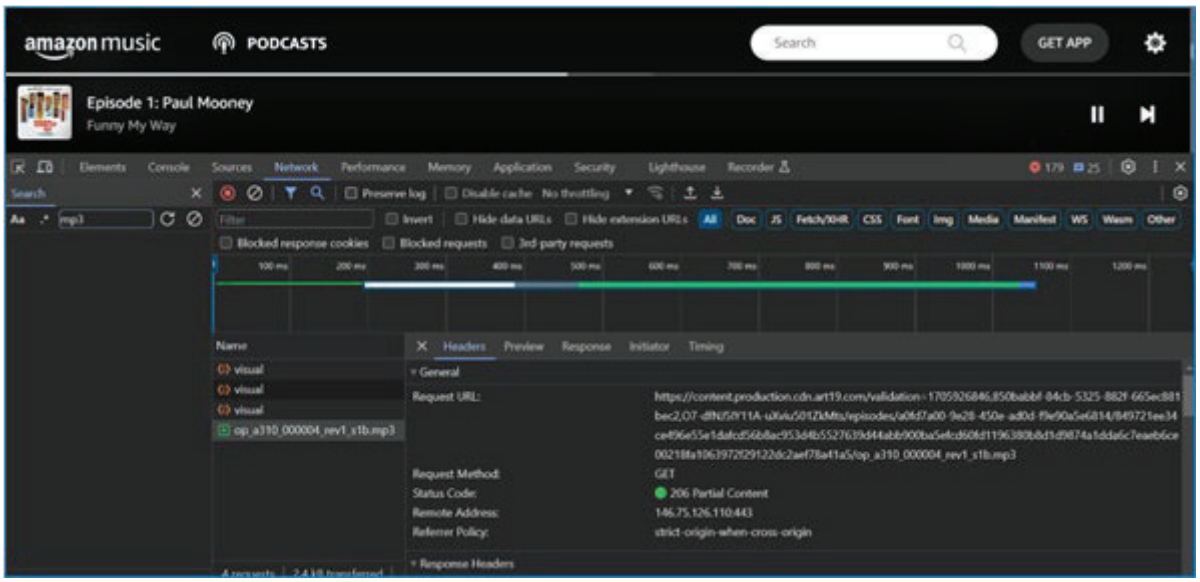
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



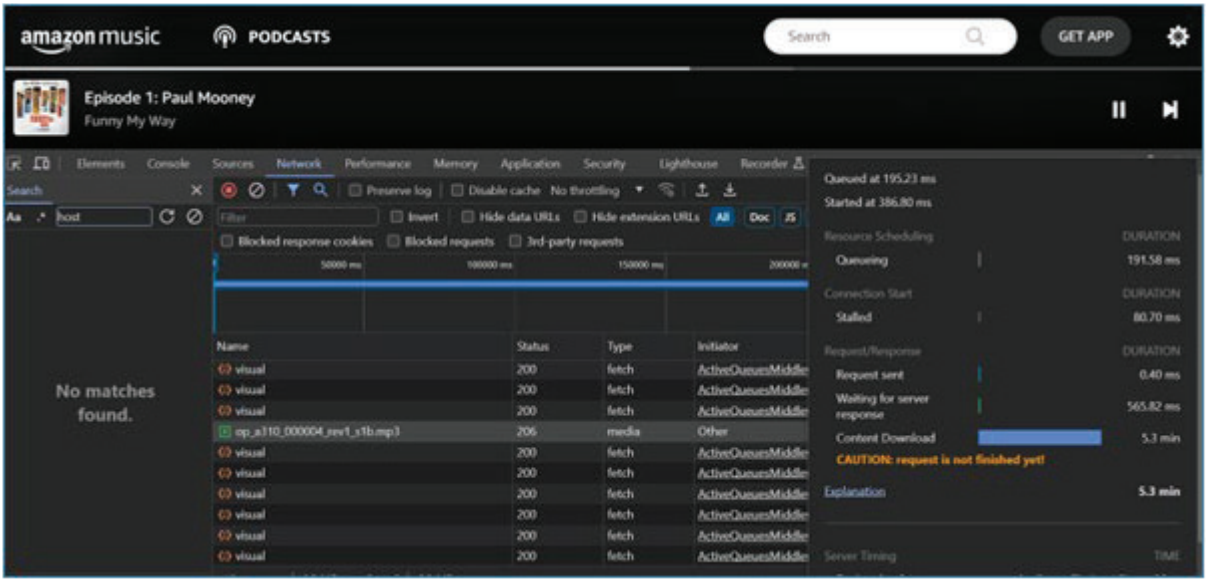
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



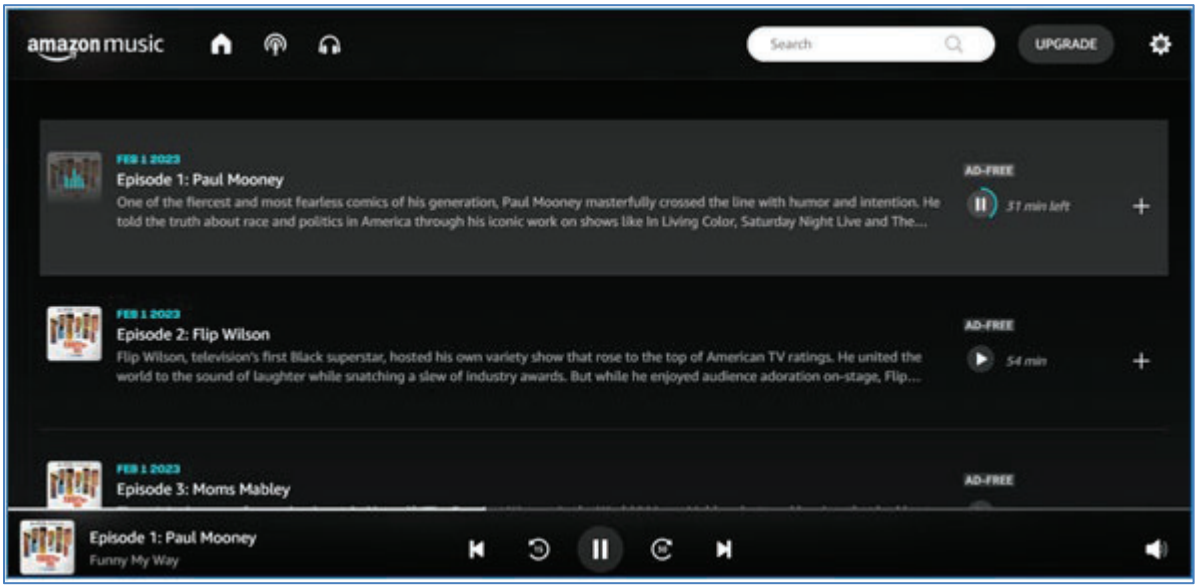
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



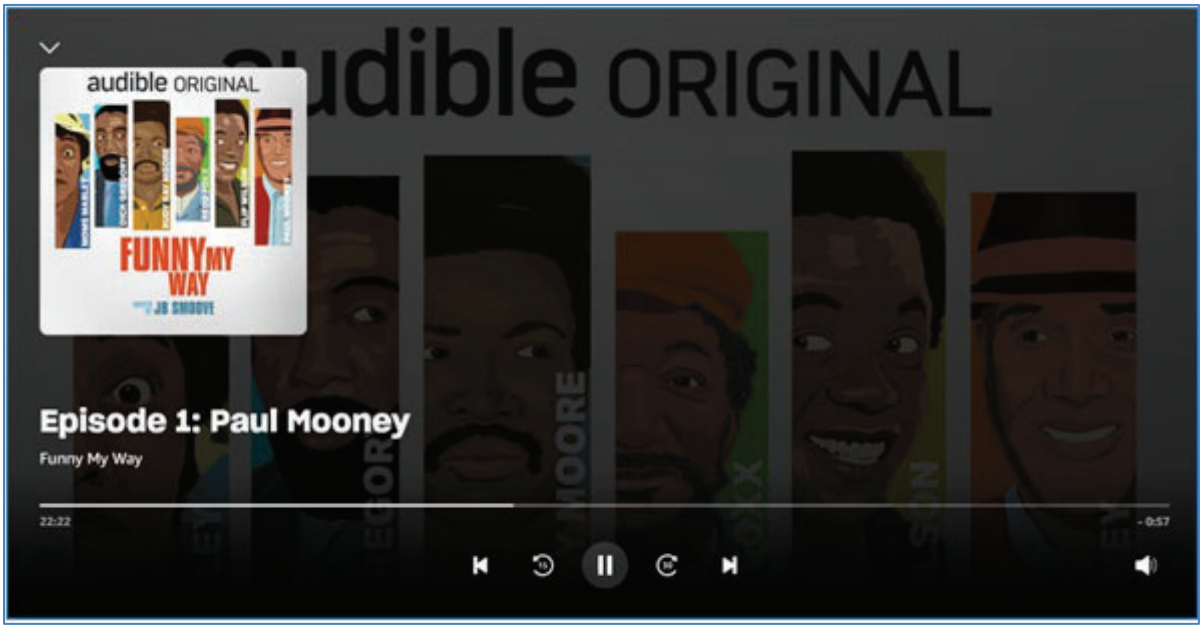
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

127. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising tracking a current

position in the first media stream as the first digital content is rendered (e.g., determining and displaying the progress bar for the audio/media content).

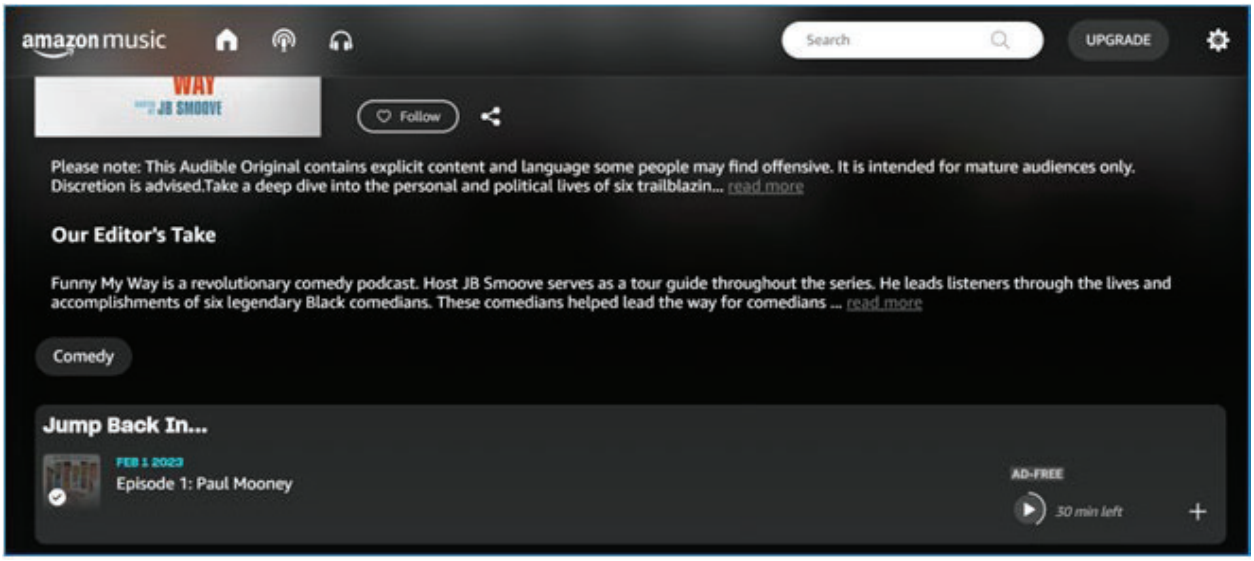


See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

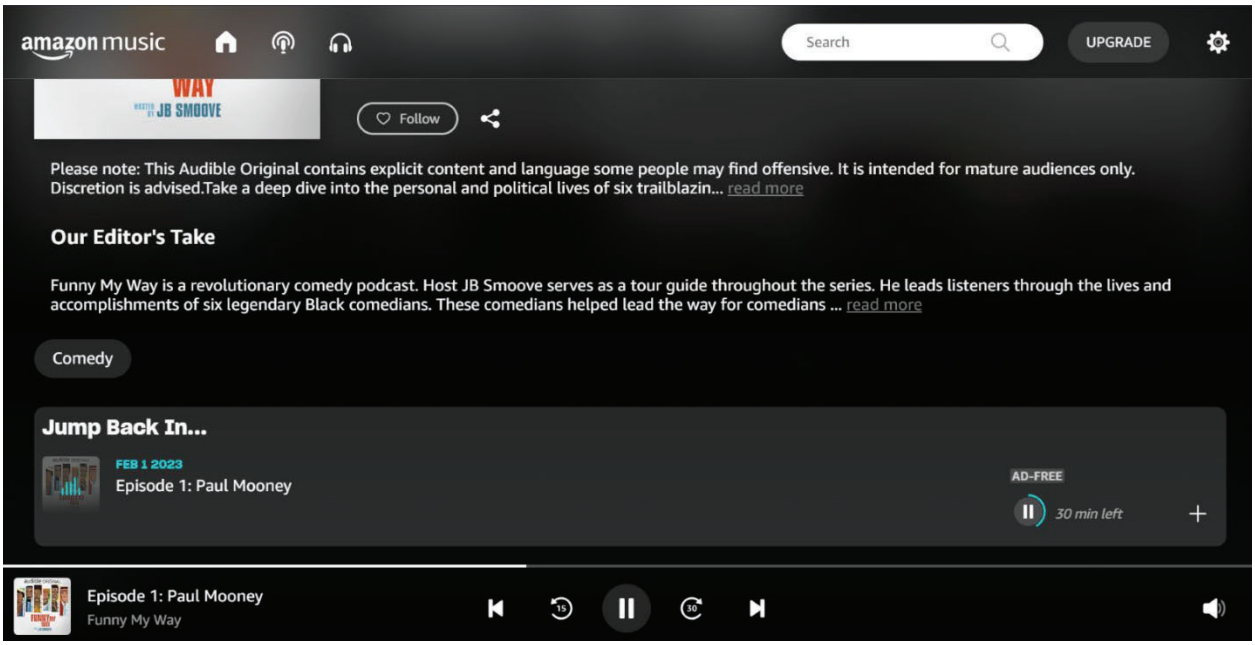


See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

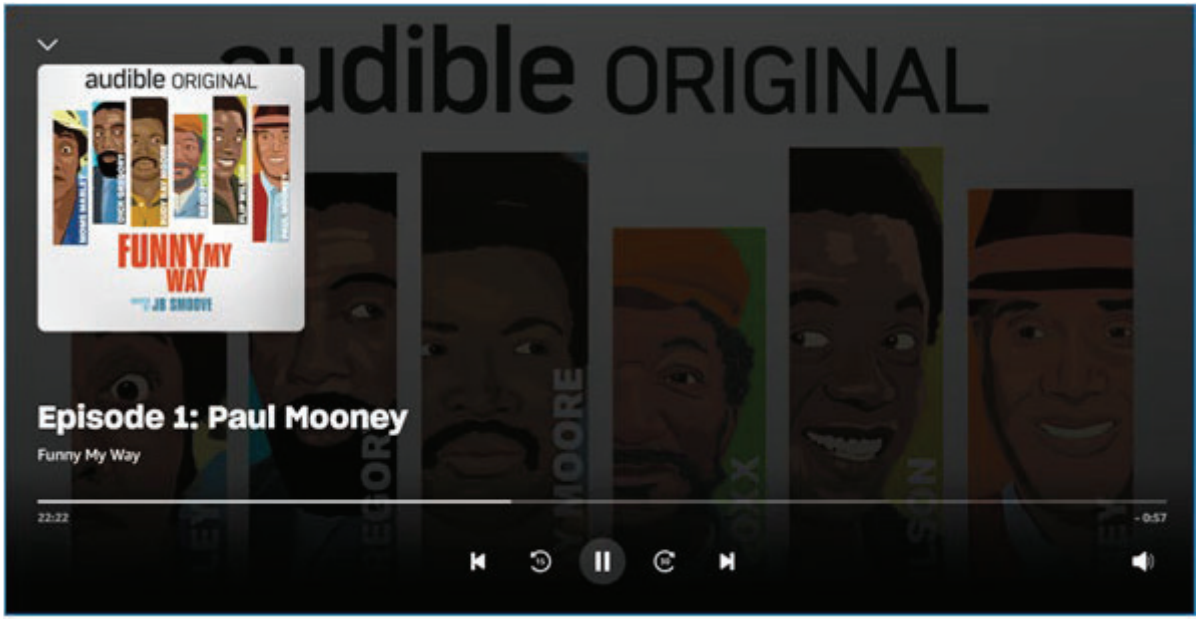
128. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising the step of creating a bookmark (e.g., creating a data component that includes timing information to indicate the stop/pause position for the media stream) by setting the current position as a bookmarked position, the bookmark including information for identifying the media work and the bookmarked position.



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

129. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising transferring the bookmark to a second client device via the network (e.g., the "Jump Back In" feature allows a

second device accessing Prime Music application or website to begin listening to an audio stream where a first device left off).

Amazon Music Unlimited Streaming Limits on Multiple Devices

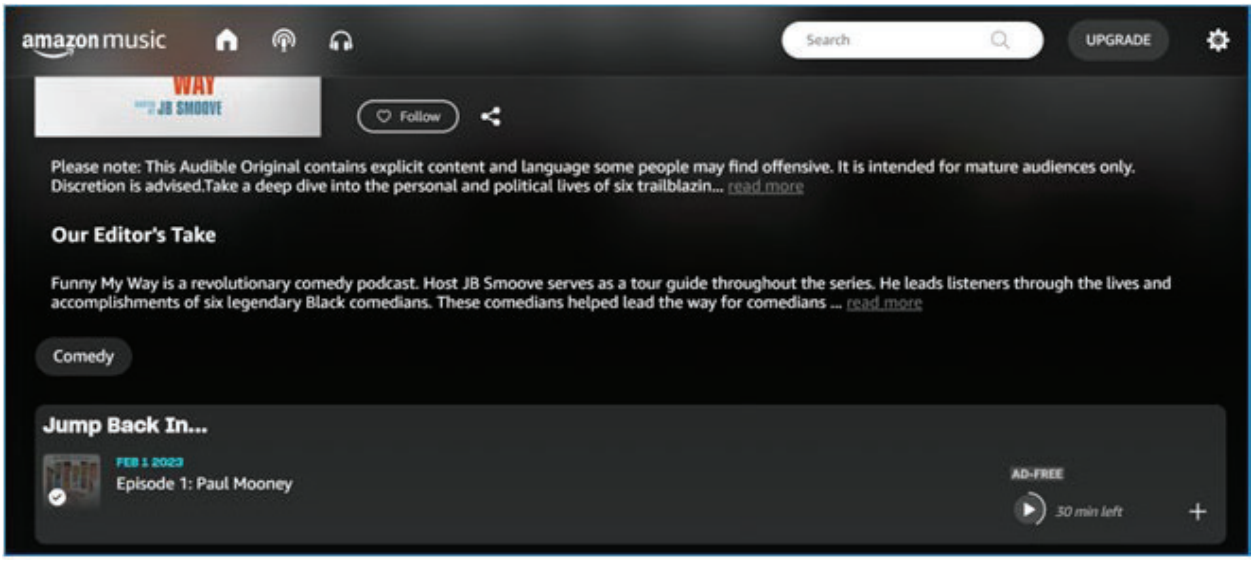
Each plan has different streaming limitations.

The Amazon Music Unlimited Family Plan allows you to stream up to six devices at the same time.

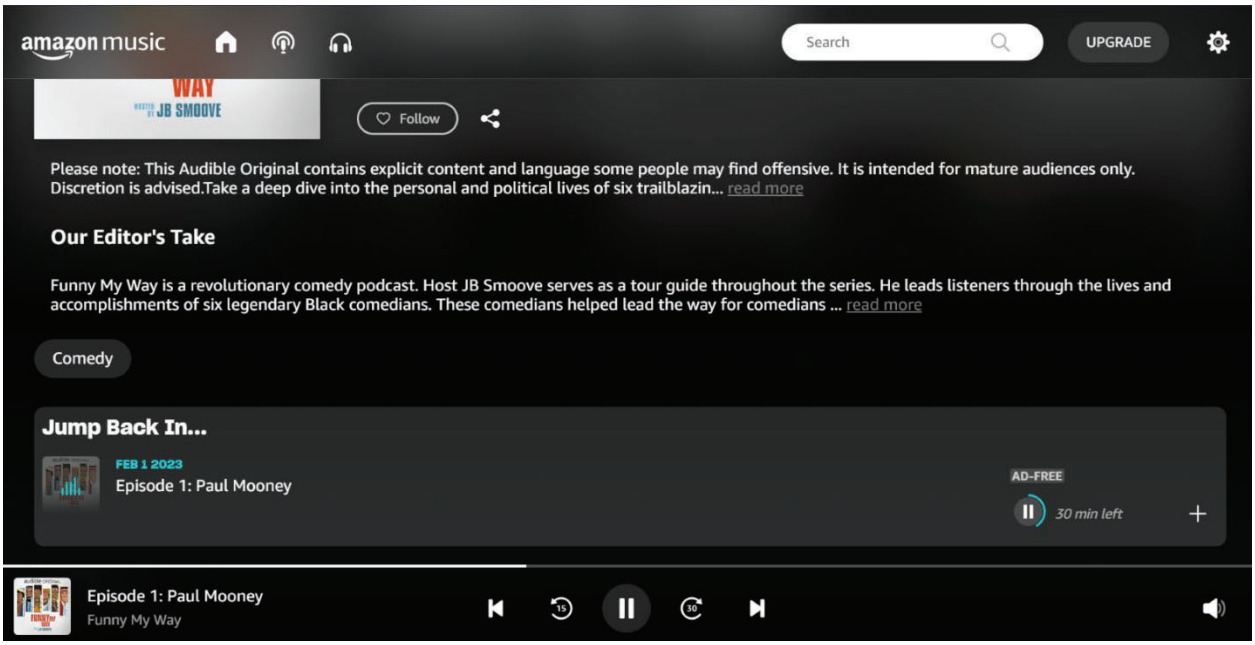
The Amazon Music Unlimited Individual Plan allows you to listen to Amazon Music Unlimited titles on all your devices. Streaming is limited to one device at a time.

The Amazon Music Unlimited Single Device Plan only allows for streaming Amazon Music Unlimited titles from the device you started your subscription on.

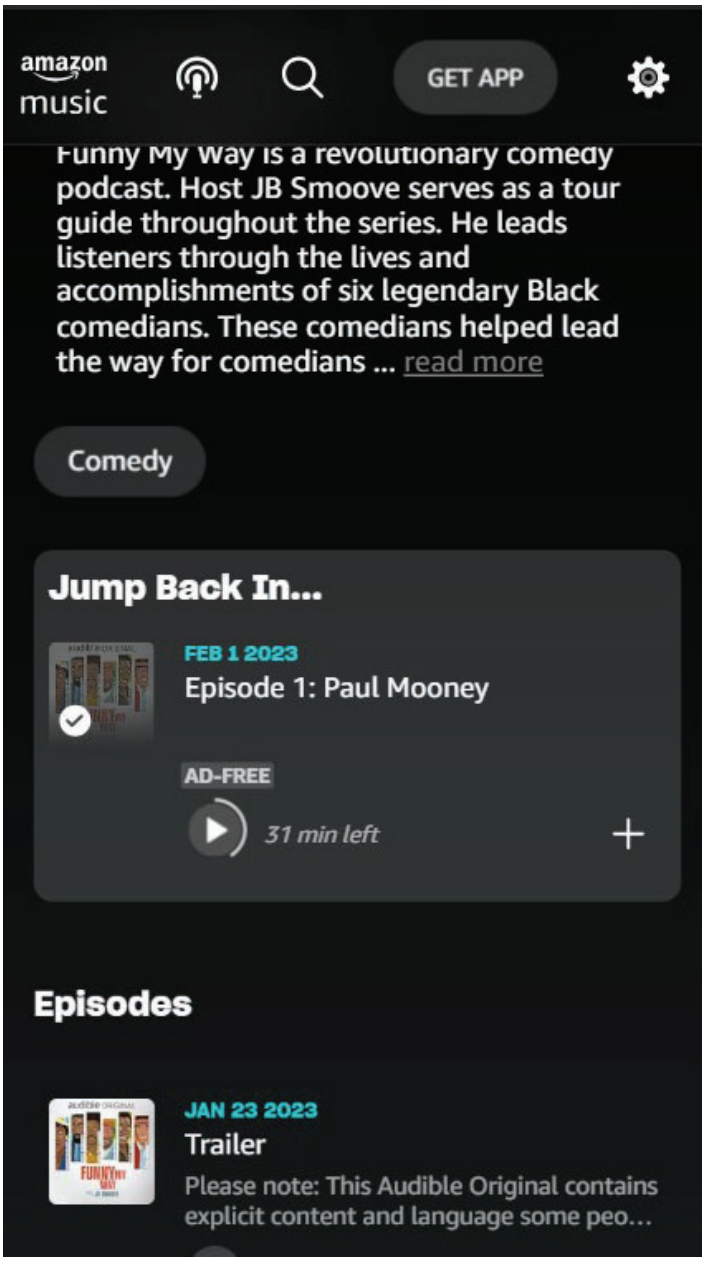
See https://www.amazon.com/gp/help/customer/display.html?nodeId=GA3WMK9TWFN8PEK8.



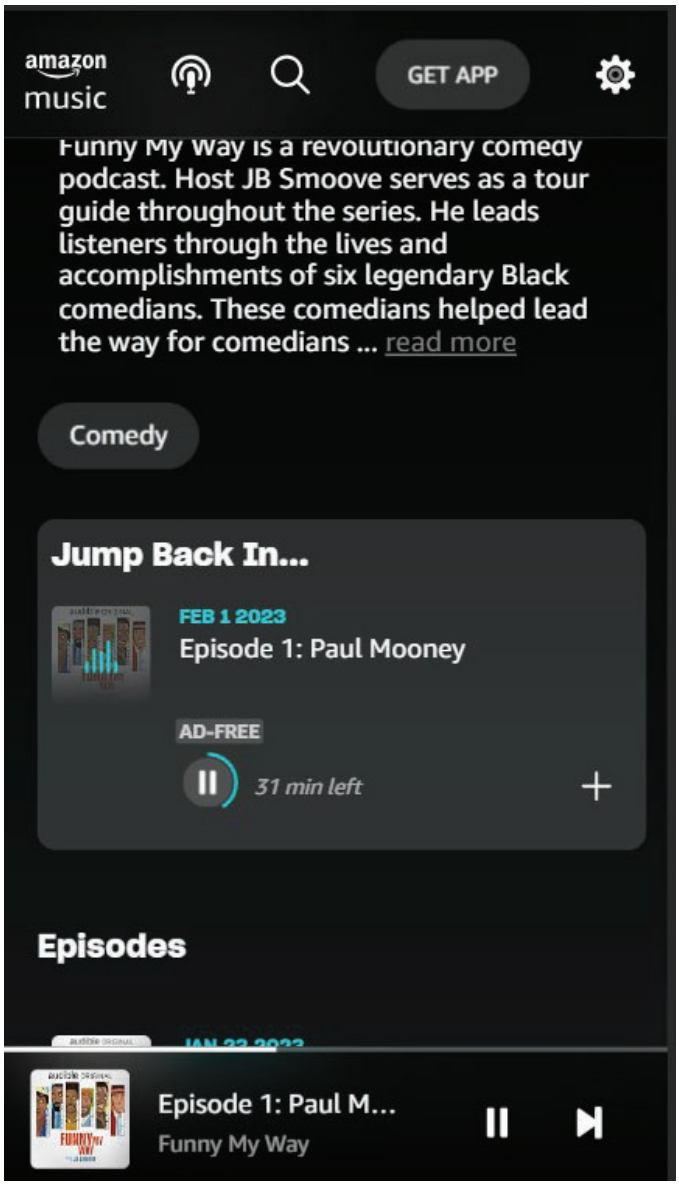
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



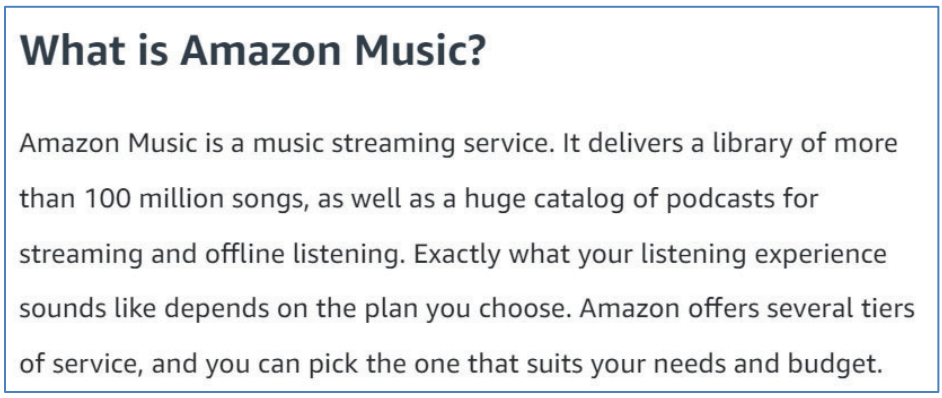
See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

130. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising downloading second digital content corresponding to the media work from the network accessible library to the second client device via the network, the second digital content including at least a second portion of the first media stream (e.g., a second "chunk" of the first media stream) or at least a

portion of a second media stream (e.g., a first "chunk" of the second media stream), the second portion of the first media stream or the portion of the second media stream including the bookmarked position.



What is Amazon Music?

Amazon Music is a music streaming service. It delivers a library of more than 100 million songs, as well as a huge catalog of podcasts for streaming and offline listening. Exactly what your listening experience sounds like depends on the plan you choose. Amazon offers several tiers of service, and you can pick the one that suits your needs and budget.

See https://www.aboutamazon.com/news/entertainment/what-is-amazon-music.

Streaming services, such as Amazon Music, have revolutionized the way we listen to music. Unlike traditional methods of purchasing and downloading individual songs or albums, streaming platforms allow users to access an extensive catalog of music for a monthly subscription fee or through free ad-supported options. This vast library of songs is stored on remote servers, and users can stream them directly to their devices over the internet.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

Streaming services like Amazon Music rely on internet connectivity to deliver music to your devices. Each time you play a song, it is streamed from Amazon's servers to your device in real-time. This constant streaming requires a significant amount of data, which can quickly add up, especially for those with limited data plans or slower internet connections.

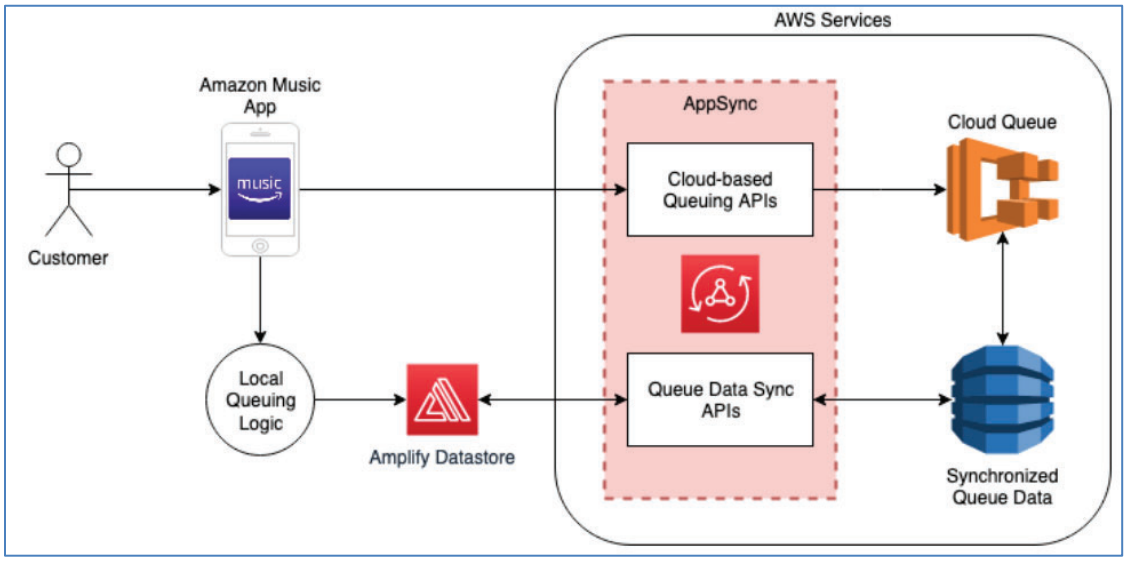
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

To stream music without issues, you need a strong Internet connection. To troubleshoot streaming issues:

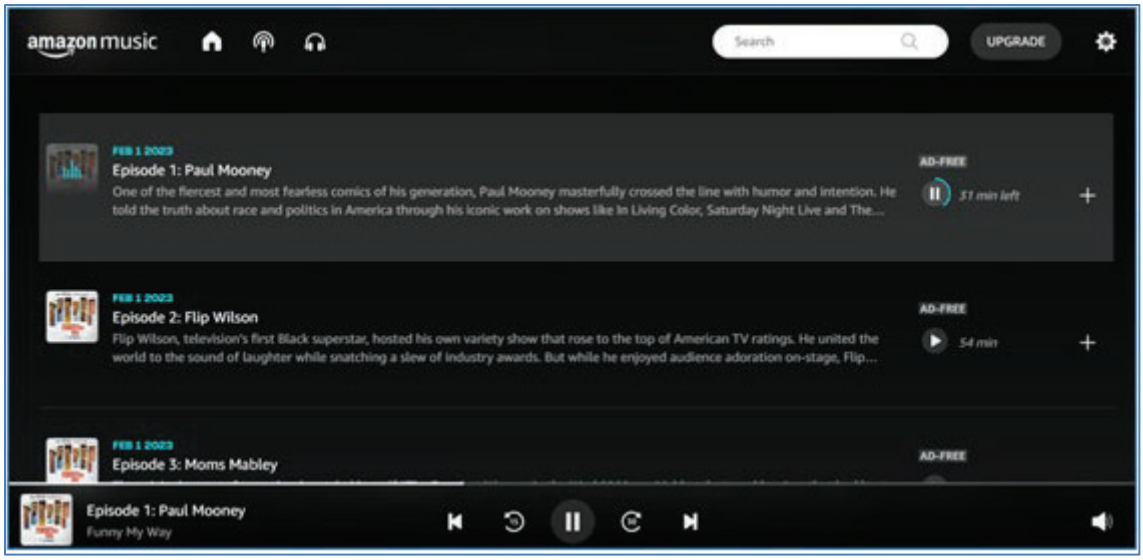
- Confirm that your device is connected to Wi-Fi or a mobile network.
- If using a mobile network, confirm that the Amazon Music app settings allow for Cellular.
- Force stop and reopen the app.

To stream HD and Ultra HD music with Amazon Music Unlimited, you need a strong Internet connection. To troubleshoot streaming issues:

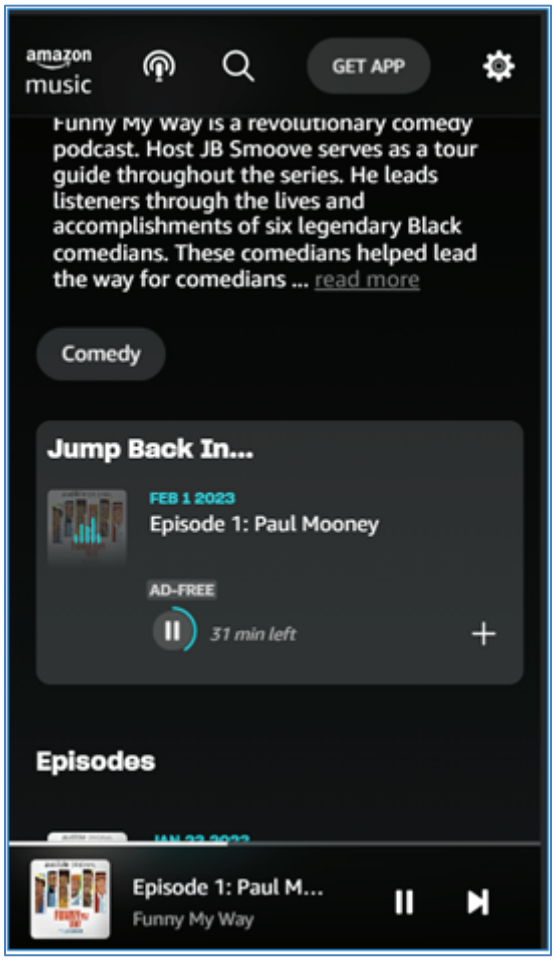
See https://www.amazon.com/gp/help/customer/display.html?nodeId=GPTQFUDBXQY85W6M.



See https://aws.amazon.com/blogs/mobile/amazon-music-unifies-music-queuing-at-scale-using-aws-appsync-and-aws-amplify/.



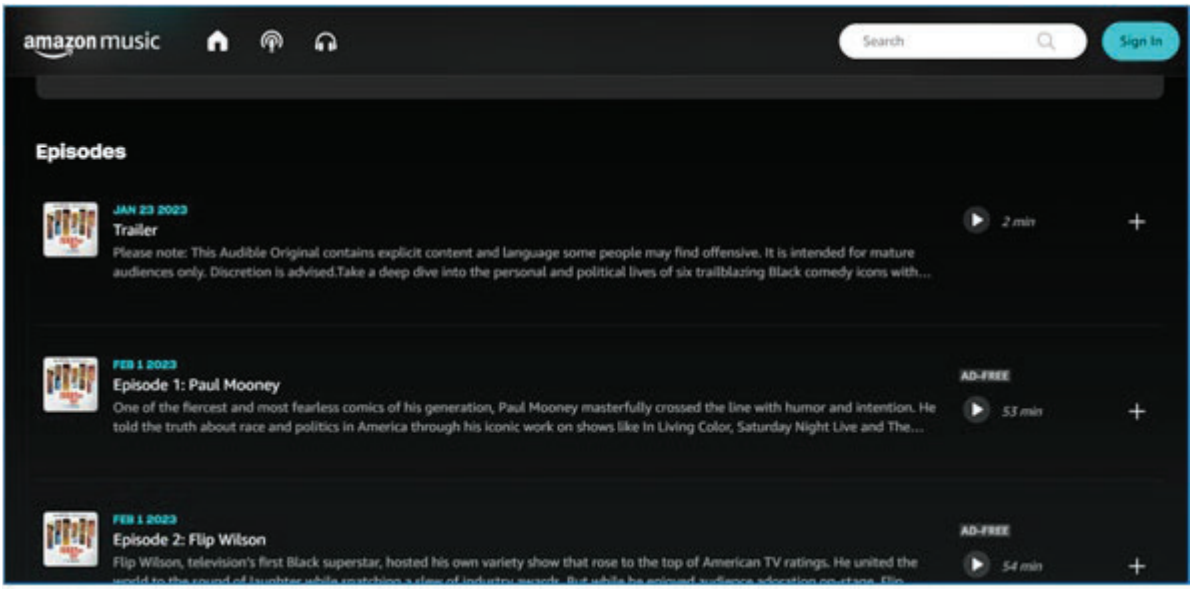
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



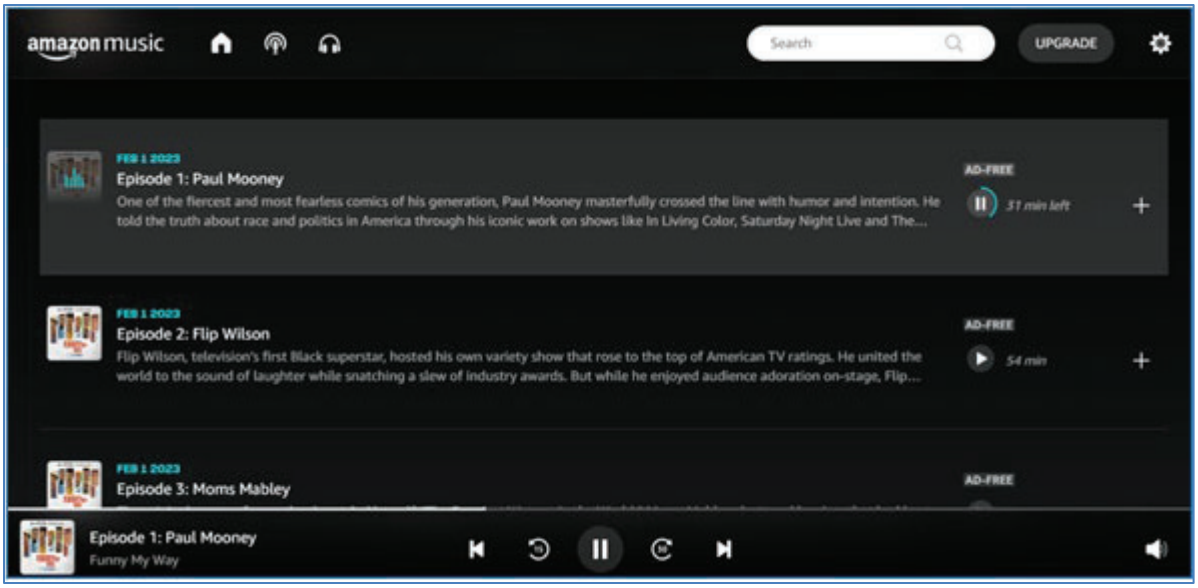
See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

> In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

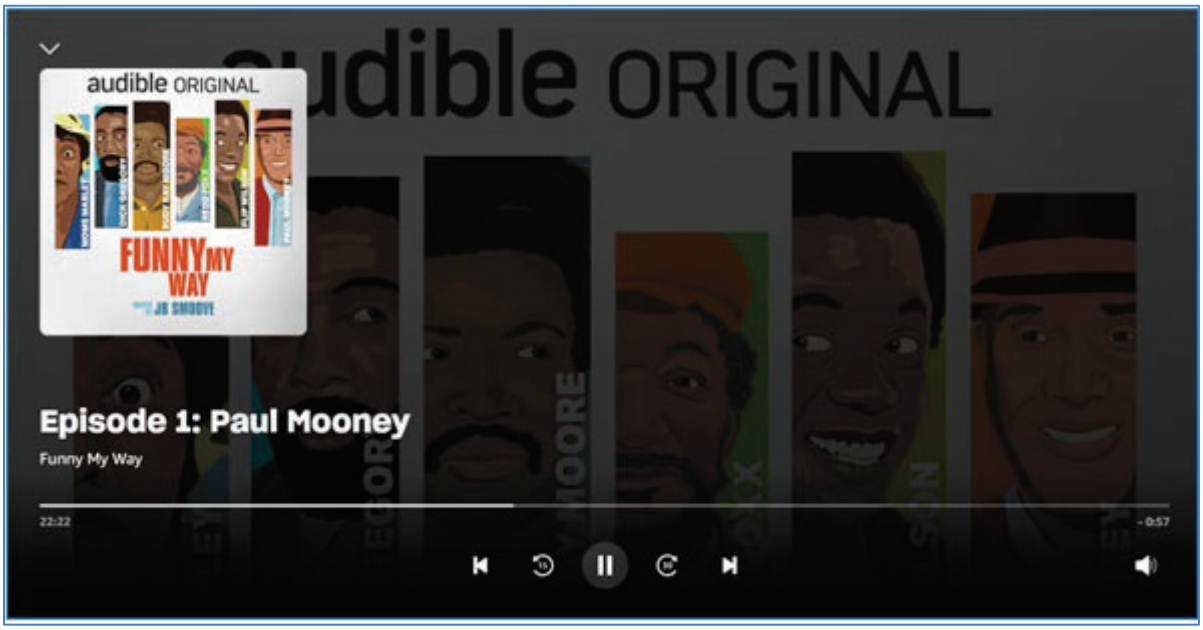
See https://developer.amazon.com/docs/music/playback_overview.html.



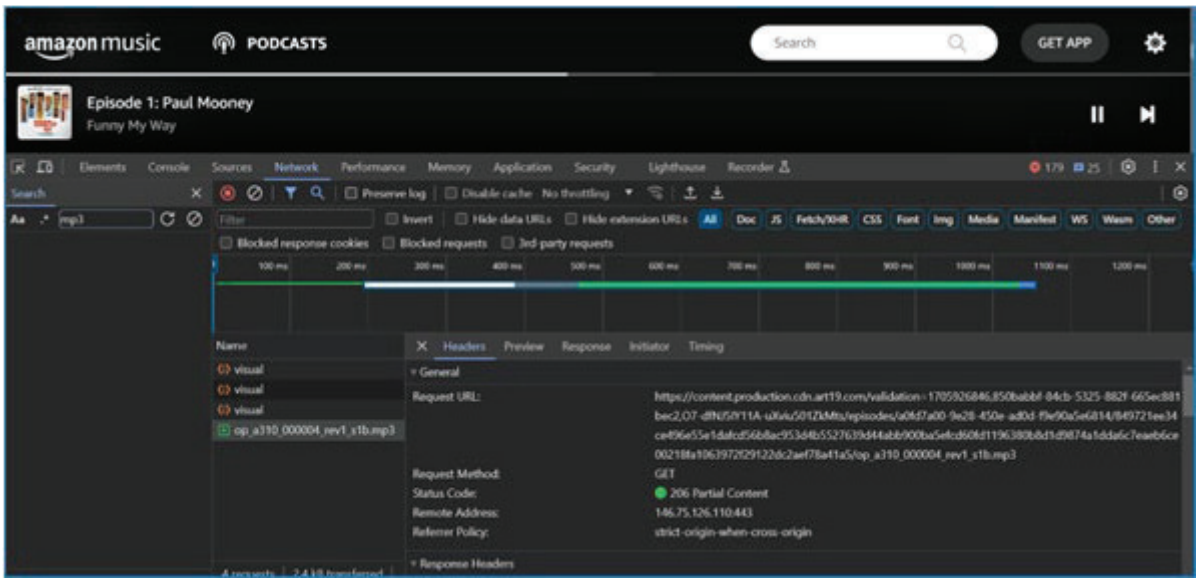
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



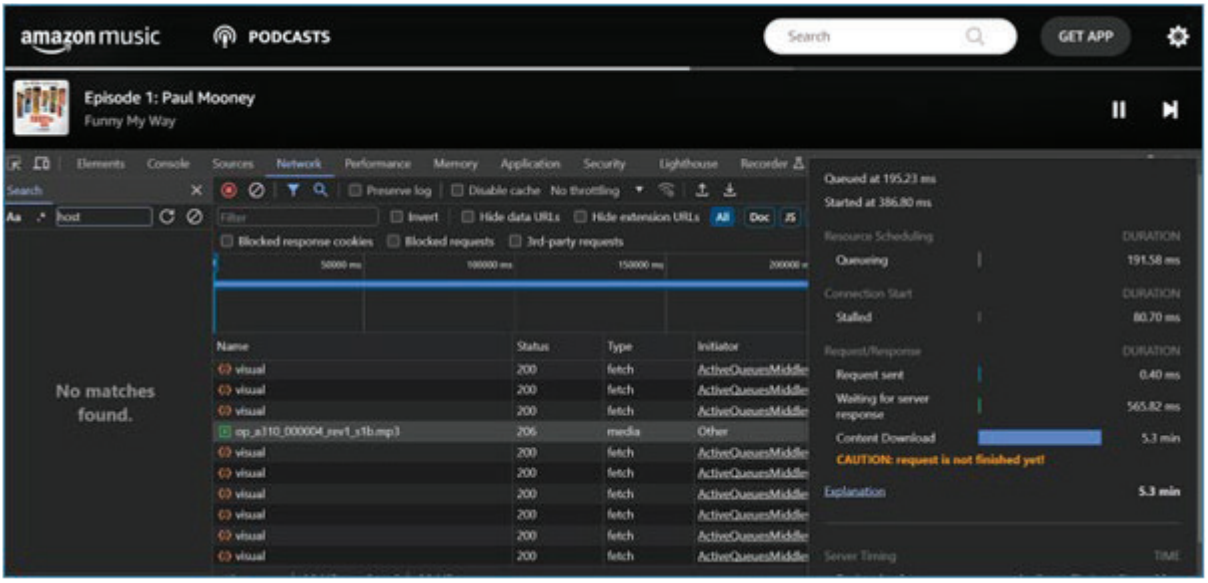
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

Caching

A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.

131. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising storing (e.g., buffering chunks of audio stream data on a client-side device) the second digital content on the second client device.

Streaming services, such as Amazon Music, have revolutionized the way we listen to music. Unlike traditional methods of purchasing and downloading individual songs or albums, streaming platforms allow users to access an extensive catalog of music for a monthly subscription fee or through free ad-supported options. This vast library of songs is stored on remote servers, and users can stream them directly to their devices over the internet.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

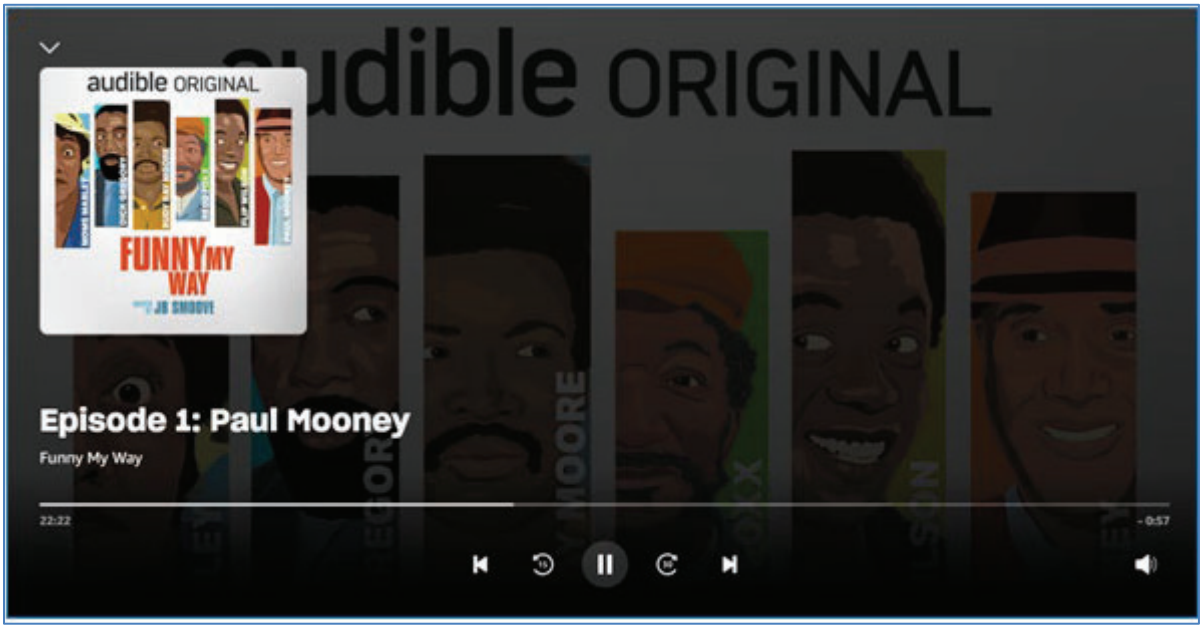
See https://developer.amazon.com/docs/music/playback_overview.html.

Caching

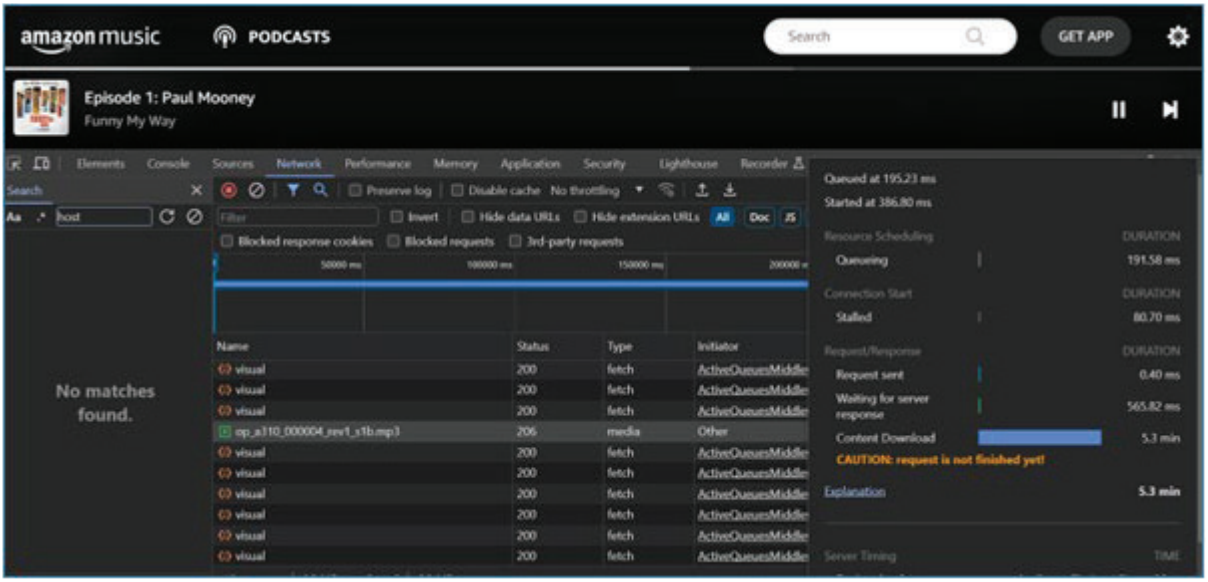
A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

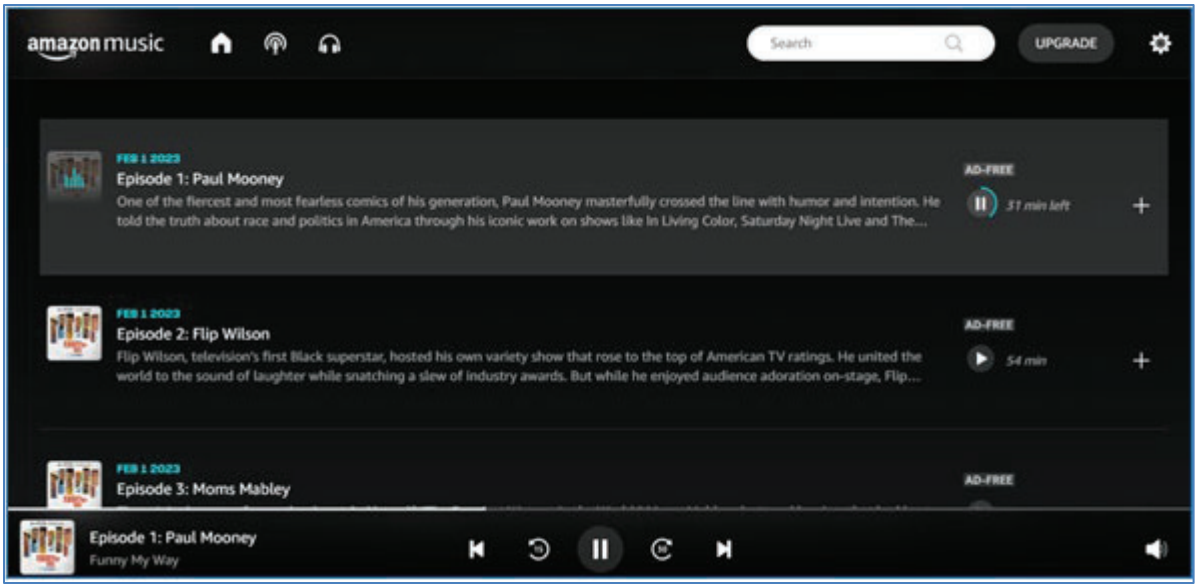


See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

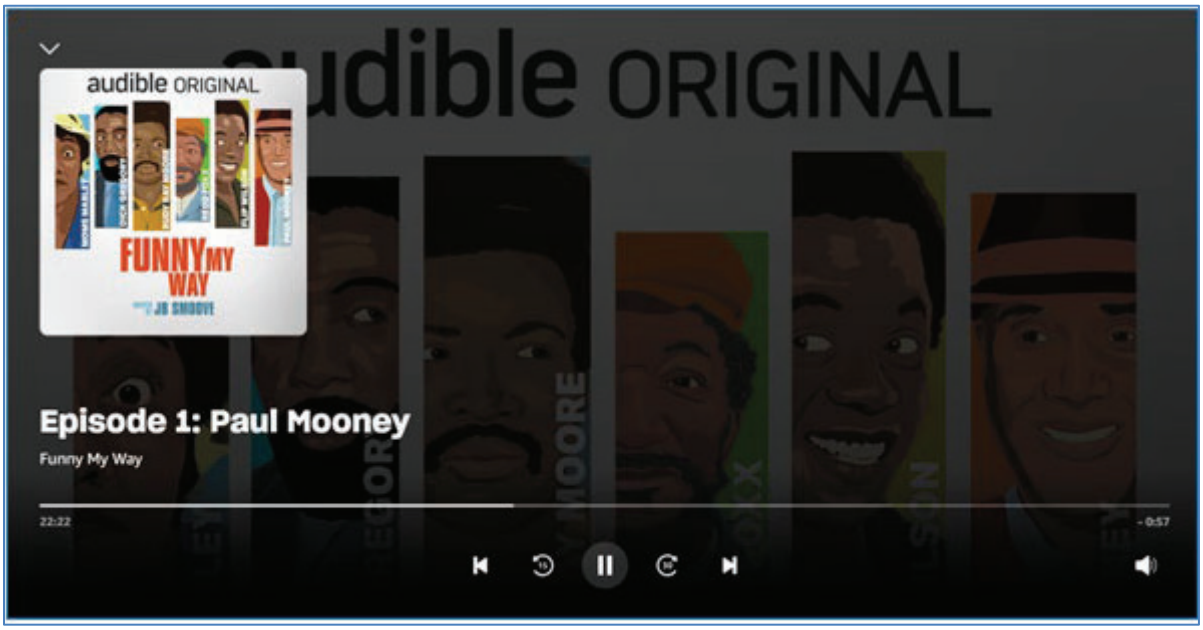
132. On information and belief, one or more components of Prime Music provide a method for rendering (e.g., presenting the media content) digital content across multiple client devices comprising rendering at least a portion (e.g., "chunk") of the second digital content on the second client device in dependence upon the bookmarked position (e.g., through the "Jump Back In" feature).

> When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

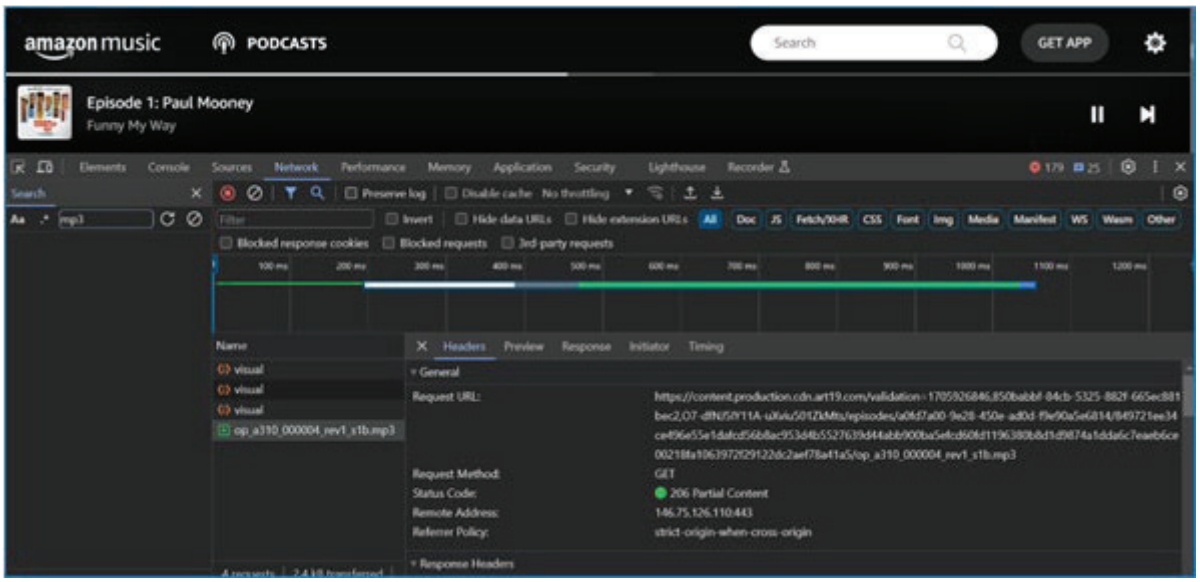


See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

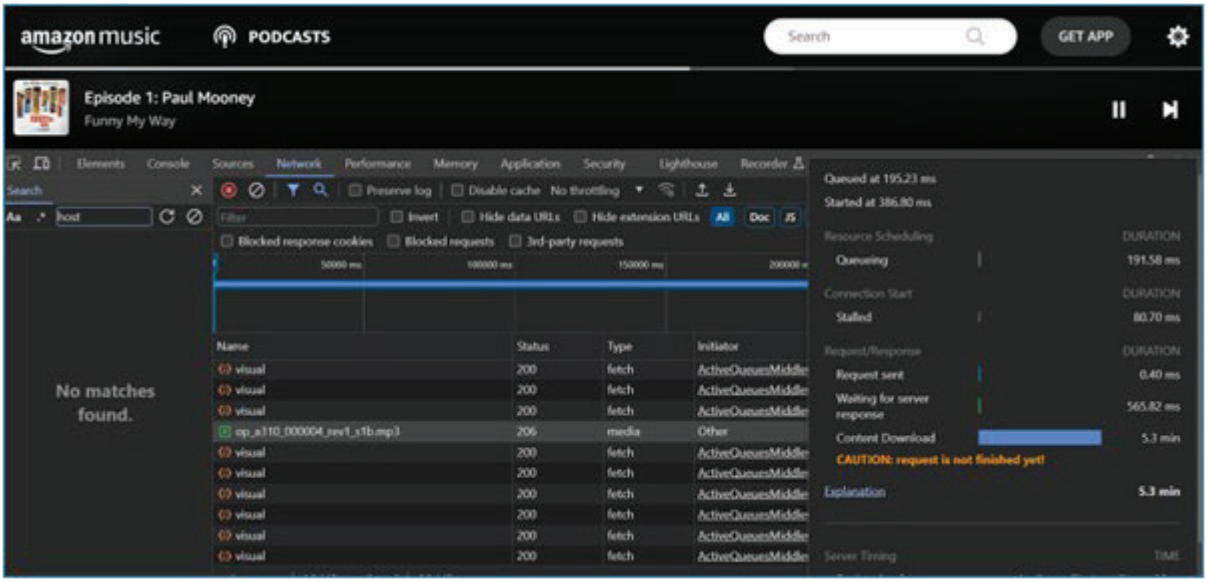


See Screenshot taken during product testing of Amazon Music (desktop version,

music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

133. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising identifying a range of content surrounding the bookmarked position in the second digital content as content to

be retained (e.g., adding new data chunks to a buffered section of the audio stream, and/or the ability go forward and backward from a progress bar position).

> In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

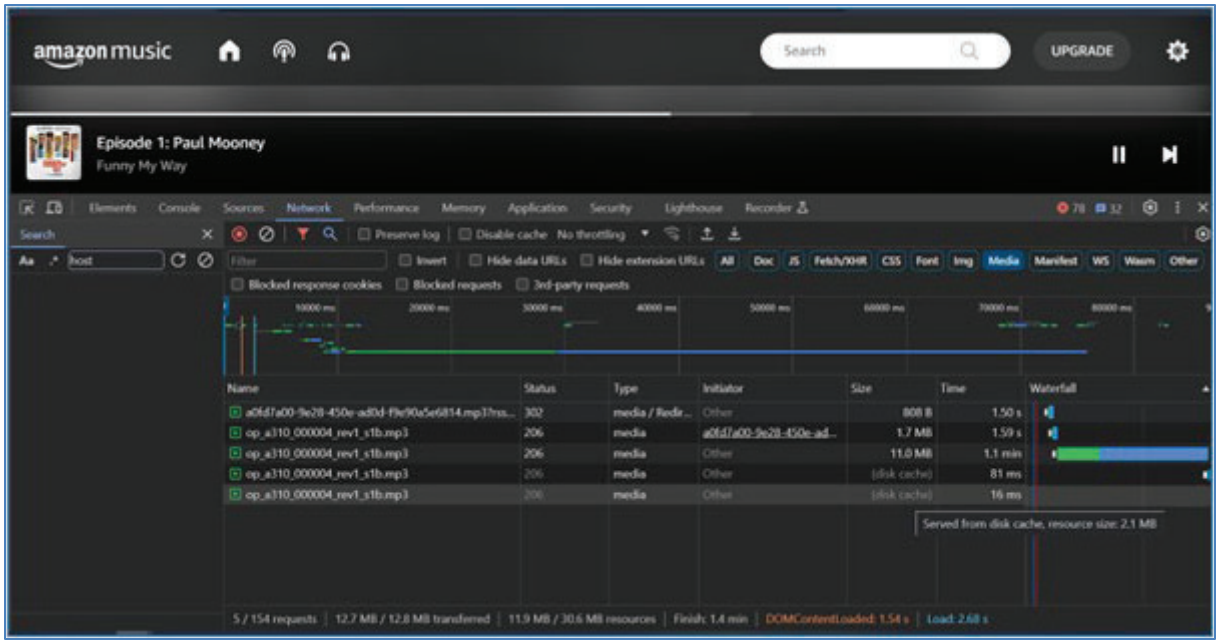
See https://developer.amazon.com/docs/music/playback_overview.html.

> When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



See Screenshot taken during network analysis of Amazon Music.

134. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising releasing storage resources allocated to all content of the second digital content that is not identified as content to be retained on the second client device (e.g., clearing the cache, and/or automatically "discard[ing]" previously played data).

Caching

A "cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

The disk cache stores resources fetched from the web so that they can be accessed quickly at a latter time if needed. The main characteristics of Chromium disk cache are:

- The cache should not grow unbounded so there must be an algorithm for deciding when to remove old entries.
- While it is not critical to lose some data from the cache, having to discard the whole cache should be minimized. The current design should be able to gracefully handle application crashes, no matter what is going on at that time, only discarding the resources that were open at that time. However, if the whole computer crashes while we are updating the cache, everything on the cache probably will be discarded.
- Access to previously stored data should be reasonably efficient, and it should be possible to use synchronous or asynchronous operations.
- We should be able to avoid conflicts that prevent us from storing two given resources simultaneously. In other words, the design should avoid cache trashing.
- It should be possible to remove a given entry from the cache, and keep working with a given entry while at the same time making it inaccessible to other requests (as if it was never stored).
- The cache should not be using explicit multithread synchronization because it will always be called from the same thread. However, callbacks should avoid reentrancy problems so they must be issued through the thread's message loop.

See https://www.chromium.org/developers/design-documents/network-stack/disk-cache/.

135. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising determining whether sufficient storage is available on the second client device to meet storage demand after releasing the storage resources allocated to the content of the second digital content that is not identified as content to be retained (e.g., buffering the maximum number of chunks of media content permitted by the available memory/cache after clearing the cache and/or discarding previously played data).

Caching

A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.

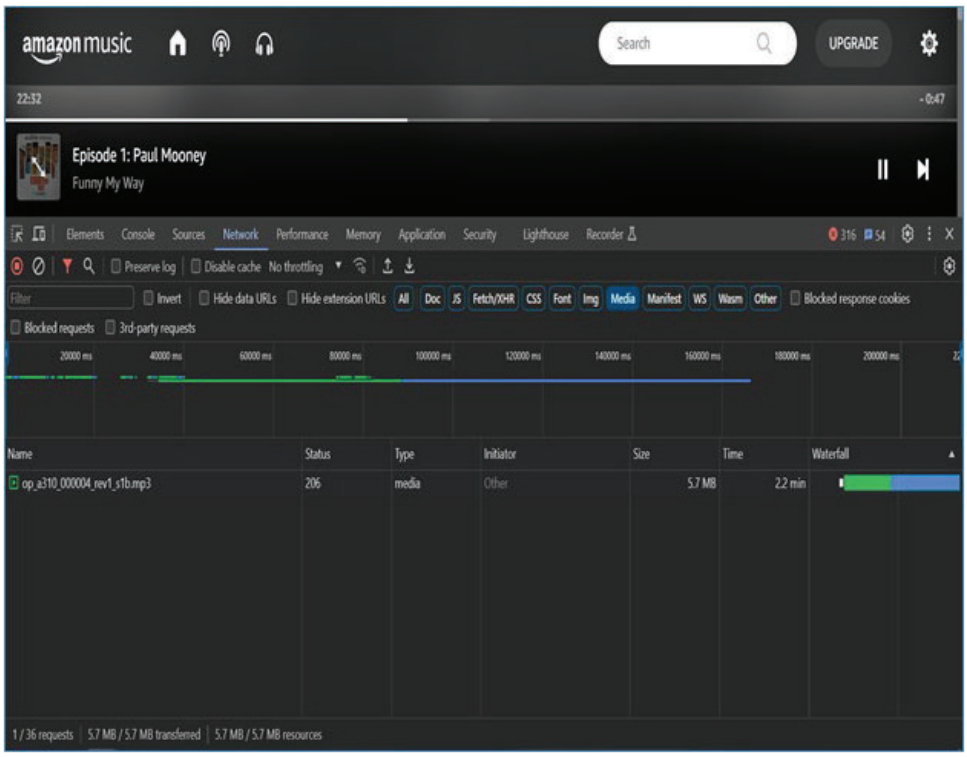
When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

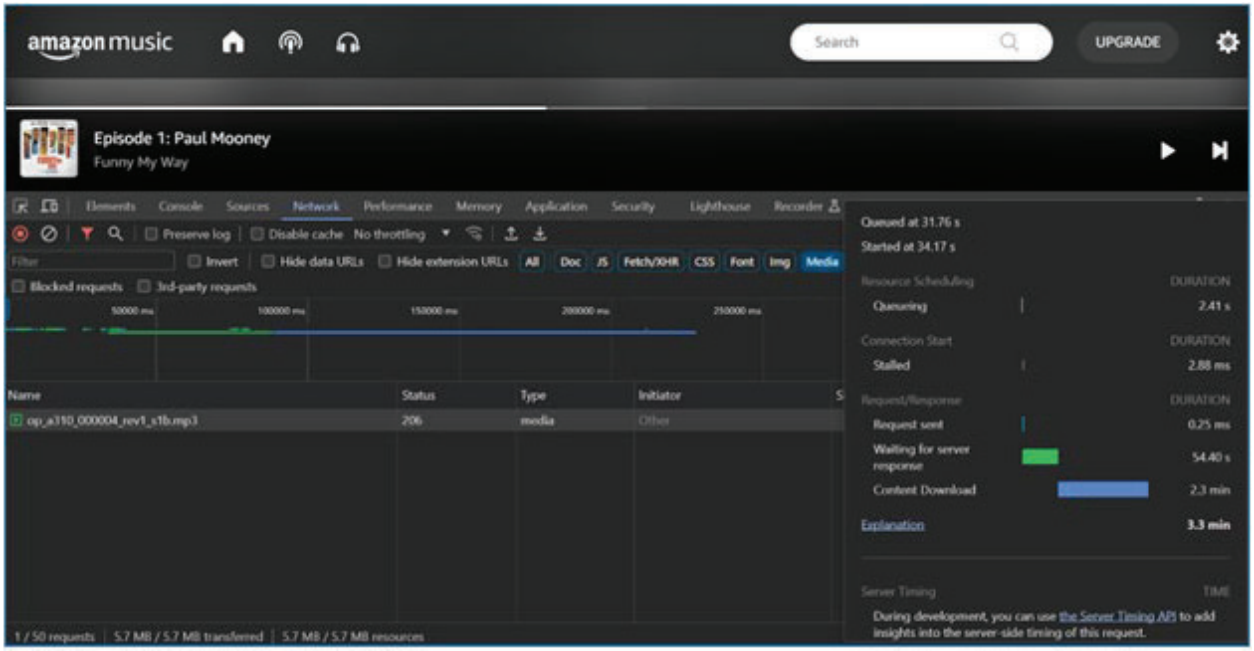
The disk cache stores resources fetched from the web so that they can be accessed quickly at a latter time if needed. The main characteristics of Chromium disk cache are:

- The cache should not grow unbounded so there must be an algorithm for deciding when to remove old entries.
- While it is not critical to lose some data from the cache, having to discard the whole cache should be minimized. The current design should be able to gracefully handle application crashes, no matter what is going on at that time, only discarding the resources that were open at that time. However, if the whole computer crashes while we are updating the cache, everything on the cache probably will be discarded.
- Access to previously stored data should be reasonably efficient, and it should be possible to use synchronous or asynchronous operations.
- We should be able to avoid conflicts that prevent us from storing two given resources simultaneously. In other words, the design should avoid cache trashing.
- It should be possible to remove a given entry from the cache, and keep working with a given entry while at the same time making it inaccessible to other requests (as if it was never stored).
- The cache should not be using explicit multithread synchronization because it will always be called from the same thread. However, callbacks should avoid reentrancy problems so they must be issued through the thread's message loop.

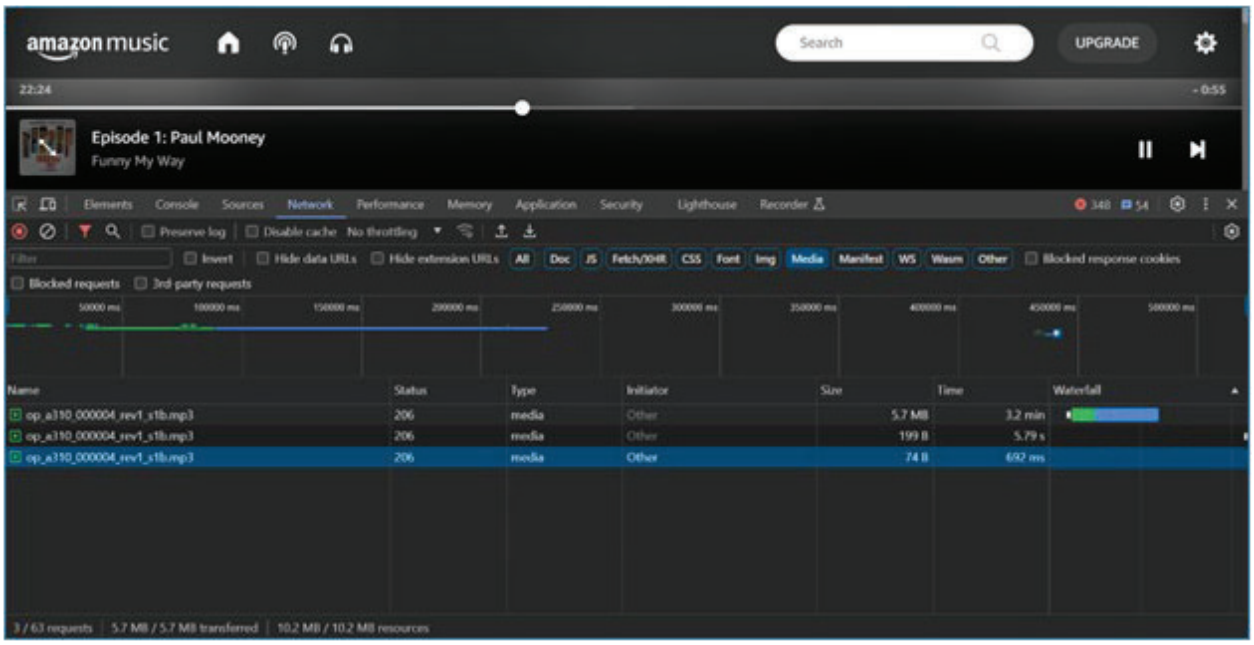
See https://www.chromium.org/developers/design-documents/network-stack/disk-cache/.



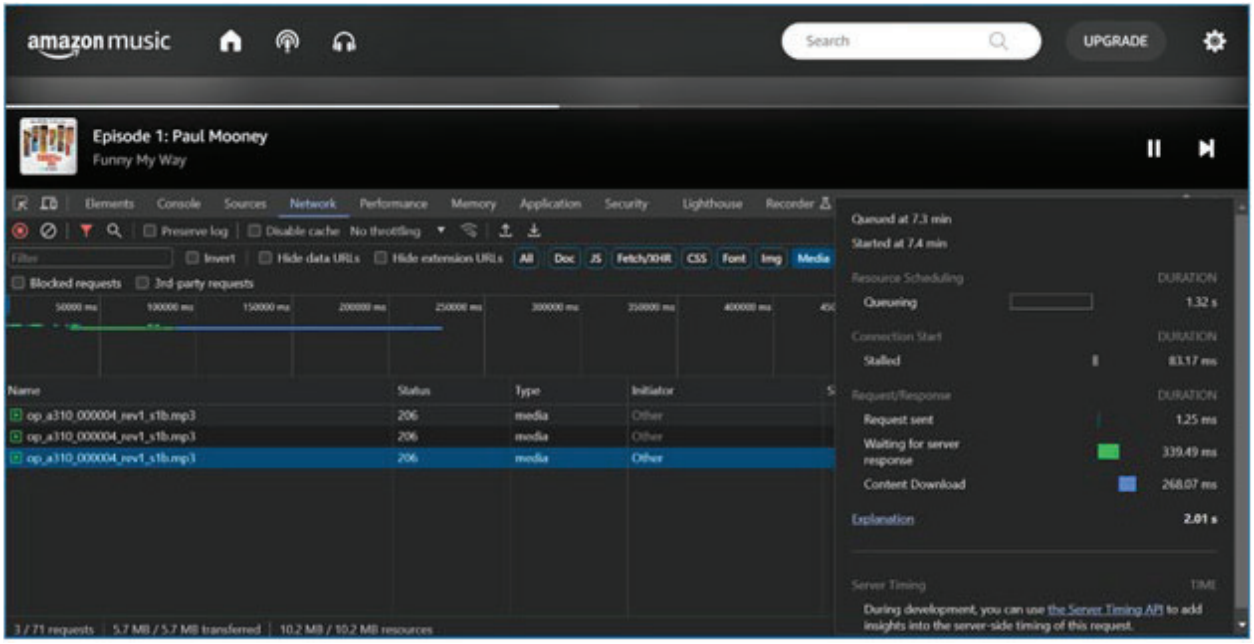
See Screenshot taken during network analysis of Amazon Music.



See Screenshot taken during network analysis of Amazon Music.



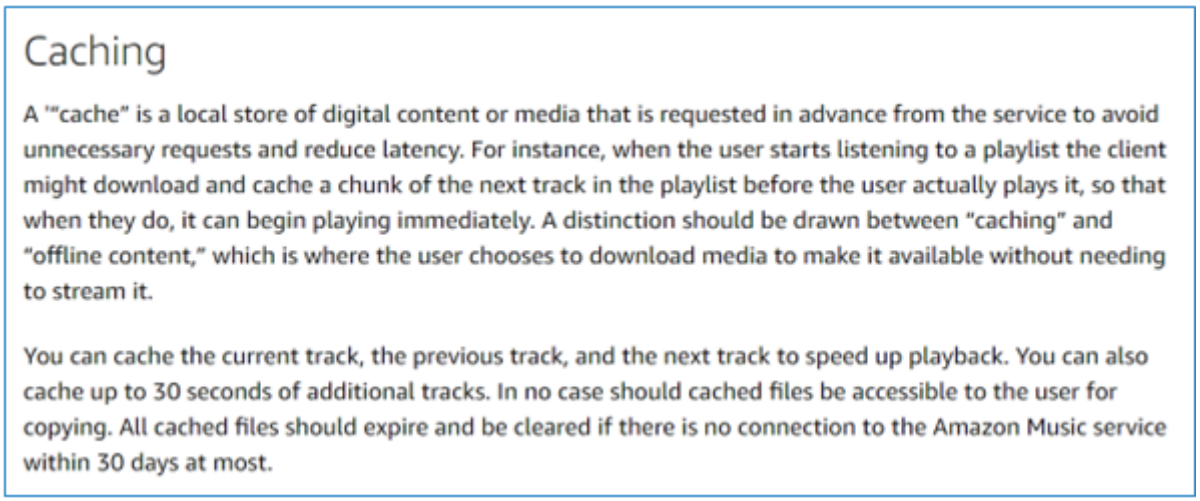
See Screenshot taken during network analysis of Amazon Music.



See Screenshot taken during network analysis of Amazon Music.

136. On information and belief, one or more components of Prime Music provide a method for rendering digital content across multiple client devices comprising if insufficient storage is available, narrowing the range of content surrounding the bookmarked position that is

identified as content to be retained (e.g., buffering more or less chunks of media content based on the available memory/cache on the client device).

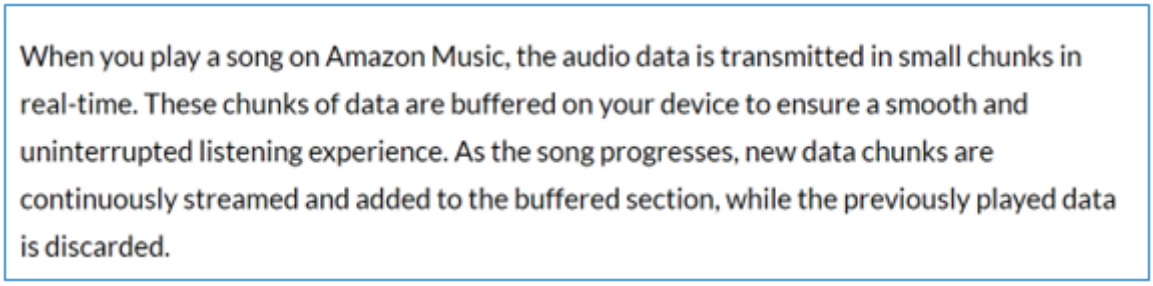


Caching

A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

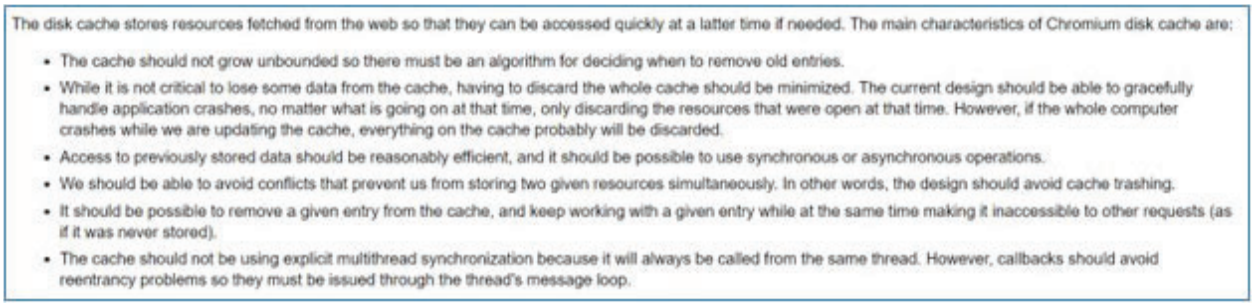
You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.



When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



The disk cache stores resources fetched from the web so that they can be accessed quickly at a latter time if needed. The main characteristics of Chromium disk cache are:

- The cache should not grow unbounded so there must be an algorithm for deciding when to remove old entries.
- While it is not critical to lose some data from the cache, having to discard the whole cache should be minimized. The current design should be able to gracefully handle application crashes, no matter what is going on at that time, only discarding the resources that were open at that time. However, if the whole computer crashes while we are updating the cache, everything on the cache probably will be discarded.
- Access to previously stored data should be reasonably efficient, and it should be possible to use synchronous or asynchronous operations.
- We should be able to avoid conflicts that prevent us from storing two given resources simultaneously. In other words, the design should avoid cache trashing.
- It should be possible to remove a given entry from the cache, and keep working with a given entry while at the same time making it inaccessible to other requests (as if it was never stored).
- The cache should not be using explicit multithread synchronization because it will always be called from the same thread. However, callbacks should avoid reentrancy problems so they must be issued through the thread's message loop.

See https://www.chromium.org/developers/design-documents/network-stack/disk-cache/.



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

> In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

See https://developer.amazon.com/docs/music/playback_overview.html.

> Taking a close look, we see that compressed streams are not single units, but are segments, each usually several seconds long. Segmenting is the process of breaking video into small parts for network transport and playback. This gives the consuming device the ability to switch between bitrates at segmentation points. The viewing device requests a copy of the manifest and updates every few seconds. When a device experiences a weak connection or plays the video in a windowed mode at small size, it requests a smaller bitrate segment. When a connection is stronger, the device can request a higher bitrate segment if screen size requires it, and so on. There are different metrics used at the player level to decide which quality to display (based on CPU utilization, network bandwidth, memory available, etc.).

See https://aws.amazon.com/blogs/media/back-to-basics-http-video-streaming/.

137. On information and belief, Defendants directly infringe at least claim 1 of the '922 patent in violation of 35 U.S.C. § 271(a) by making, using, selling, selling access to, importing, offering for sale, and/or offering to sell access to Prime Music.

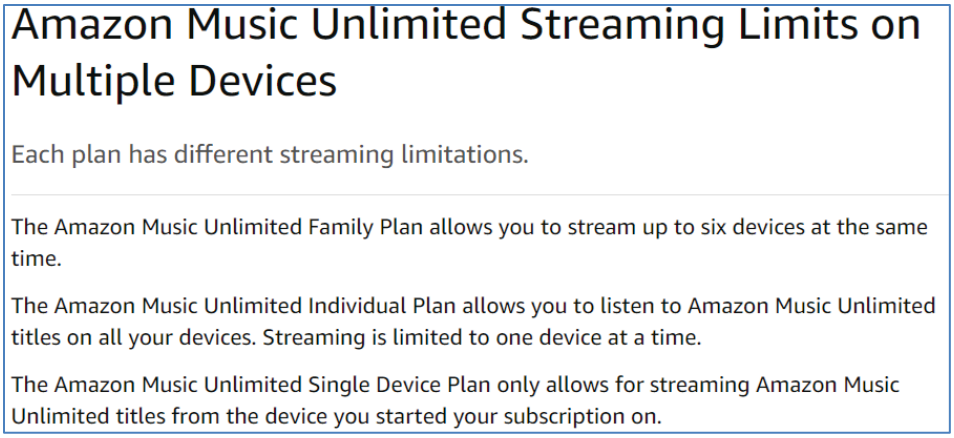
138. Defendants' infringement has damaged Audio Pod and caused/continues to cause it to suffer irreparable harm and damages.

COUNT VI - Infringement of the '266 patent by Prime Music

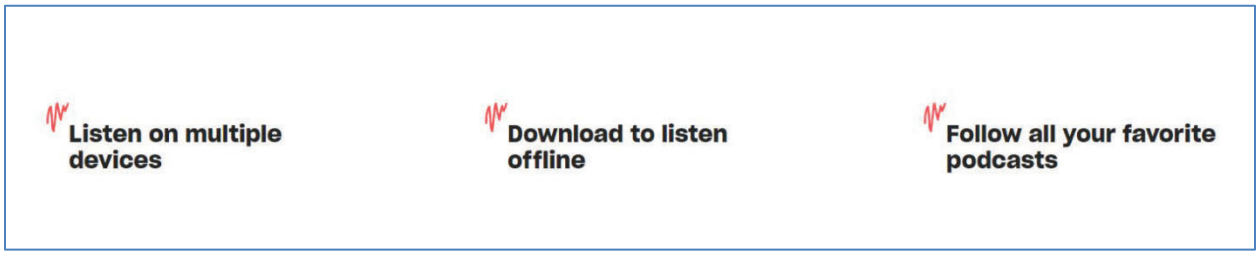
139. Audio Pod repeats, realleges, and incorporates by reference, as if fully set forth here, the allegations of the preceding paragraphs above.

140. On information and belief, Defendants (or those acting on their behalf) make, use, sell, sell access to, import, offer to sell and/or offer to sell access to Prime Music in the United States that infringes (literally and/or under the doctrine of equivalents) at least claim 1 of the '266 patent.

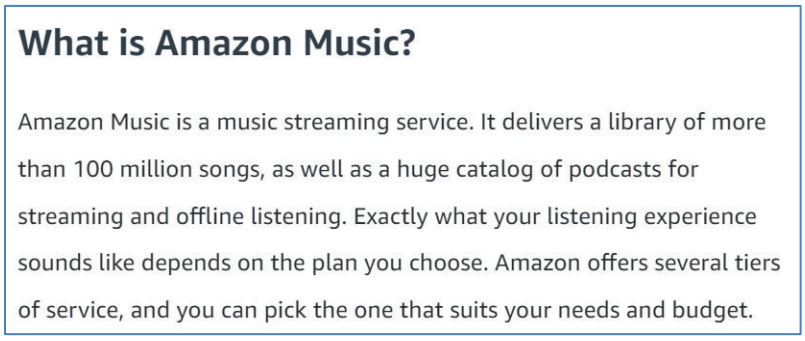
141. On information and belief, one or more components of Prime Music employs, provides, and dictates the performance of a method of rendering digital content across multiple client devices (e.g., desktops, laptops, smartphones, tablets, smart TVs, Echo devices, Alexa-enabled devices, Fire TV, etc.).



See https://www.amazon.com/gp/help/customer/display.html?nodeId=GA3WMK9TWFN8PEK8.



See https://www.amazon.com/music/lp/podcasts.



See https://www.aboutamazon.com/news/entertainment/what-is-amazon-music

Feature	Amazon Music Free	Amazon Music Prime	Amazon Music Unlimited
Amazon Prime Membership Needed	No	Yes	No
Available Titles	Discover new music and podcasts based on your likes. Curated Playlists on demand for Android and iOS. Thousands of Stations and top Playlists. Millions of podcast episodes	All the music ad-free. The most ad-free top podcasts **All-Access Playlists** – pick and play any song on-demand with no skip limits, or download them to listen offline. **Note:** Except when listening to All-Access Playlists, skip limits are applicable in most other scenarios. User-created Playlists have skip limits unless they solely include "purchased songs"	Pick and play any song, ad-free. The most ad-free top podcasts Personalized Stations and thousands of Playlists SD, HD, Ultra HD, and Spatial Audio Listen offline Unlimited skips **Note:** Amazon Music Unlimited Single Device Plan doesn't have access to HD, Ultra HD, or Spatial Audio.

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GW3PHAUCZM8L7W9L.

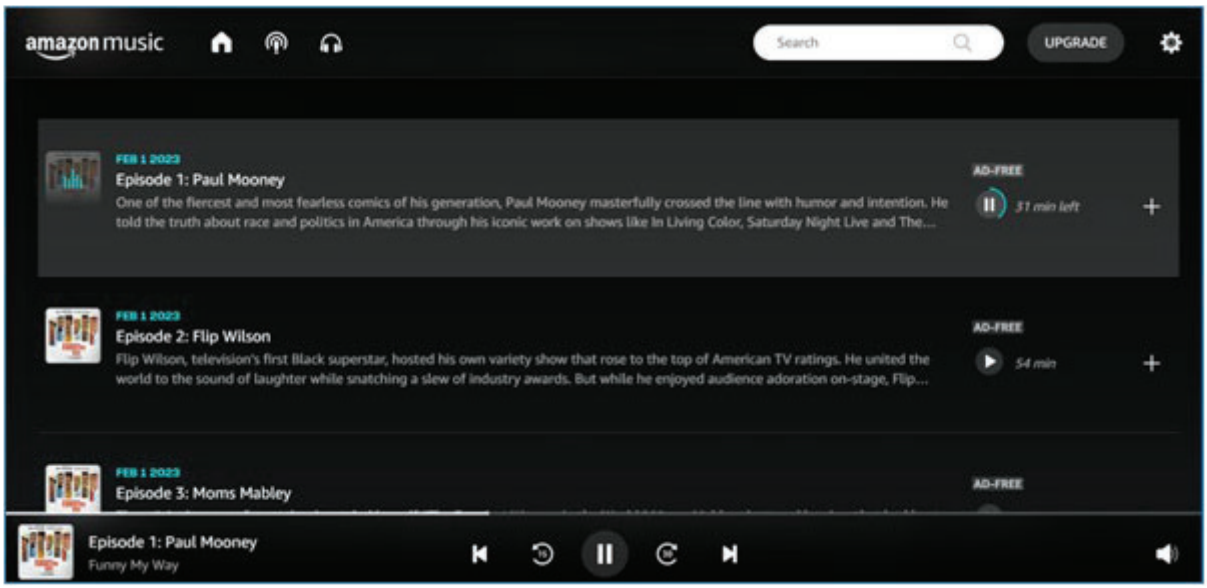
Supported Devices	Supported Echo devices. A growing list of Alexa-enabled devices. iOS Android Fire TV devices Fire Tablet Amazon Music for Web	All Amazon Music supported devices. **Note:** All-Access Playlists requested on Echo devices play in shuffle mode. On Fire TV, you can only listen to music. There are no podcasts available. On Fire Tablet, you don't have a full music catalog or ad-free podcasts availability.	All Amazon Music supported devices (HD, Ultra HD, and Spatial Audio are available on select devices). **Amazon Music Unlimited Single Device Plan**: Echo devices Fire TV devices

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GW3PHAUCZM8L7W9L.

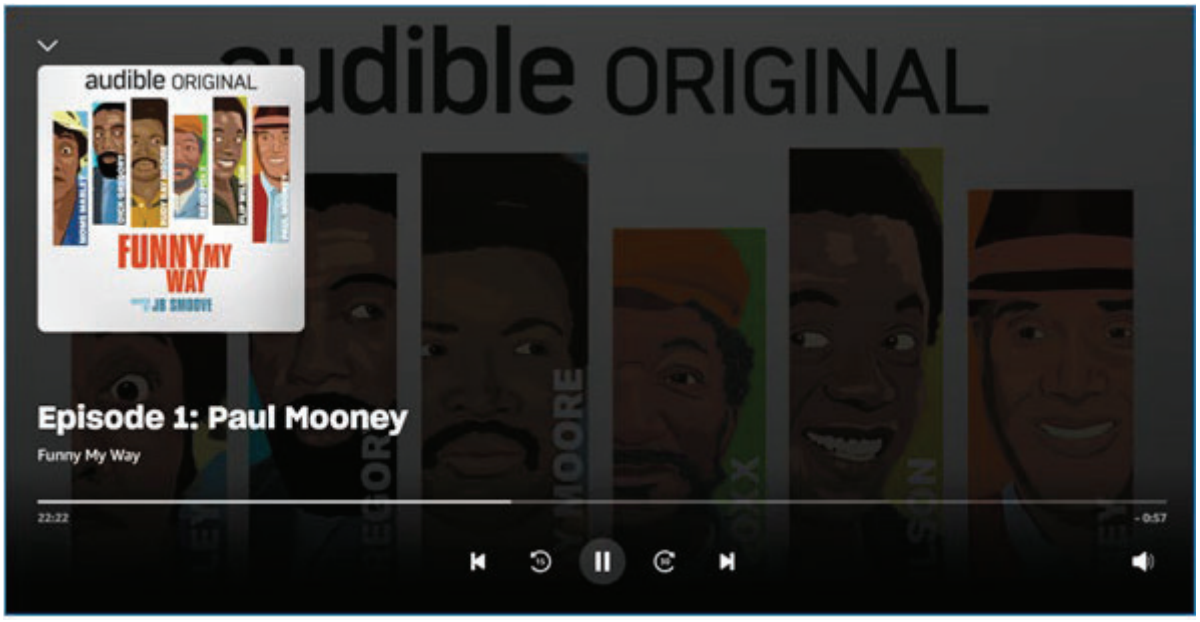
142. On information and belief, one or more components of Prime Music provide a method of rendering (e.g., presenting the media content) digital content across multiple client devices comprising rendering on a first client device (e.g., a desktop, laptop, smartphone, tablet, smart TV, Echo device, Alexa-enabled device, Fire TV) at least a portion (e.g., "chunk") of primary digital content.

> When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

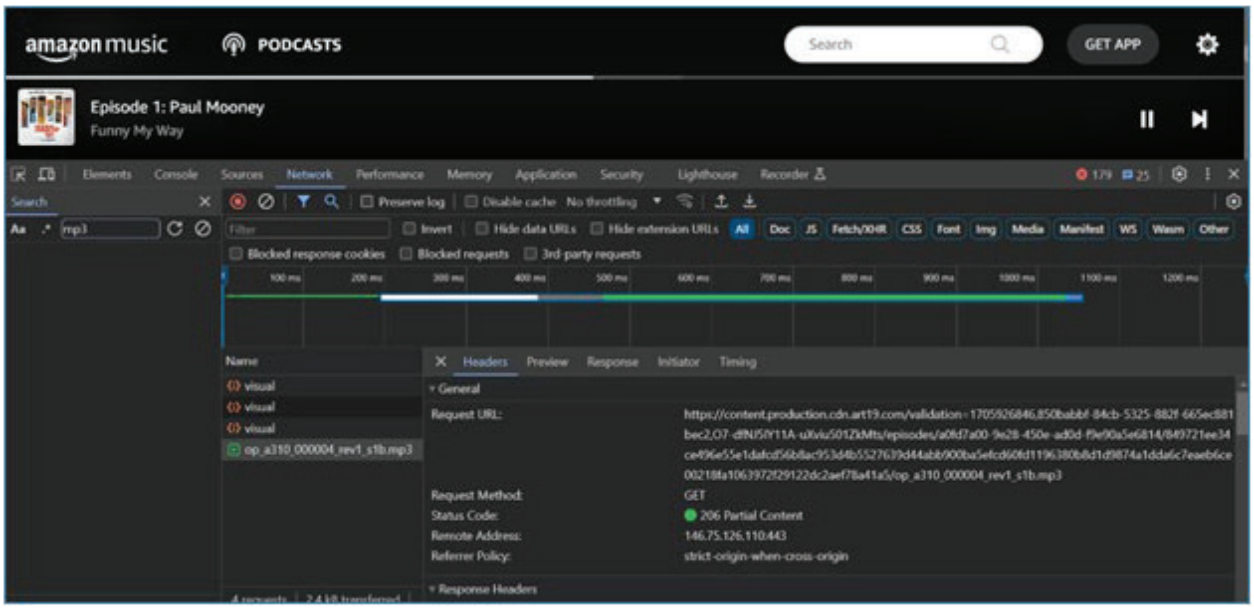
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



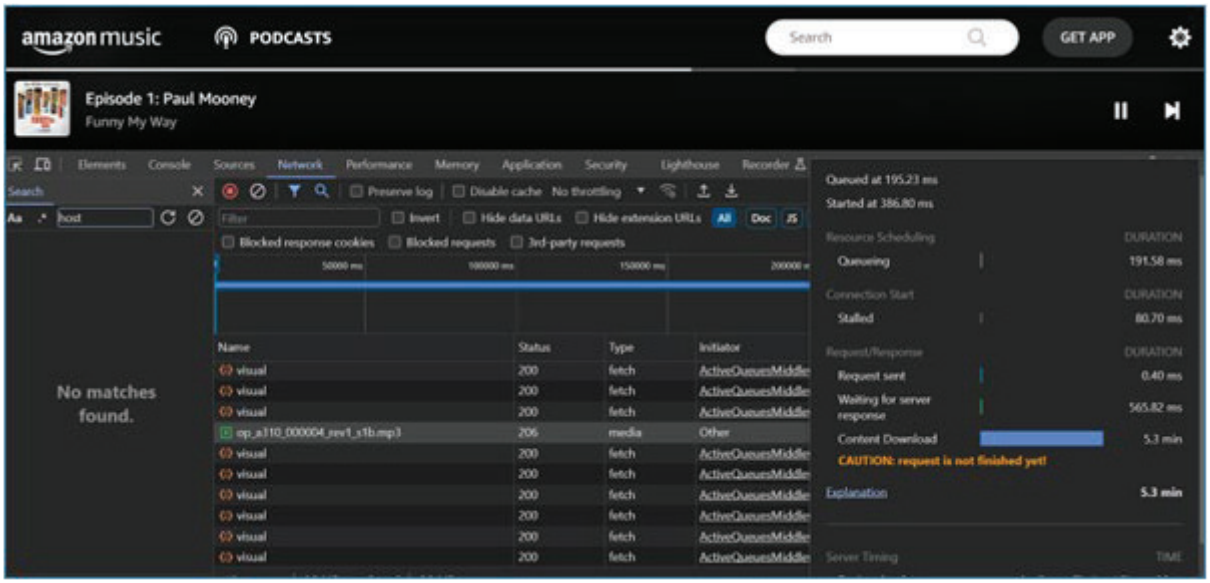
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



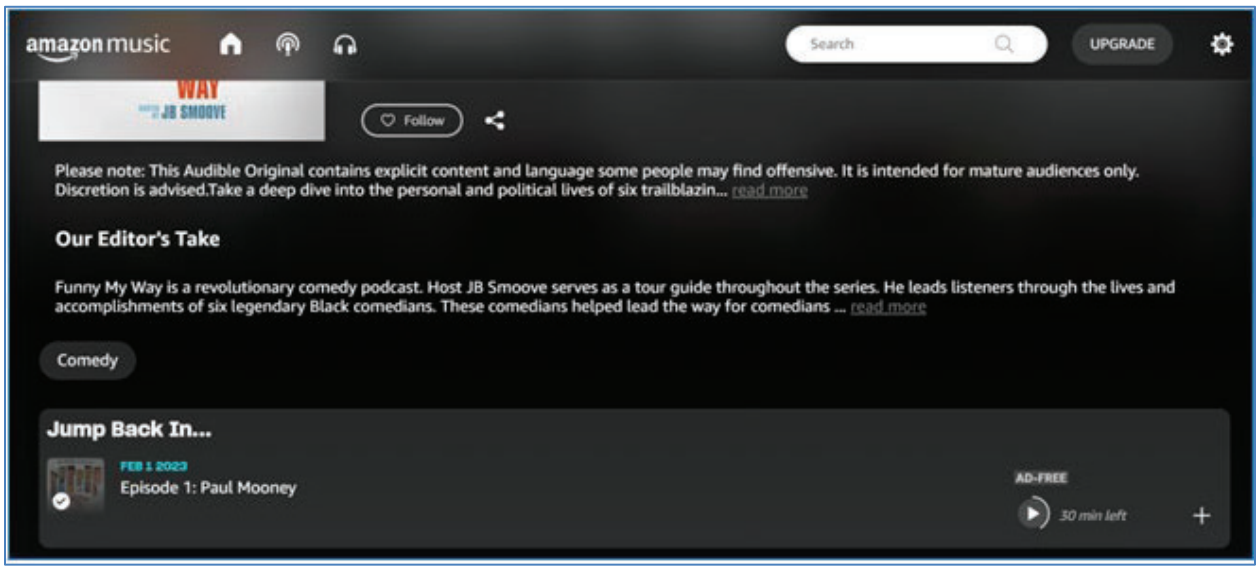
See Screenshot taken during network analysis of Amazon Music (desktop version, music.amazon.com).



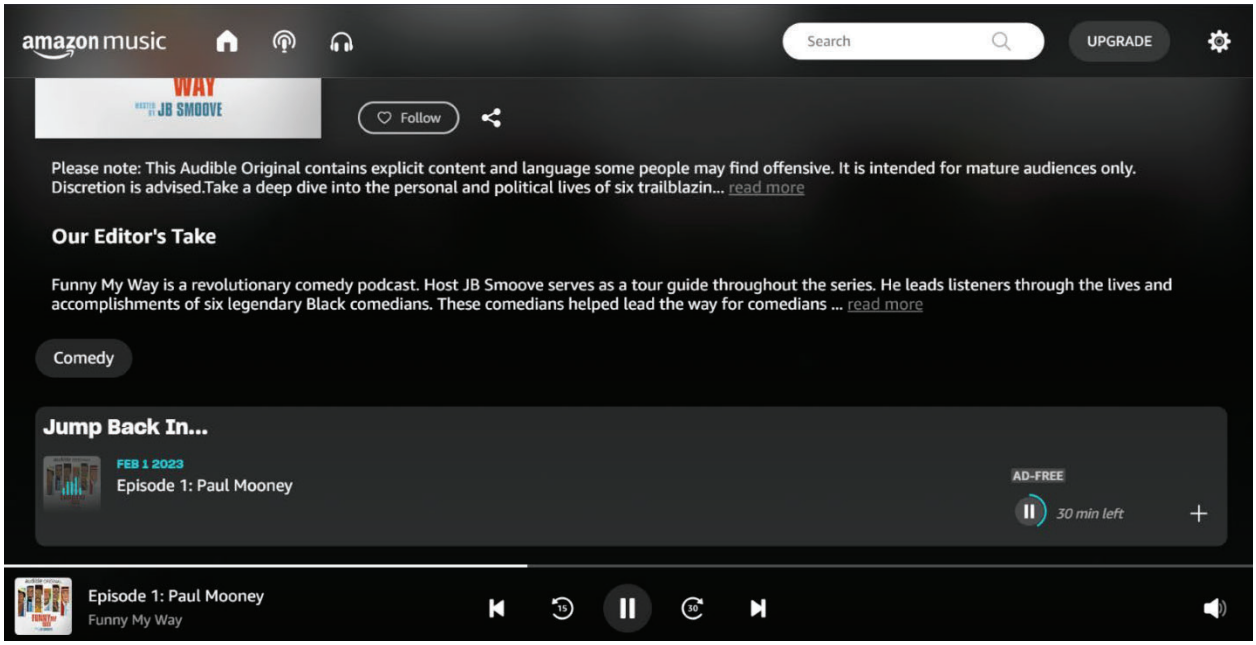
See Screenshot taken during network analysis of Amazon Music (desktop version, music.amazon.com).

143. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising determining on the first client device an identifier (e.g., a data element representing at least a particular item of

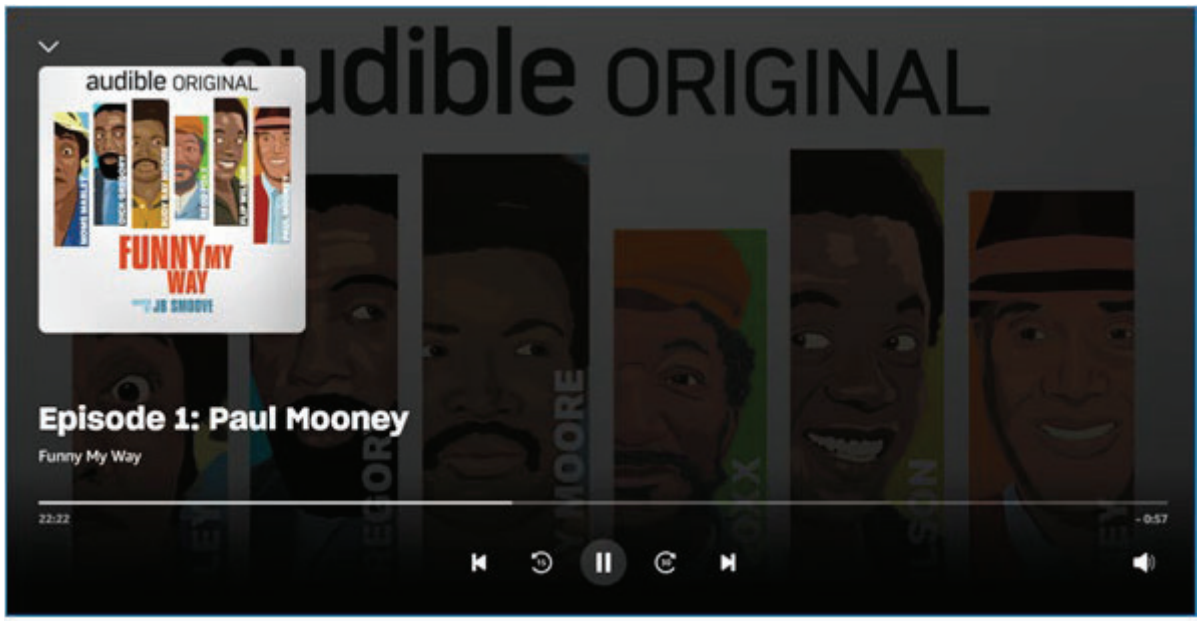
media content (e.g., a movie, show, and/or episode)) corresponding to the primary digital content, wherein the identifier identifies a descriptor (e.g., Media Presentation Description ("MPD")) of the primary content.



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).

Scope of MPEG-DASH

Figure 2 illustrates a simple streaming scenario between an HTTP server and a DASH client. In this figure, the multimedia content is captured and stored on an HTTP server and is

delivered using HTTP. The content exists on the server in two parts: Media Presentation Description (MPD), which describes a manifest of the available content, its various alternatives, their URL addresses, and other characteristics; and segments, which contain the actual multimedia bitstreams in the form of chunks, in single or multiple files.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/T_MM1_TheMPEGDASHStandard.pdf.

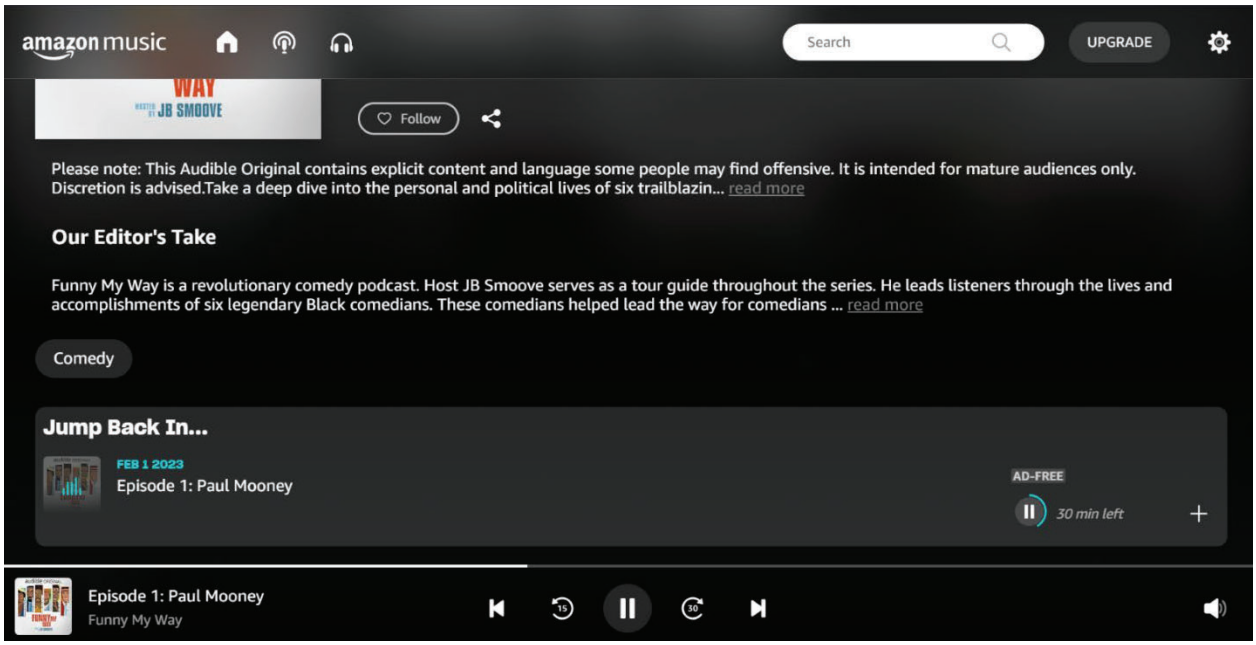
MPEG-DASH: The Standard for Multimedia Streaming Over Internet

In order to play the content, the DASH client first obtains the MPD. The MPD can be delivered using HTTP, email, thumb drive, broadcast or other transports. By parsing the MPD, the DASH client learns about the timing of the program, the availability of media content, the media types, resolutions, minimum and maximum bandwidths and the existence of various encoded alternatives of multimedia components, the accessibility features and the required digital right management (DRM), the location of each media component on the network and other characteristic of the content. Using this information, the DASH client selects the appropriate encoded alternative and starts streaming of the content by fetching the segments using HTTP GET requests.

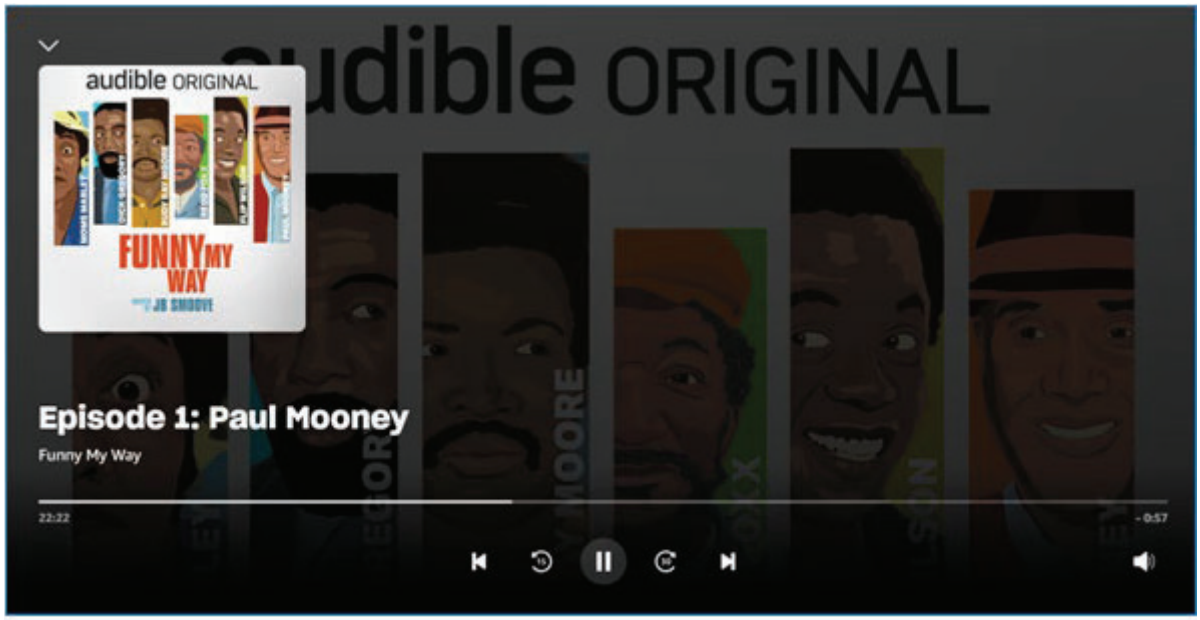
Dynamic HTTP streaming requires various bitrate alternatives of the multimedia content to be available at the server. In addition, the multimedia content may consist of several media components (e.g. audio, video, text), each of which may have different characteristics. In MPEG-DASH, these characteristics are described by MPD which is an XML document.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf.

144. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising determining on the first client device a first position (e.g., creating a data component that includes timing information to indicate the stop/pause position for the media stream, and/or a progress bar or cursor) in the primary digital content.



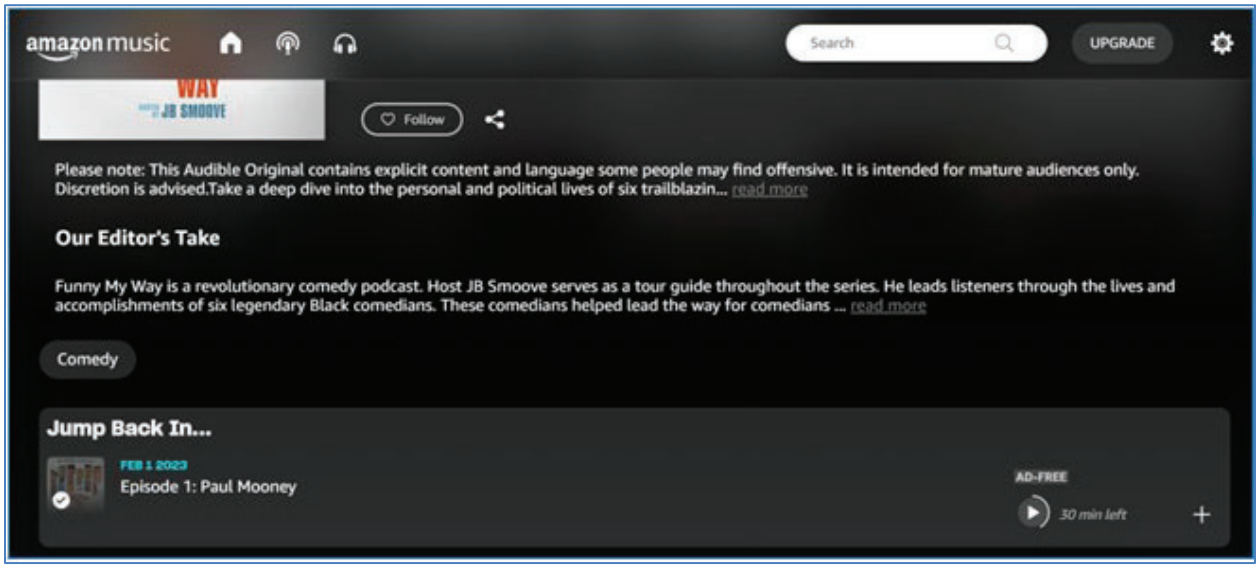
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



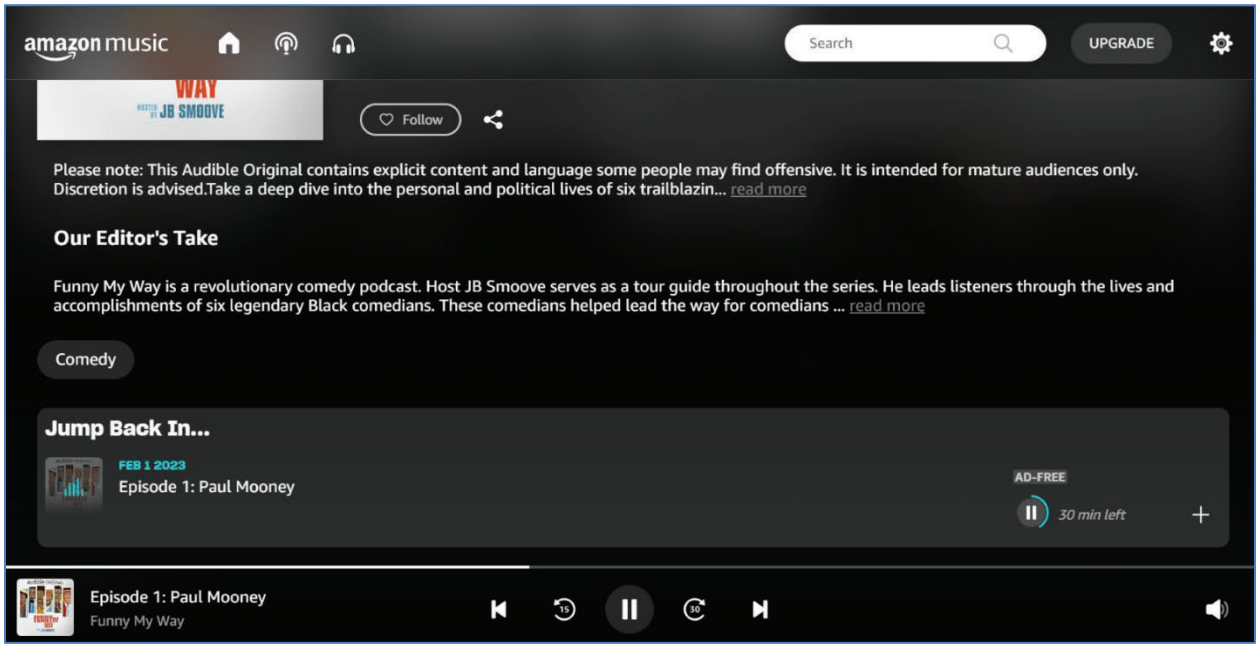
See Screenshot taken during product.

145. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising transferring the identifier and the first position from the first client device to a second client device via a network

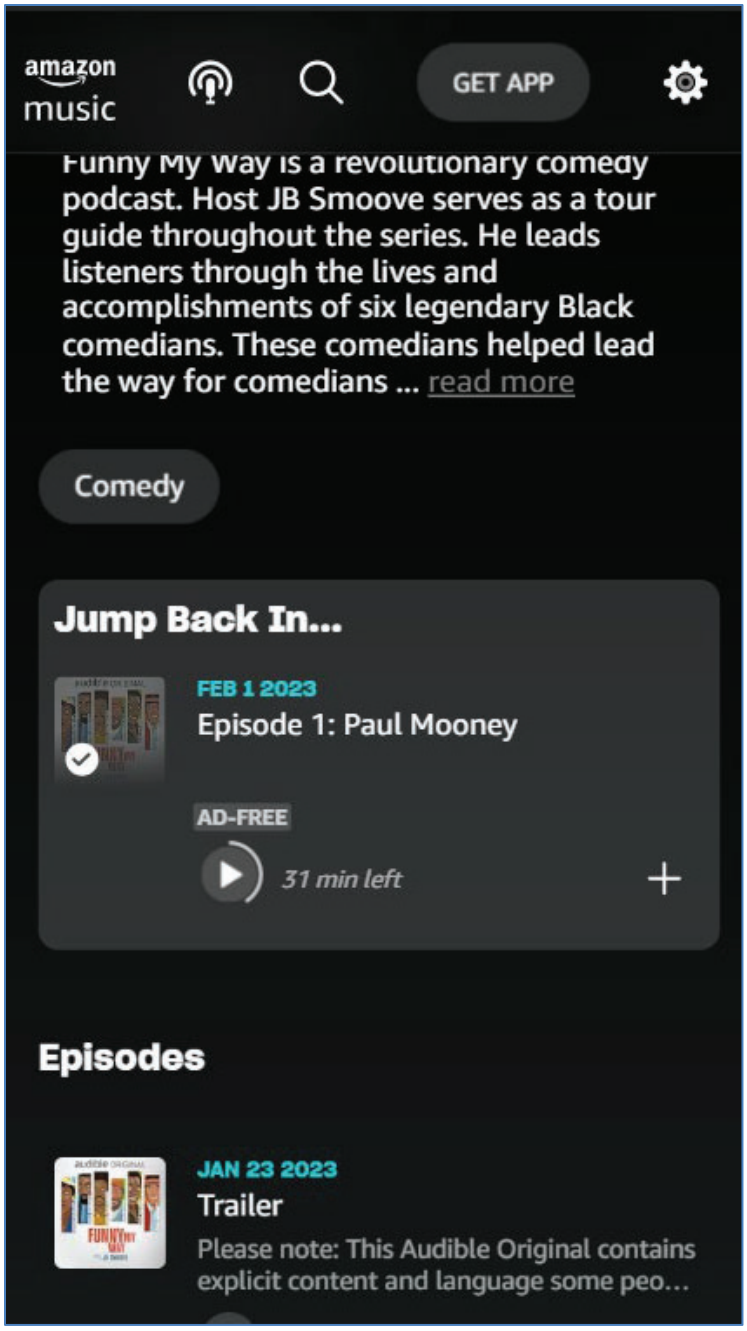
accessible library (e.g., the "Jump Back In" feature allows a second device accessing Prime Music application or website to begin listening to an audio stream where a first device left off).



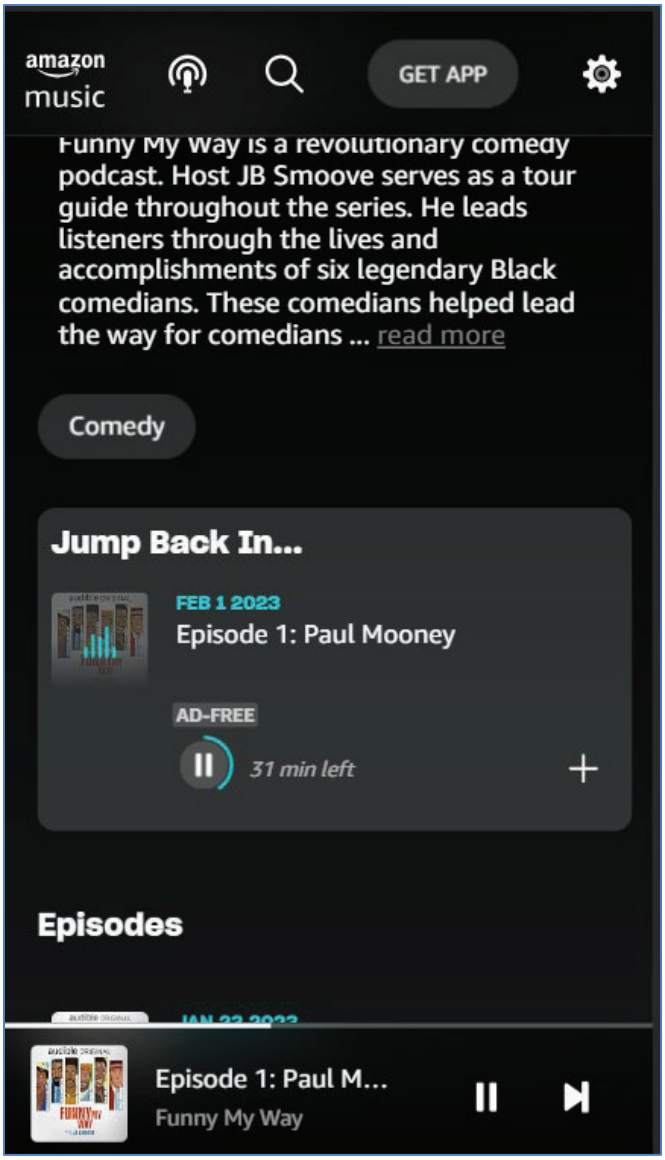
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com.

146. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising downloading the descriptor from the network accessible library to the second client device by using the identifier (e.g., through the "Jump Back In" feature).

Streaming services, such as Amazon Music, have revolutionized the way we listen to music. Unlike traditional methods of purchasing and downloading individual songs or albums, streaming platforms allow users to access an extensive catalog of music for a monthly subscription fee or through free ad-supported options. This vast library of songs is stored on remote servers, and users can stream them directly to their devices over the internet.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

See https://developer.amazon.com/docs/music/playback_overview.html.

Caching

A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

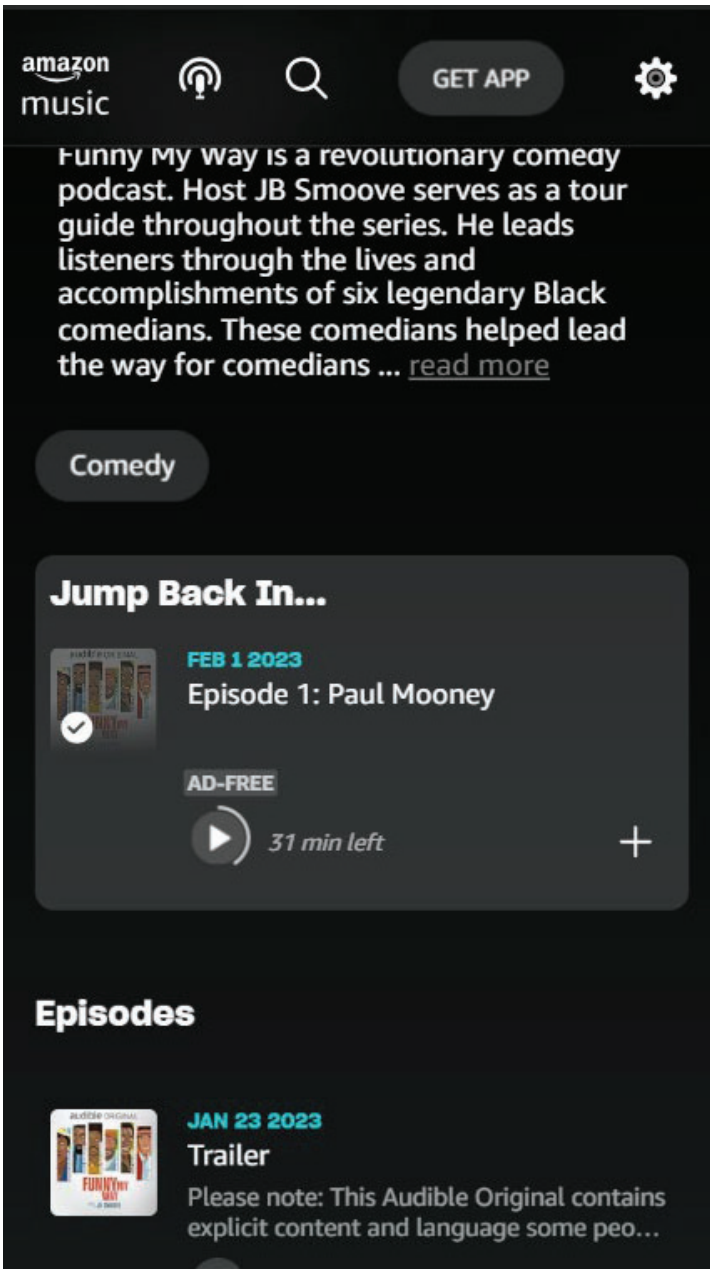
See https://developer.amazon.com/docs/music/API_browse_caching.html.

> Streaming services like Amazon Music rely on internet connectivity to deliver music to your devices. Each time you play a song, it is streamed from Amazon's servers to your device in real-time. This constant streaming requires a significant amount of data, which can quickly add up, especially for those with limited data plans or slower internet connections.

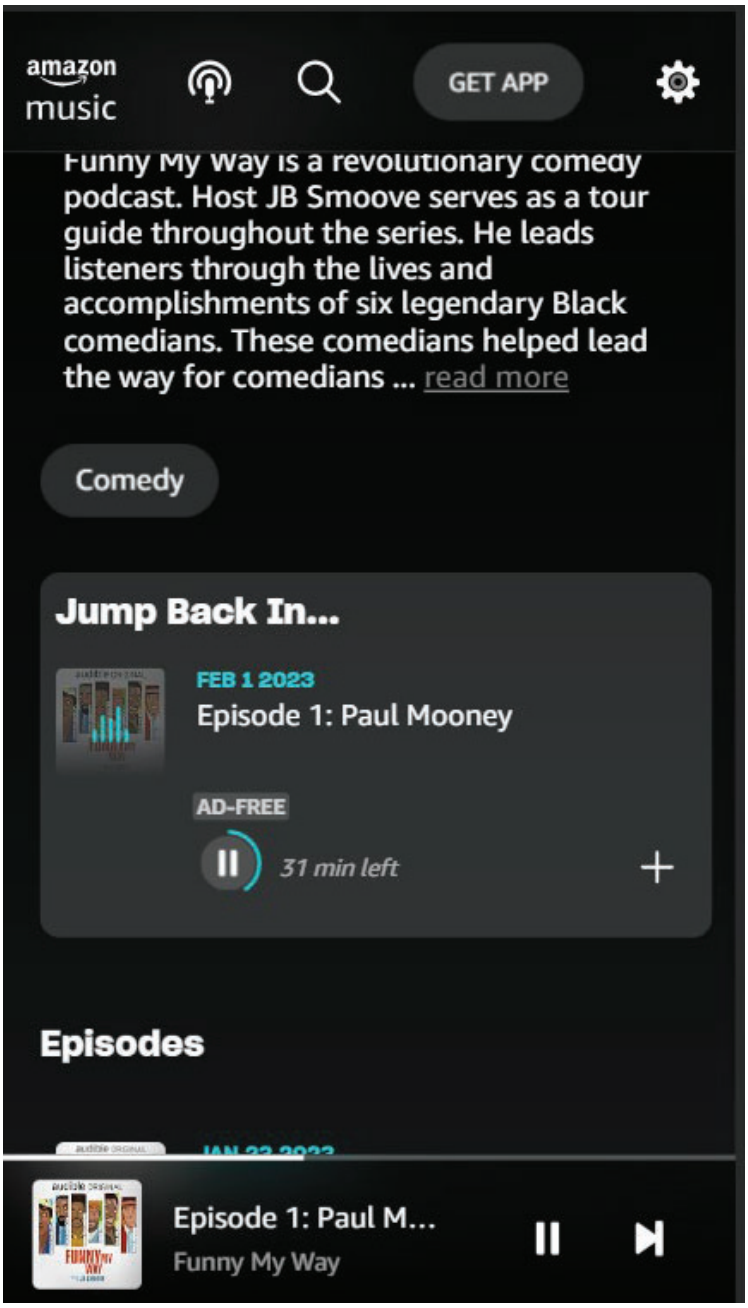
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette

> To stream music without issues, you need a strong Internet connection. To troubleshoot streaming issues:
>
> - Confirm that your device is connected to Wi-Fi or a mobile network.
> - If using a mobile network, confirm that the Amazon Music app settings allow for Cellular.
> - Force stop and reopen the app.
>
> To stream HD and Ultra HD music with Amazon Music Unlimited, you need a strong Internet connection. To troubleshoot streaming issues:

See https://www.amazon.com/gp/help/customer/display.html?nodeId=GPTQFUDBXQY85W6M.



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



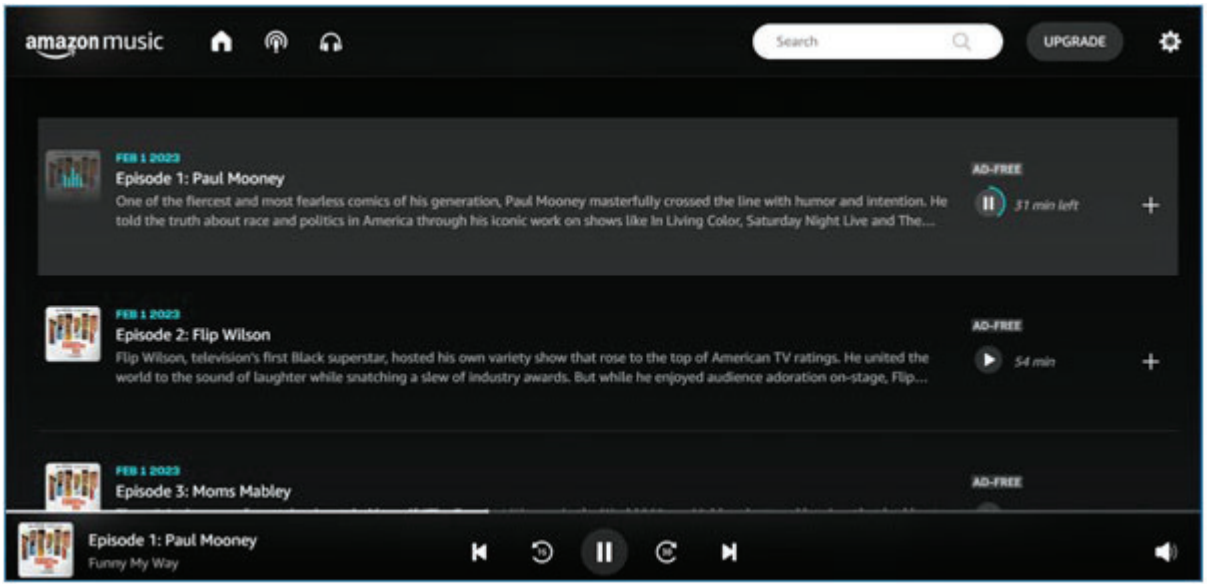
See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

147. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising rendering on the second client device at least a portion (e.g., "chunk") of secondary other digital content associated with the primary digital content (e.g., subtitle streams corresponding to a media work

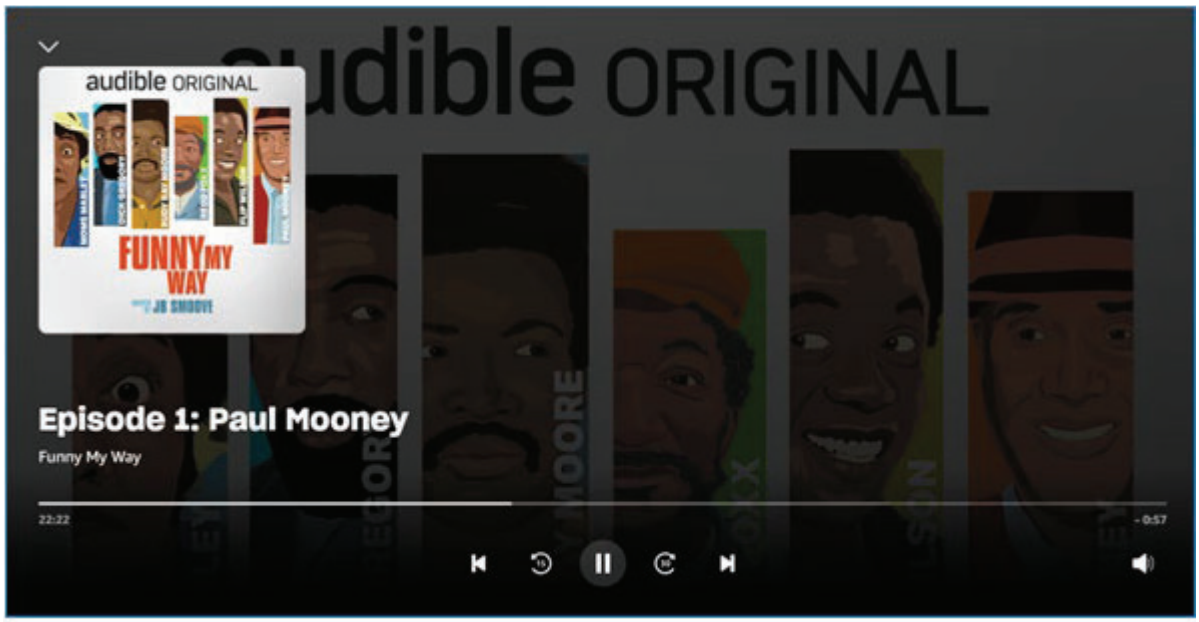
(e.g., a song or audio work)) by using the descriptor and the first position, wherein the secondary digital content is ancillary to the primary digital content (e.g., subtitle streams are ancillary compared to the song or audio work), and wherein the secondary digital content is rendered on the second client device simultaneously and in synchronization with the rendering of the primary digital content on the first client device (e.g., through the "Jump Back In" feature).

> When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

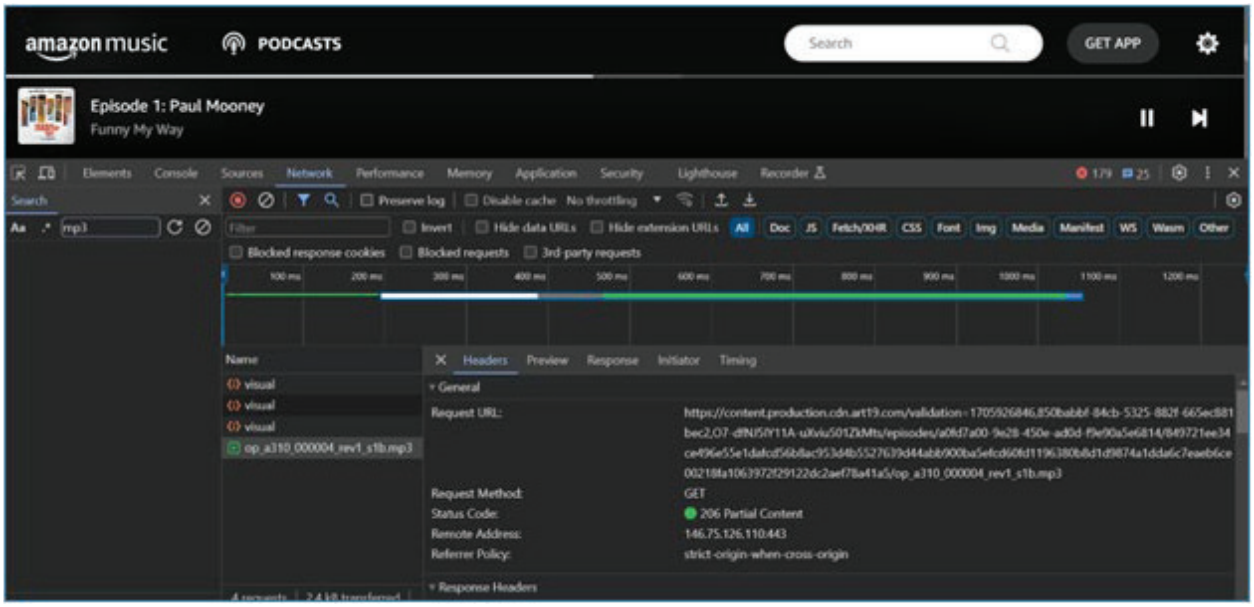
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



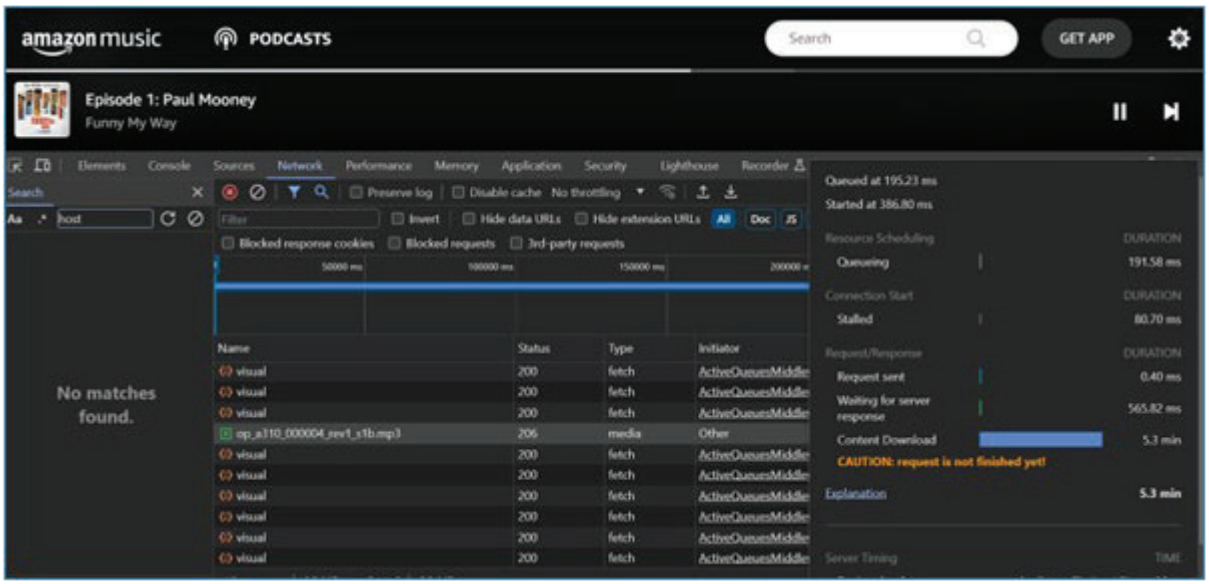
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



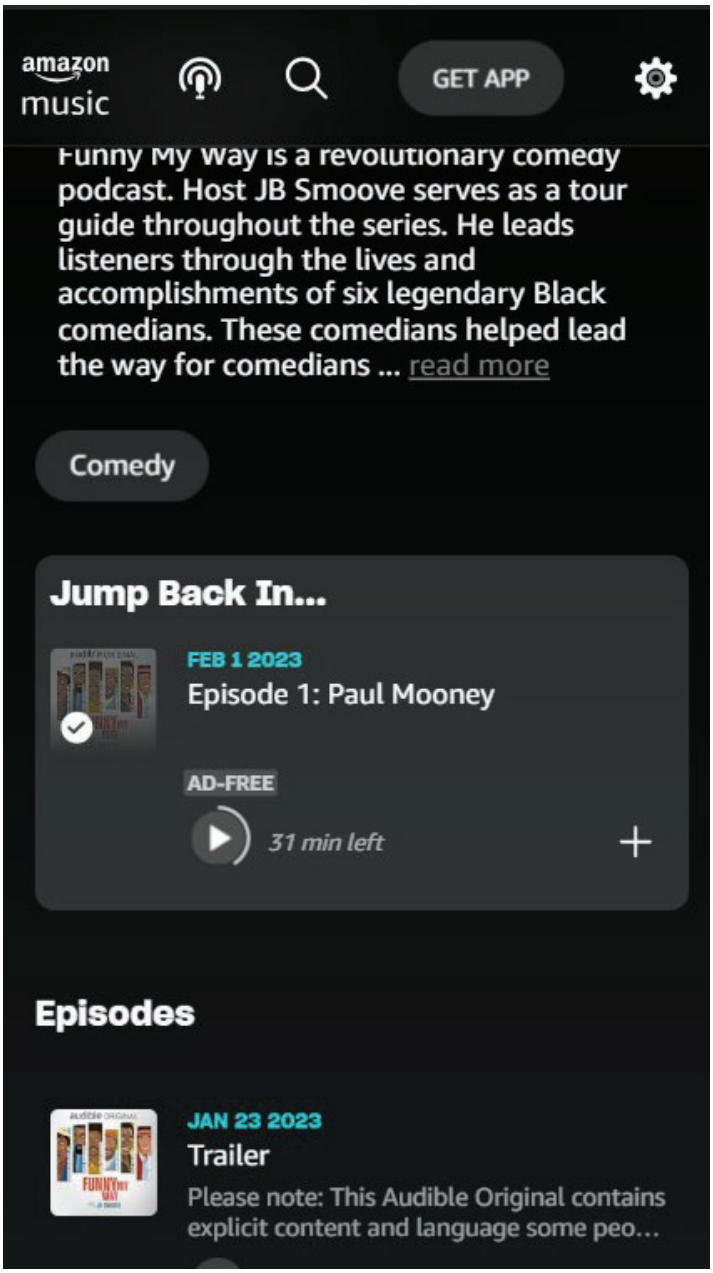
See Screenshot taken during product testing of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during network analysis of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during network analysis of Amazon Music (desktop version, music.amazon.com).



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

4.3 DASH data model overview

The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent or certain periods of a Media Presentation may belong to the same Asset, for example a Media Presentation is a collection of main program composed of multiple periods, each assigned to the same Asset, and interleaved with inserted advertisement periods. Each media content period is composed of one or multiple **media content components,** for example audio components in various languages, different video components providing different views of the same program, subtitles in different language, etc.. Each media content component has an assigned **media content component type**, for example audio or video.

DASH is based on a hierarchical data model aligned with the presentation in Figure 3. A DASH Media Presentation is described by a **Media Presentation Description** document. This describes the sequence of **Periods** (see 5.3.2) in time that make up the Media Presentation. A Period typically represents a media content period during which a consistent set of encoded versions of the media content is available i.e. the set of available bitrates, languages, captions, subtitles etc. does not change during a Period.

NOTE This is not strictly true, since the MPD may also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client could in principle construct a single request for multiple Segments, but this would not be the typical case.

DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different media components and to enable seamless switching of different coded versions of the same media components. This timeline is referred as Media Presentation timeline. The Media Segments themselves contain accurate Media Presentation timing information enabling synchronization of components and seamless switching.

See https://github.com/liwf616/awesome-live-stream/raw/master/Ebook/ISO_IEC 23009-1_2014.pdf.

148. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising identifying a range of content surrounding the first position in the primary digital content as content to be retained (e.g., adding new data chunks to a buffered section of the audio stream, and/or the ability go forward and backward from a stop/pause/progress bar/cursor position).

> In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

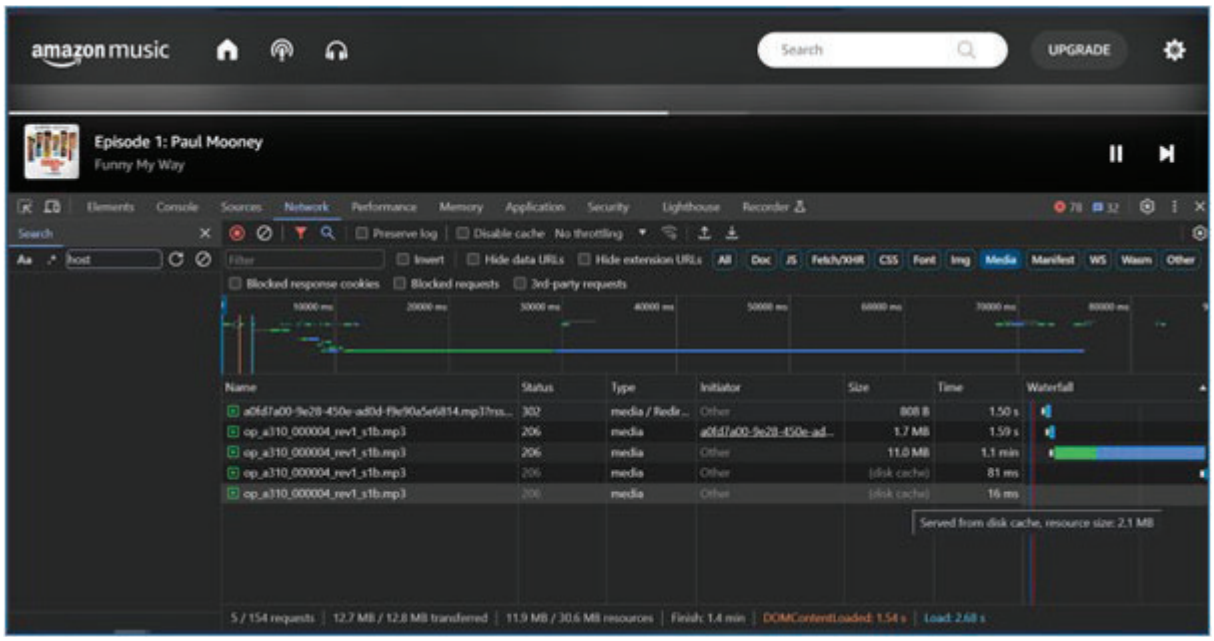
See https://developer.amazon.com/docs/music/playback_overview.html

> When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



See Screenshot taken during network analysis of Amazon Music.

149. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising releasing storage resources allocated to all content of the primary digital content that is not identified as content to be retained on the first client device (e.g., clearing the cache, and/or automatically "discard[ing]" previously played data).

Caching

A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.

The disk cache stores resources fetched from the web so that they can be accessed quickly at a latter time if needed. The main characteristics of Chromium disk cache are:

- The cache should not grow unbounded so there must be an algorithm for deciding when to remove old entries.
- While it is not critical to lose some data from the cache, having to discard the whole cache should be minimized. The current design should be able to gracefully handle application crashes, no matter what is going on at that time, only discarding the resources that were open at that time. However, if the whole computer crashes while we are updating the cache, everything on the cache probably will be discarded.
- Access to previously stored data should be reasonably efficient, and it should be possible to use synchronous or asynchronous operations.
- We should be able to avoid conflicts that prevent us from storing two given resources simultaneously. In other words, the design should avoid cache trashing.
- It should be possible to remove a given entry from the cache, and keep working with a given entry while at the same time making it inaccessible to other requests (as if it was never stored).
- The cache should not be using explicit multithread synchronization because it will always be called from the same thread. However, callbacks should avoid reentrancy problems so they must be issued through the thread's message loop.

See https://www.chromium.org/developers/design-documents/network-stack/disk-cache/.

150. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising identifying content in the secondary digital content that is related to the range of content surrounding the first position in the primary digital content as content to be retained (e.g., through the "Jump Back In" feature).

In order to support varying network conditions, streaming files are broken into short segments (sometimes called 'chunks'). Multiple versions of each chunk are provided in several different levels of quality, bit-rates, and codecs. To keep the experience seamless, lower quality chunks can be requested for faster download. Once bandwidth improves, higher quality chunks can be requested once more. These are all packaged together in a Manifest file, which is an XML file defined by the MPEG-DASH format (.mpd). Amazon Music also offers media in a variety of high-definition and ultra-high-definition streaming formats. Segments in these higher-quality formats will be contained within the Manifest if requested by the client.

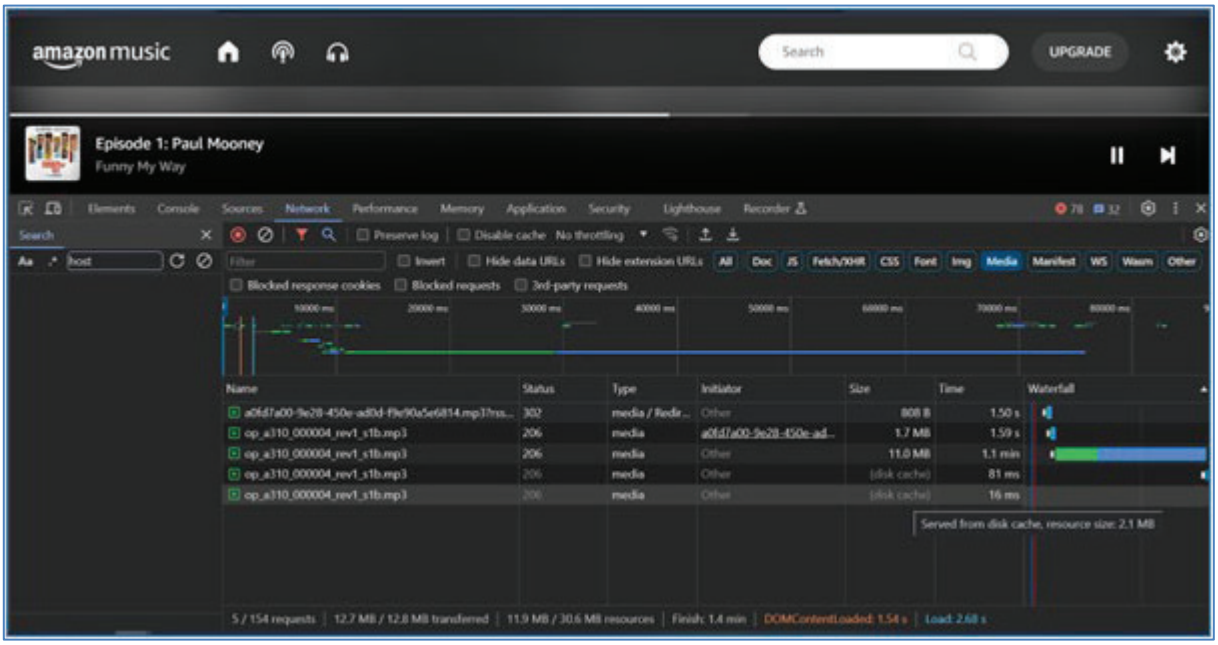
See https://developer.amazon.com/docs/music/playback_overview.html.

When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

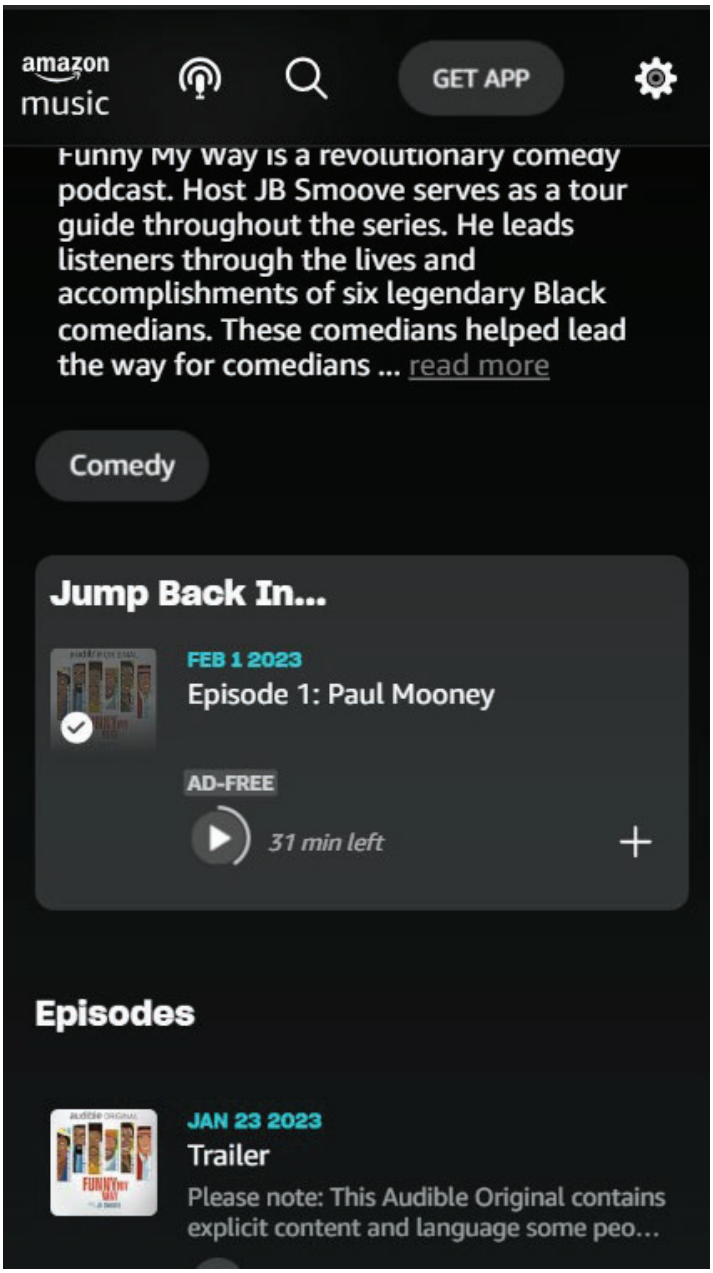
See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).



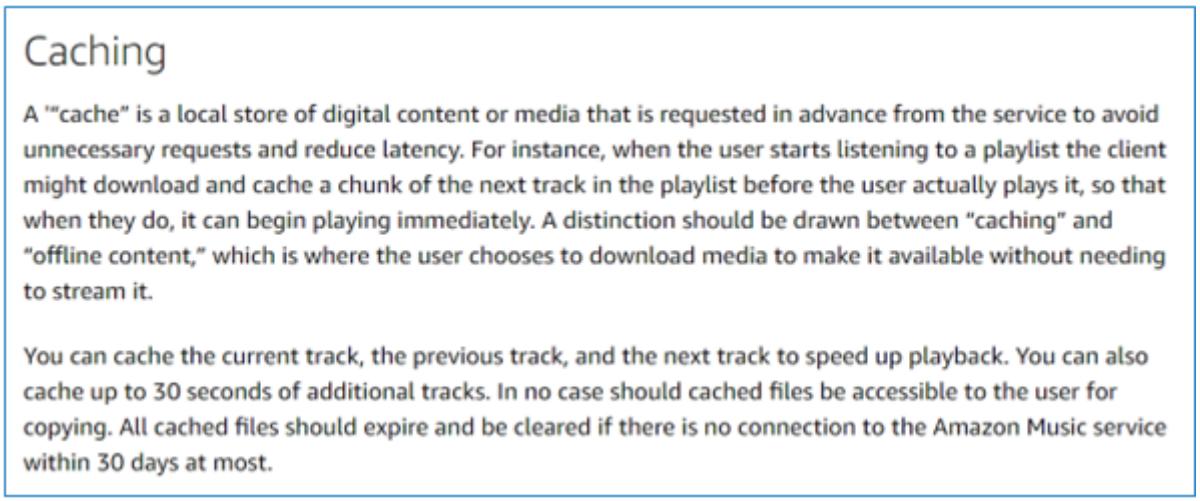
See Screenshot taken during network analysis of Amazon Music.



See Screenshot taken during product testing of Amazon Music (mobile version, music.amazon.com).

151. On information and belief, one or more components of Prime Music provide a method of rendering digital content across multiple client devices comprising releasing storage resources allocated to all content of the secondary digital content that is not identified as content

to be retained on the second client device (e.g., clearing the cache, and/or automatically "discard[ing]" previously played data).

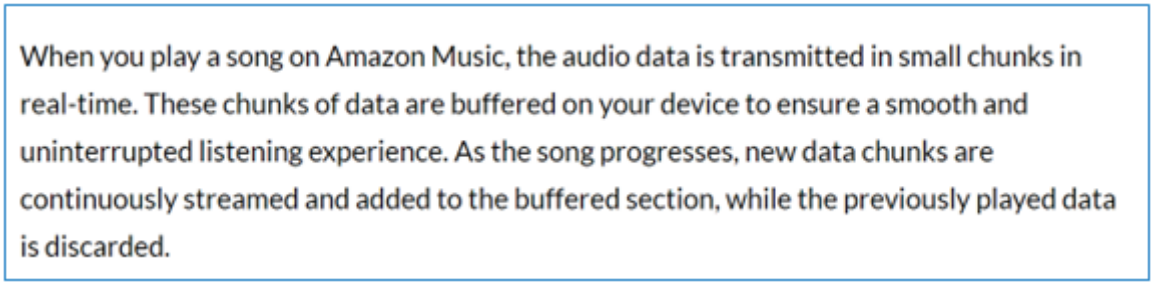


Caching

A '"cache" is a local store of digital content or media that is requested in advance from the service to avoid unnecessary requests and reduce latency. For instance, when the user starts listening to a playlist the client might download and cache a chunk of the next track in the playlist before the user actually plays it, so that when they do, it can begin playing immediately. A distinction should be drawn between "caching" and "offline content," which is where the user chooses to download media to make it available without needing to stream it.

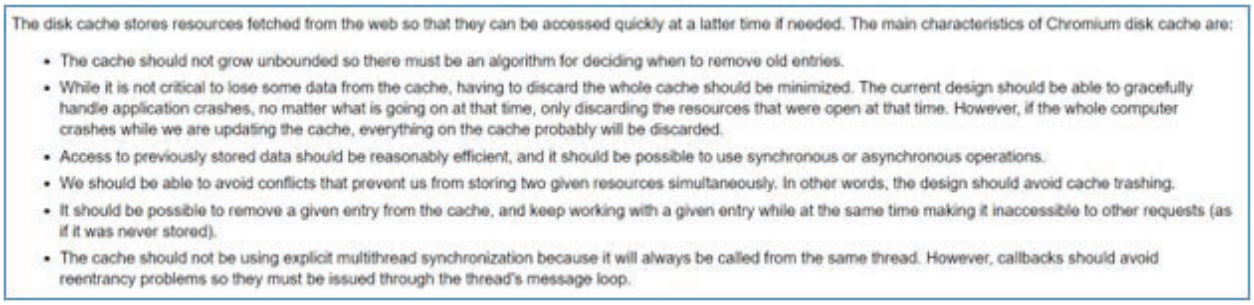
You can cache the current track, the previous track, and the next track to speed up playback. You can also cache up to 30 seconds of additional tracks. In no case should cached files be accessible to the user for copying. All cached files should expire and be cleared if there is no connection to the Amazon Music service within 30 days at most.

See https://developer.amazon.com/docs/music/API_browse_caching.html.



When you play a song on Amazon Music, the audio data is transmitted in small chunks in real-time. These chunks of data are buffered on your device to ensure a smooth and uninterrupted listening experience. As the song progresses, new data chunks are continuously streamed and added to the buffered section, while the previously played data is discarded.

See https://robots.net/tech/why-is-amazon-music-using-so-much-data/#google_vignette.



The disk cache stores resources fetched from the web so that they can be accessed quickly at a latter time if needed. The main characteristics of Chromium disk cache are:

- The cache should not grow unbounded so there must be an algorithm for deciding when to remove old entries.
- While it is not critical to lose some data from the cache, having to discard the whole cache should be minimized. The current design should be able to gracefully handle application crashes, no matter what is going on at that time, only discarding the resources that were open at that time. However, if the whole computer crashes while we are updating the cache, everything on the cache probably will be discarded.
- Access to previously stored data should be reasonably efficient, and it should be possible to use synchronous or asynchronous operations.
- We should be able to avoid conflicts that prevent us from storing two given resources simultaneously. In other words, the design should avoid cache trashing.
- It should be possible to remove a given entry from the cache, and keep working with a given entry while at the same time making it inaccessible to other requests (as if it was never stored).
- The cache should not be using explicit multithread synchronization because it will always be called from the same thread. However, callbacks should avoid reentrancy problems so they must be issued through the thread's message loop.

See https://www.chromium.org/developers/design-documents/network-stack/disk-cache/.

152. On information and belief, Defendants directly infringe at least claim 1 of the ’266 patent in violation of 35 U.S.C. § 271(a) by making, using, selling, selling access to, importing, offering for sale, and/or offering to sell access to Prime Music.

153. Defendants’ infringement has damaged Audio Pod and caused/continues to cause it to suffer irreparable harm and damages.

JURY DEMANDED

154. Pursuant to Federal Rule of Civil Procedure 38(b), Audio Pod hereby requests a trial by jury on all issues so triable.

PRAYER FOR RELIEF

Audio Pod respectfully requests this Court to enter judgment in Audio Pod’s favor and against Amazon as follows:

a. finding that Amazon has infringed one or more claims of the ’720 patent under 35 U.S.C. § 271(a);

b. finding that Amazon has infringed one or more claims of the ’922 patent under 35 U.S.C. § 271(a);

c. finding that Amazon has infringed one or more claims of the ’266 patent under 35 U.S.C. § 271(a);

d. finding that Amazon has infringed one or more claims of the ’488 patent under 35 U.S.C. § 271(a);

e. awarding Audio Pod damages under 35 U.S.C. § 284, or otherwise permitted by law, including enhanced damages for willful infringement and/or supplemental damages for any continued post-verdict infringement;

f. awarding Audio Pod pre-judgment and post-judgment interest on the damages award and costs;

g. awarding cost of this action (including all disbursements) and attorney fees pursuant to 35 U.S.C. § 285, or as otherwise permitted by the law; and

h. awarding such other costs and further relief that the Court determines to be just and equitable.

Dated: May 30, 2024

Respectfully submitted,

/s/ Chandran B. Iyer

Ronald M. Daignault (pro hac vice to be filed)*
Chandran B. Iyer (VA Bar No. 94100)
Kevin H. Sprenger (VA Bar No. 98588)
rdaignault@daignaultiyer.com
cbiyer@daignaultiyer.com
ksprenger@daignaultiyer.com
DAIGNAULT IYER LLP
8618 Westwood Center Drive - Suite 150
Vienna, VA 22182
Tel.: (202) 330-1666

Attorneys for Plaintiff Audio Pod IP LLC

**Not admitted to practice in Virginia*