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, AUDIBLE, INC., AND AMAZON WEB SERVICES, INC., Defendants.	Case No. 1:24-cv-00914 Jury Trial Demanded

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Audio Pod IP, LLC ("Audio Pod") files this complaint against Amazon.com, Inc., Amazon.com LLC, Audible, Inc., and Amazon Web Services, Inc. (hereinafter collectively "Amazon" or "Defendants") for infringement of United States Patent No. 9,729,907 (the "Patent-in-Suit"), attached here as Exhibit 1. Audio Pod alleges that Defendants have willfully and/or otherwise infringed the Patent-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.

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 Audible, Inc. is a corporation organized and existing under the laws of the state of Delaware with a principal place of business at One Washington Park, 16th Floor, Newark, New Jersey 07102.

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

11. On information and belief, Audible, 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.

12. Audible, Inc. 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, 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.

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

15. 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.

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

17. 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.

18. 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.

19. 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 CloudFront[1]; (2) content streaming services using Amazon CloudFront Content Delivery Network such as Amazon Audible[2], Amazon Video and Amazon Prime Video[3] (collectively "Prime Video"), and Amazon Music, Amazon Music Prime and Amazon Prime Music[4] (collectively "Prime Music"); (3) devices using Amazon CloudFront Delivery Network software including but not limited to Amazon Echo Show Products[5], Amazon Fire TV Products[6] Amazon tablet products[7] and Amazon Kindle E-Readers[8]; (4) 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; (5) combinations of products and/or services comprising, in whole or in part, two or more of the foregoing products and services; and (6) 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

[1] *See* https://aws.amazon.com/cloudfront/.

[2] See https://www.audible.com/.

[3] *See* https://www.amazon.com/gp/help/customer/display.html?nodeId=GXY5VQJFHE9VYMCK.

[4] *See* https://www.amazon.com/gp/help/customer/display.html?nodeId=GW3PHAUCZM8L7W9L, https://www.amazon.com/gp/help/customer/display.html?ref_=hp_gcs_csd_d2_327_134_GKL9M88FFWYHY7T9_bcxlTFf2fO&nodeId=GKL9M88FFWYHY7T9&qid=1716464872791&sr=134, and https://en.wikipedia.org/wiki/Amazon_Music.

[5] *See* https://en.wikipedia.org/wiki/Amazon_Echo_Show.

[6] *See* https://en.wikipedia.org/wiki/Amazon_Fire_TV#Models.

[7] *See* https://en.wikipedia.org/wiki/Amazon_Fire; *see also* https://s3-us-west-2.amazonaws.com/customerdocumentation/Kindle_Fire_1st_Gen_Help/Kindle_Fire_1st_Generation_Prime_Video_PDF.pdf.

[8] *See* https://www.amazon.com/gp/help/customer/display.html?nodeId=GK33S847NN4V6Y83&ref_=hp_help_Identify-Your-Kindle-E-Reader.

are individually and collectively referred to as the accused "Amazon CloudFront Products and Services").

20. On information and belief, Amazon, as well as the hardware and software components comprising the Amazon CloudFront Products and Services and/or that enable the Amazon CloudFront 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

21. 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).

22. 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 CloudFront Products and Services provided to individuals and businesses in the Eastern District of Virginia.

23. 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 CloudFront Products and Services in this district.

24. 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 Kindle, Audible, Amazon Prime Video and Amazon Music services. Amazon also produces retail consumer electronics including the Amazon Echo and Amazon Alexa devices.

25. 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.[9]

26. 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, DC metro area.[10] 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.*

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

[10] See https://www.aboutamazon.com/workplace/corporate-offices.

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

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

29. 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 CloudFront Products and Services in this Commonwealth and the Eastern District of Virginia.

30. 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).

31. 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.

32. 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 PATENT

United States Patent No. 9,729,907

33. On August 8, 2017, the USPTO duly and legally issued United States Patent No. 9,729,907 ("the '907 patent") entitled "Synchronizing a Plurality of Digital Media Streams by Using a Descriptor File" to inventors John McCue, Robert McCue, Gregory Shostakovsky, and Glenn McCue.

34. The '907 patent is presumed valid under 35 U.S.C. § 282.

35. Audio Pod owns all rights, title, and interest in the '907 patent.

36. Audio Pod has not granted Defendants an approval, an authorization, or a license to the rights under the '907 patent.

37. The '907 patent relates to, among other things, the synchronized rendering of differential but related media streams across multiple devices using a descriptor file.

38. The descriptor file of the '907 patent includes "synchronization time offsets and the corresponding synchronization points are stored in the descriptor file in a manner indicating a correlation therebetween, such that the descriptor file allows a synchronized rendering of the plurality of digital media streams on a client device." '907 patent, Abstract.

39. The ’907 patent solves, among other things, problems with, and improves upon, the synchronization of a plurality of digital media streams. Further, it allows the consumer to access synchronized digital media streams from multiple client devices as needed.

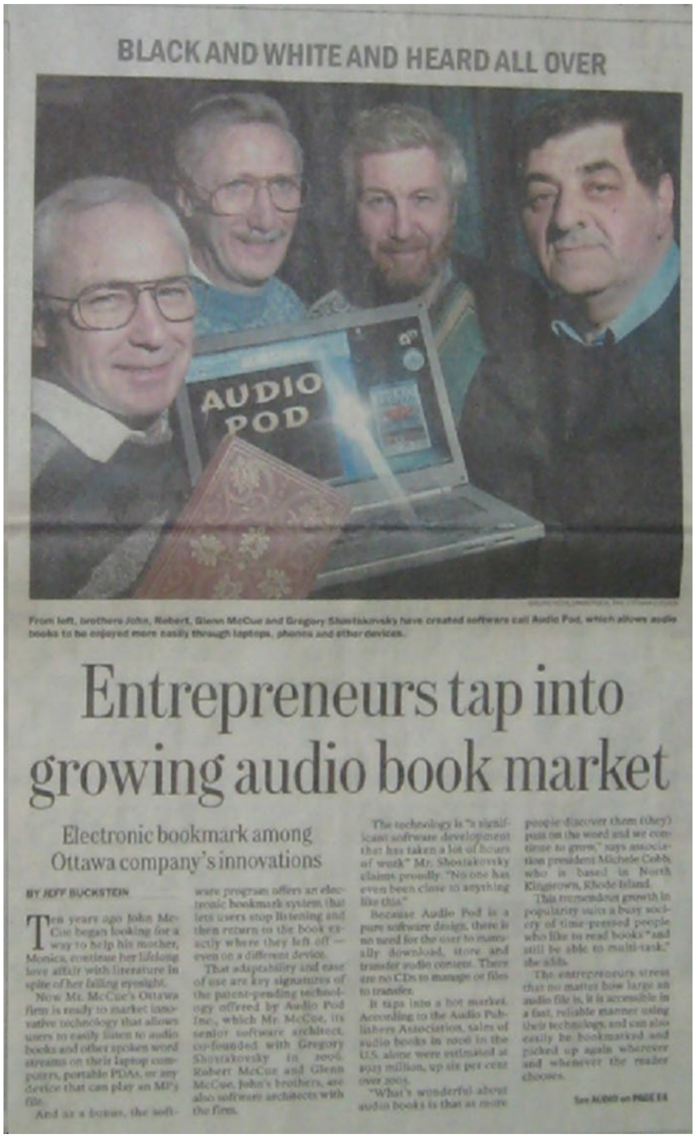
BACKGROUND OF THE INVENTIONS

40. 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.

41. 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).

42. 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.

43. 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 $923 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 2.

CONTACT WITH AMAZON

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

45. 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.

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

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

48. 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.

49. 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.

50. 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

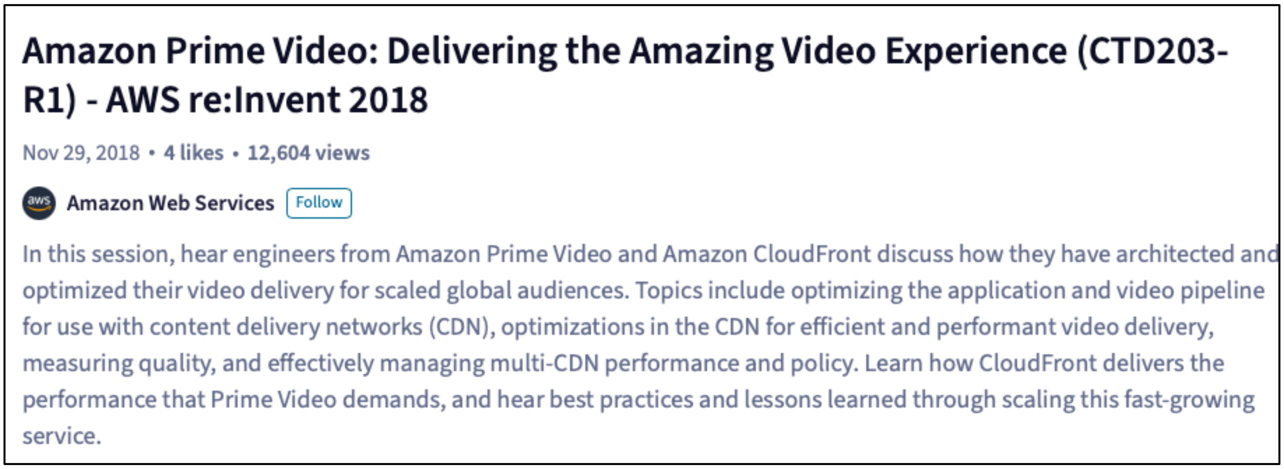
COUNT I - Infringement of the '907 patent by Amazon Cloudfront

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

52. 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 Amazon Cloudfront Products and Services in the United States that infringes (literally and/or under the doctrine of equivalents) at least claim 1 of the '907 patent.

53. On information and belief, one or more components of Amazon Cloudfront Products and Services provide a method of using MPEG-DASH for streaming content to end-

user devices (e.g. Amazon Cloudfront uses at least MPEG-DASH for streaming content to end-user devices. MPEG-DASH is a media streaming format for delivery on-demand and live content to a user).



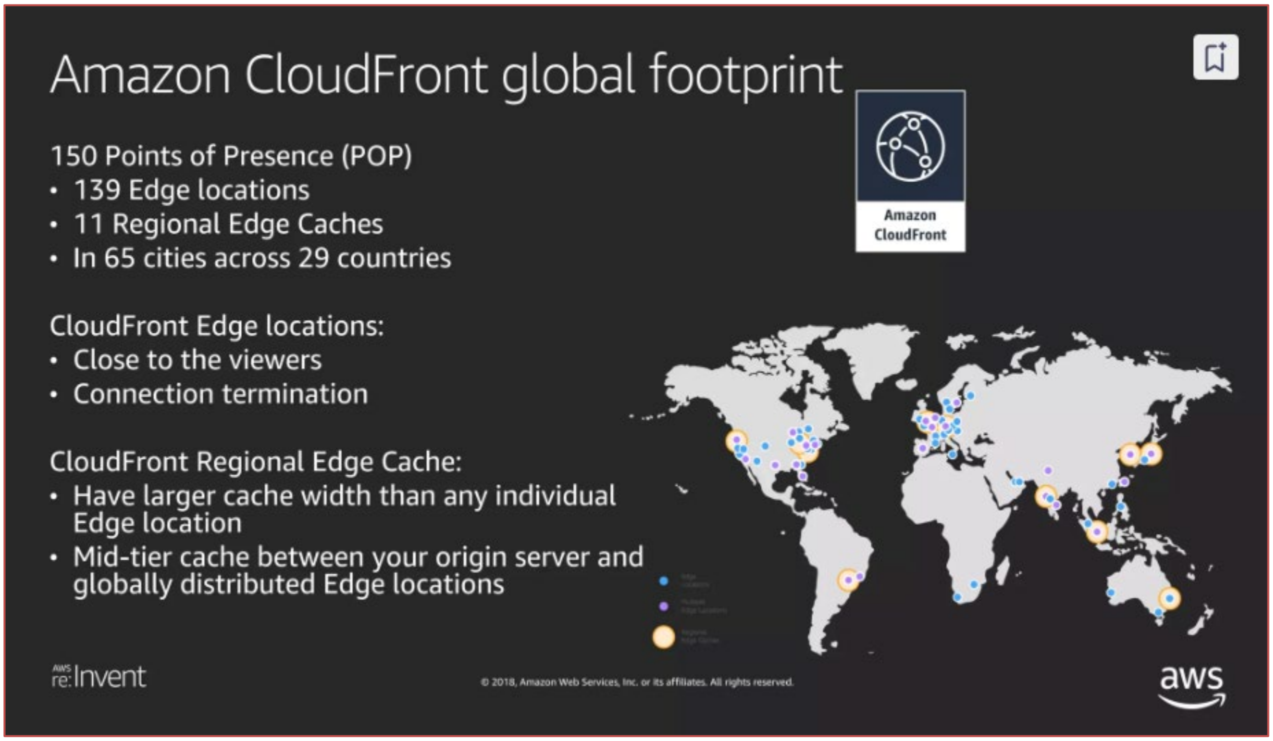
Amazon Prime Video: Delivering the Amazing Video Experience (CTD203-R1) - AWS re:Invent 2018

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/slideshow/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018/124440303.

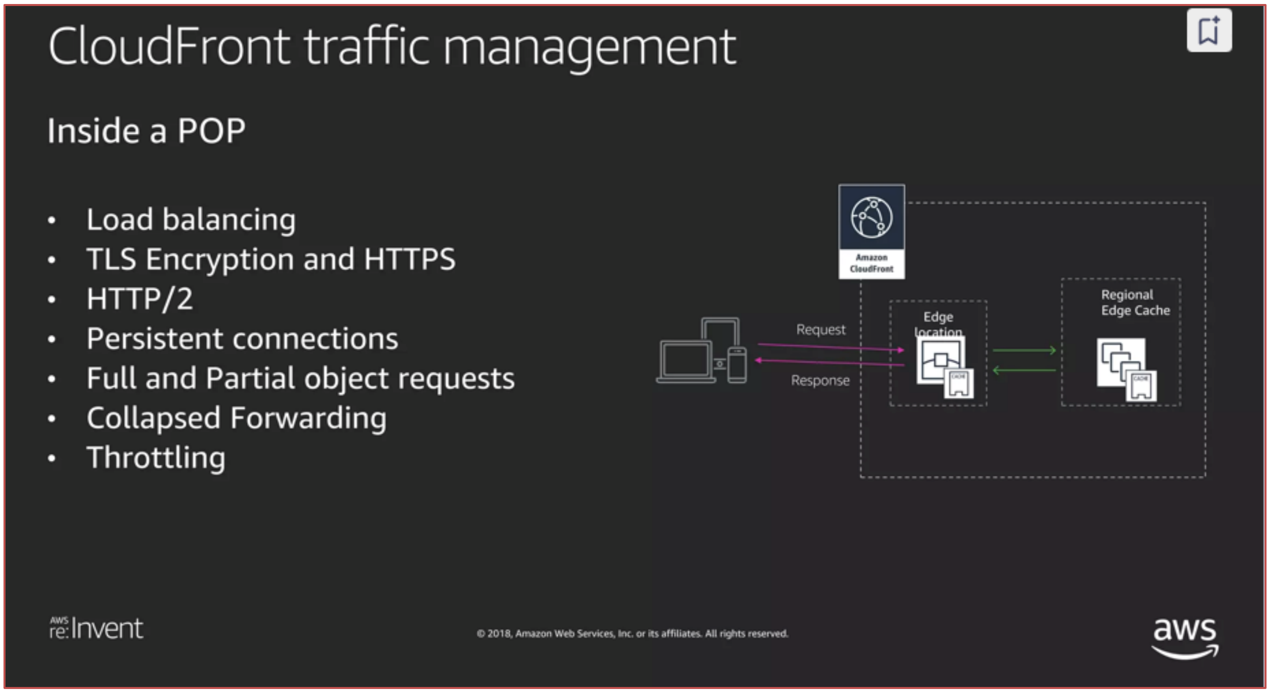


Amazon CloudFront global footprint

150 Points of Presence (POP)
- 139 Edge locations
- 11 Regional Edge Caches
- In 65 cities across 29 countries

CloudFront Edge locations:
- Close to the viewers
- Connection termination

CloudFront Regional Edge Cache:
- Have larger cache width than any individual Edge location
- Mid-tier cache between your origin server and globally distributed Edge locations



CloudFront traffic management

Inside a POP

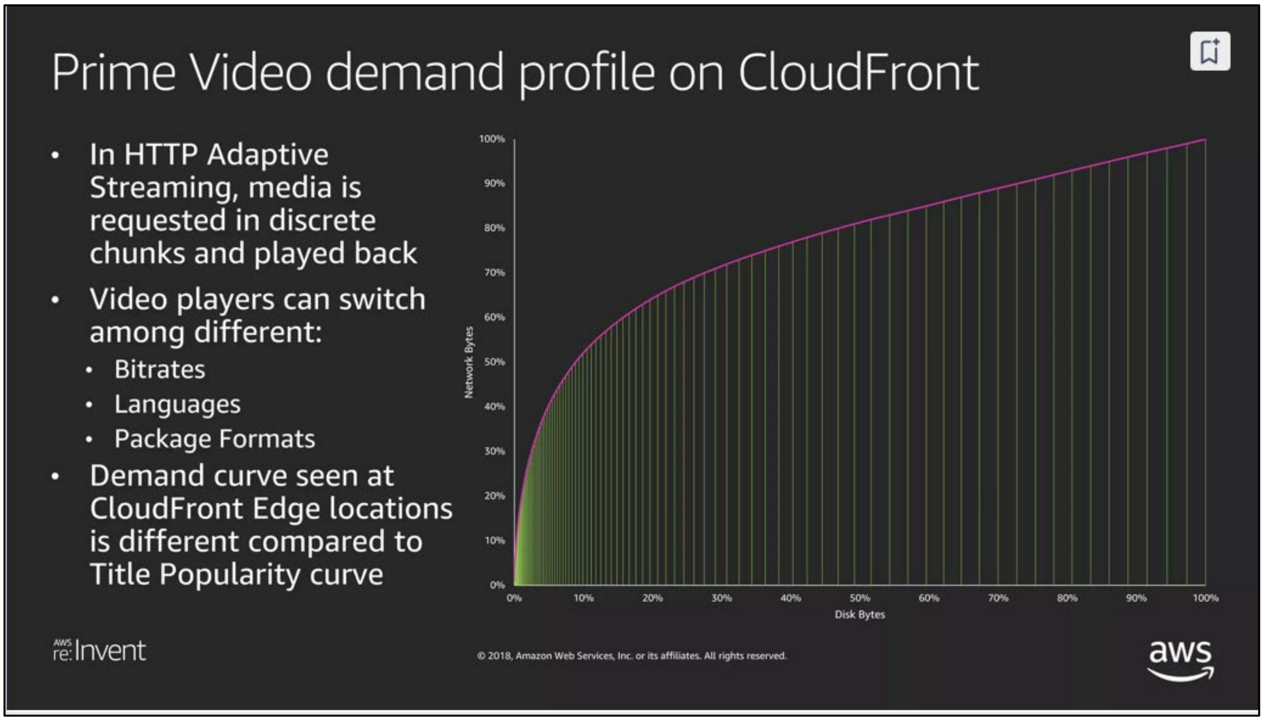
- Load balancing
- TLS Encryption and HTTPS
- HTTP/2
- Persistent connections
- Full and Partial object requests
- Collapsed Forwarding
- Throttling

Amazon CloudFront
Request
Response
Edge location
Regional Edge Cache

re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Prime Video demand profile on CloudFront

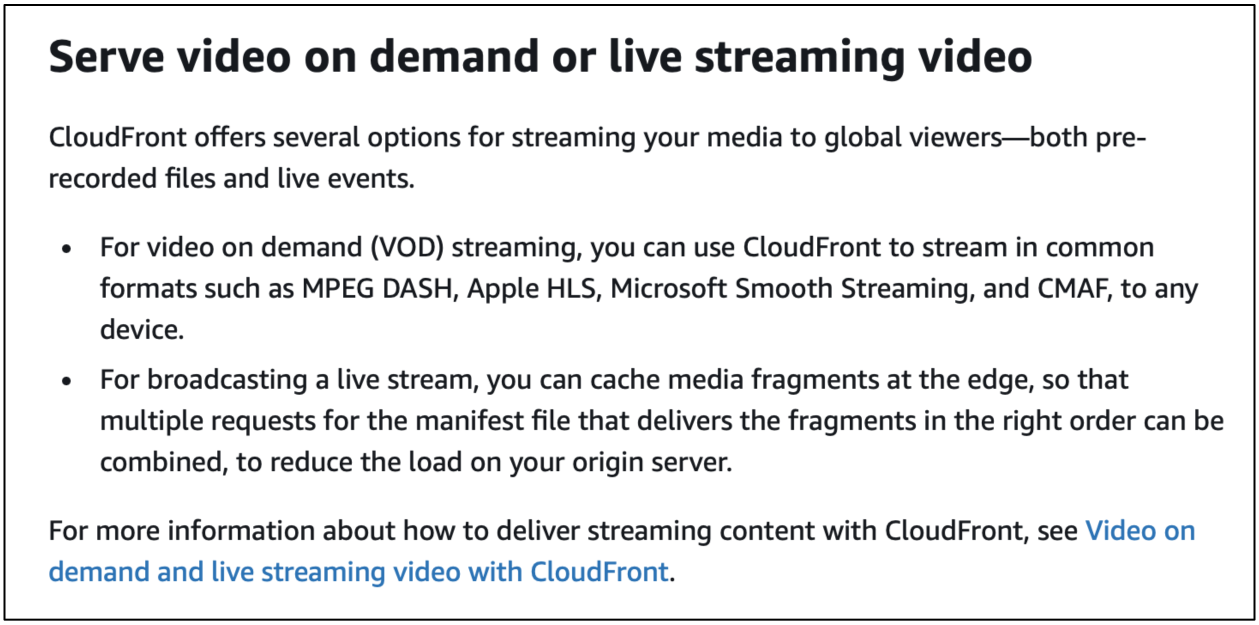
- In HTTP Adaptive Streaming, media is requested in discrete chunks and played back
- Video players can switch among different:
  - Bitrates
  - Languages
  - Package Formats
- Demand curve seen at CloudFront Edge locations is different compared to Title Popularity curve

re:Invent

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

See https://www.slideshare.net/slideshow/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018/124440303#19 (slides 11, 13, and 18).



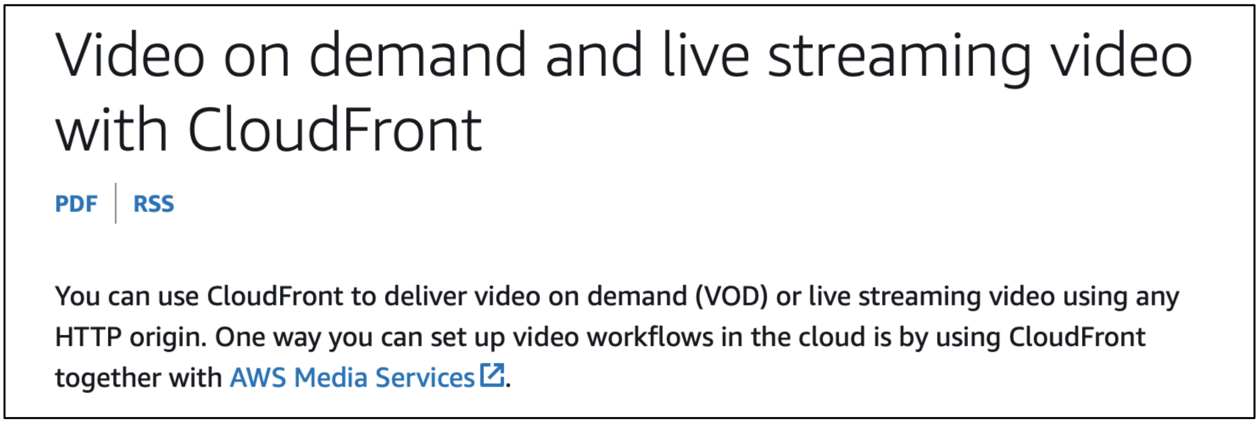
Serve video on demand or live streaming video

CloudFront offers several options for streaming your media to global viewers—both pre-recorded files and live events.

- For video on demand (VOD) streaming, you can use CloudFront to stream in common formats such as MPEG DASH, Apple HLS, Microsoft Smooth Streaming, and CMAF, to any device.
- For broadcasting a live stream, you can cache media fragments at the edge, so that multiple requests for the manifest file that delivers the fragments in the right order can be combined, to reduce the load on your origin server.

For more information about how to deliver streaming content with CloudFront, see Video on demand and live streaming video with CloudFront.

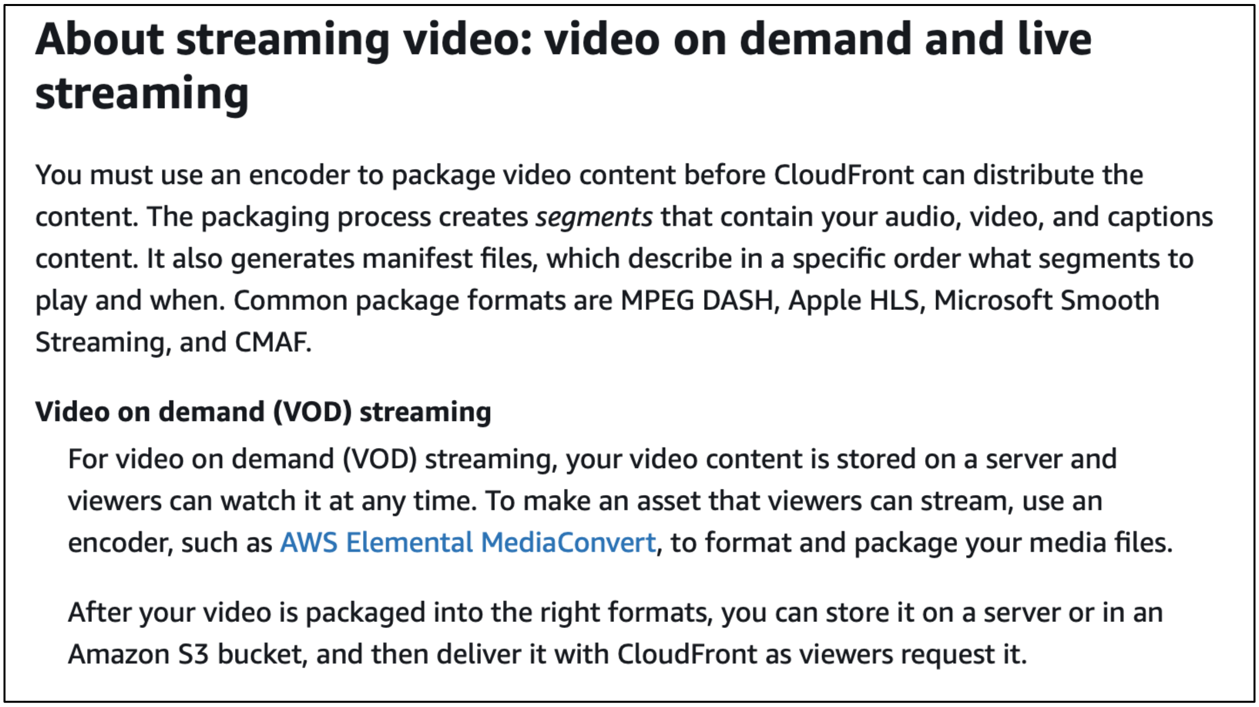
See https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/IntroductionUseCases.html.



Video on demand and live streaming video with CloudFront

You can use CloudFront to deliver video on demand (VOD) or live streaming video using any HTTP origin. One way you can set up video workflows in the cloud is by using CloudFront together with AWS Media Services.

See https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/on-demand-streaming-video.html.



About streaming video: video on demand and live streaming

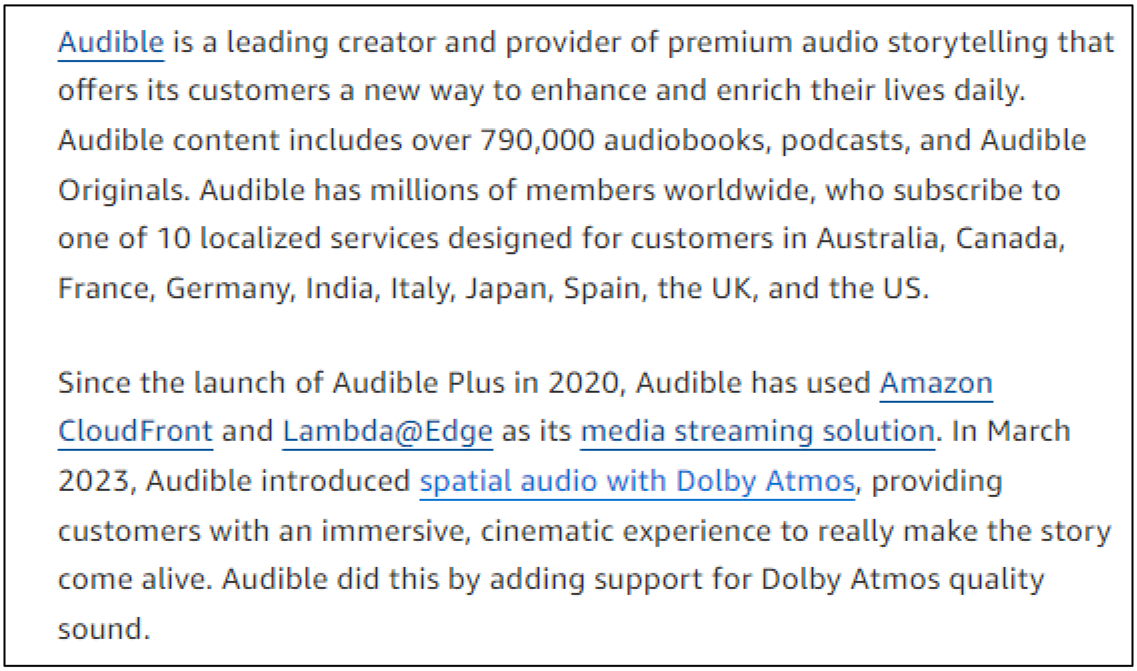
You must use an encoder to package video content before CloudFront can distribute the content. The packaging process creates *segments* that contain your audio, video, and captions content. It also generates manifest files, which describe in a specific order what segments to play and when. Common package formats are MPEG DASH, Apple HLS, Microsoft Smooth Streaming, and CMAF.

Video on demand (VOD) streaming

For video on demand (VOD) streaming, your video content is stored on a server and viewers can watch it at any time. To make an asset that viewers can stream, use an encoder, such as AWS Elemental MediaConvert, to format and package your media files.

After your video is packaged into the right formats, you can store it on a server or in an Amazon S3 bucket, and then deliver it with CloudFront as viewers request it.

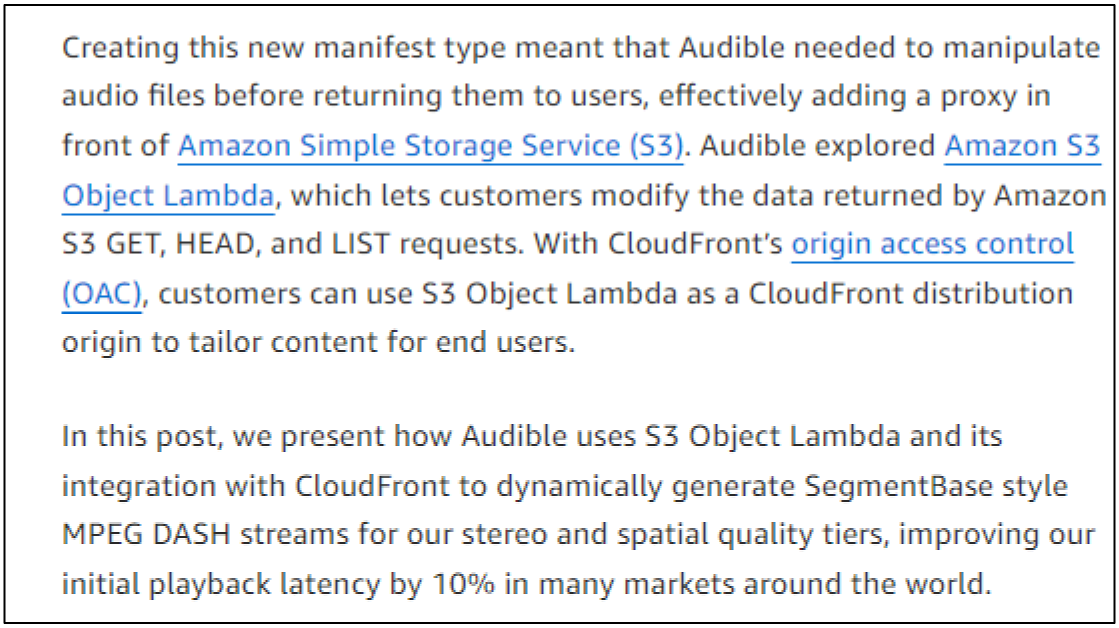
See https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/on-demand-streaming-video.html#StreamingVideo.



Audible is a leading creator and provider of premium audio storytelling that offers its customers a new way to enhance and enrich their lives daily. Audible content includes over 790,000 audiobooks, podcasts, and Audible Originals. Audible has millions of members worldwide, who subscribe to one of 10 localized services designed for customers in Australia, Canada, France, Germany, India, Italy, Japan, Spain, the UK, and the US.

Since the launch of Audible Plus in 2020, Audible has used Amazon CloudFront and Lambda@Edge as its media streaming solution. In March 2023, Audible introduced spatial audio with Dolby Atmos, providing customers with an immersive, cinematic experience to really make the story come alive. Audible did this by adding support for Dolby Atmos quality sound.

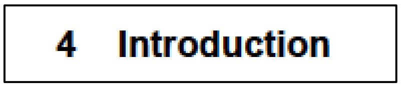
See, https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



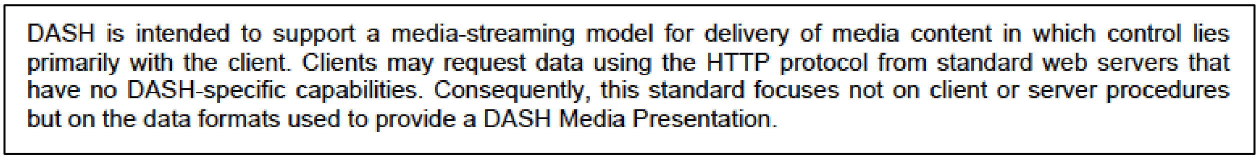
Creating this new manifest type meant that Audible needed to manipulate audio files before returning them to users, effectively adding a proxy in front of Amazon Simple Storage Service (S3). Audible explored Amazon S3 Object Lambda, which lets customers modify the data returned by Amazon S3 GET, HEAD, and LIST requests. With CloudFront's origin access control (OAC), customers can use S3 Object Lambda as a CloudFront distribution origin to tailor content for end users.

In this post, we present how Audible uses S3 Object Lambda and its integration with CloudFront to dynamically generate SegmentBase style MPEG DASH streams for our stereo and spatial quality tiers, improving our initial playback latency by 10% in many markets around the world.

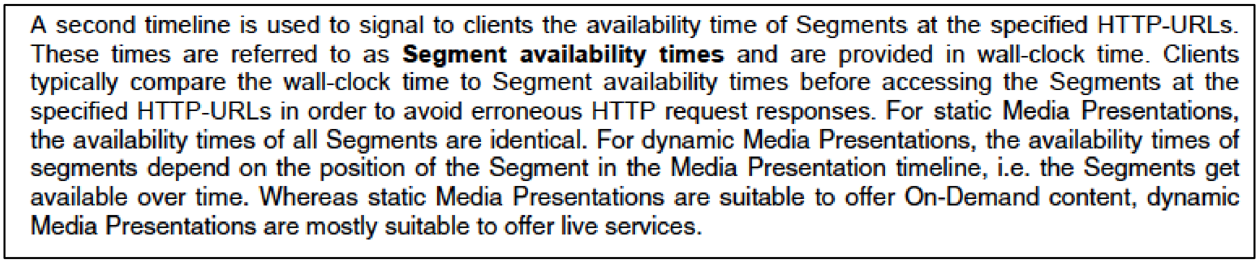
See, https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



4 Introduction

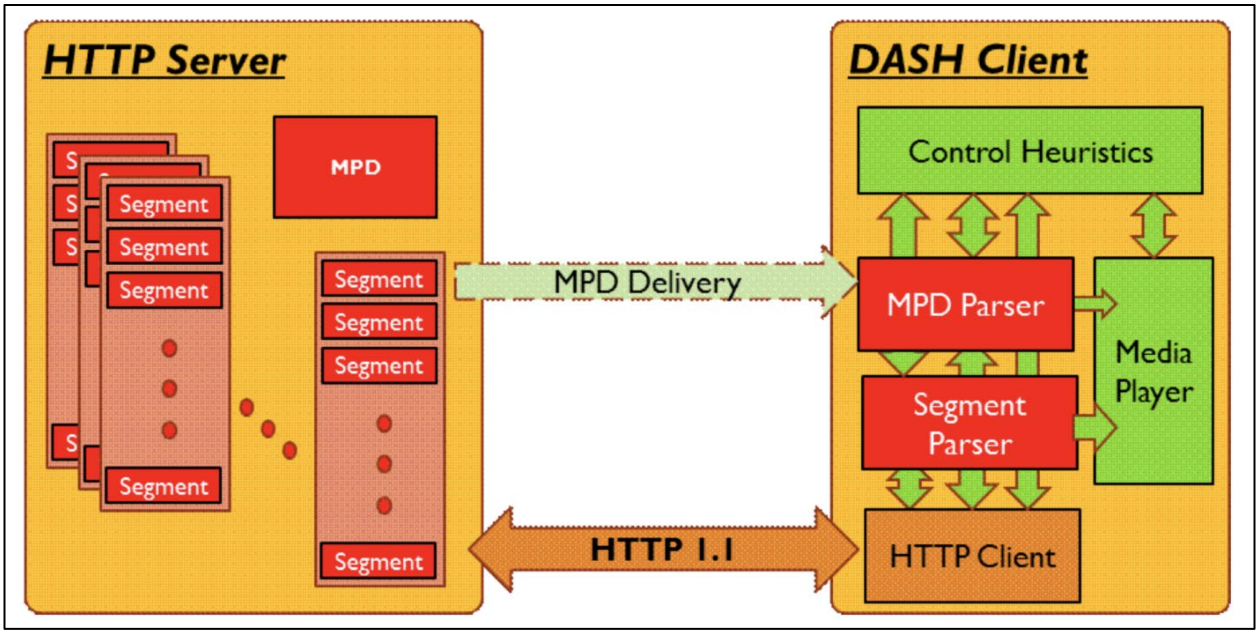


DASH is intended to support a media-streaming model for delivery of media content in which control lies primarily with the client. Clients may request data using the HTTP protocol from standard web servers that have no DASH-specific capabilities. Consequently, this standard focuses not on client or server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

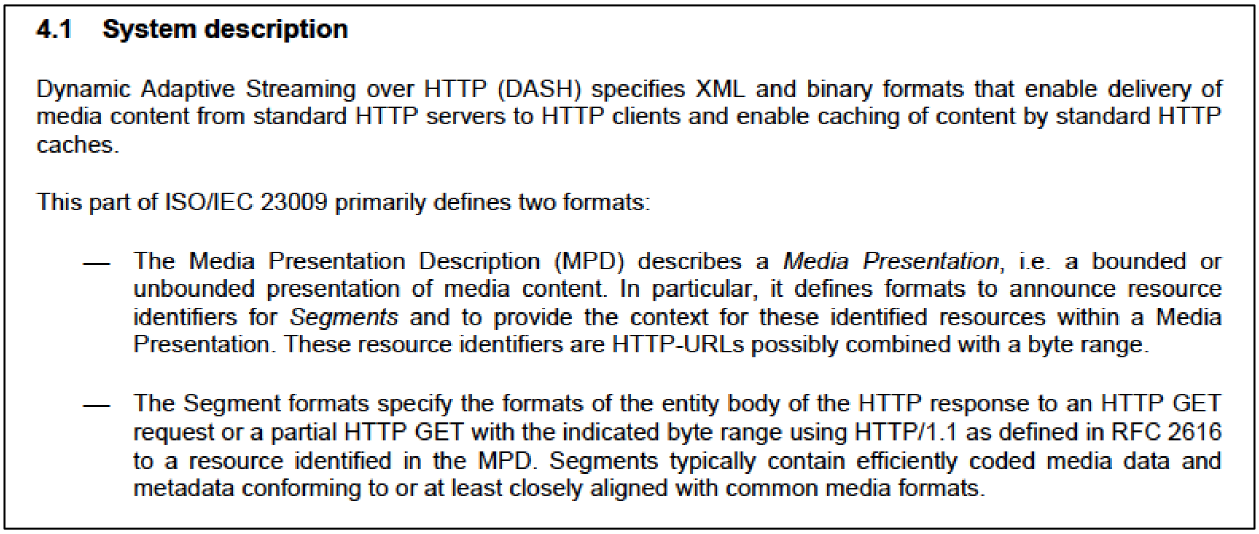


A second timeline is used to signal to clients the availability time of Segments at the specified HTTP-URLs. These times are referred to as **Segment availability times** and are provided in wall-clock time. Clients typically compare the wall-clock time to Segment availability times before accessing the Segments at the specified HTTP-URLs in order to avoid erroneous HTTP request responses. For static Media Presentations, the availability times of all Segments are identical. For dynamic Media Presentations, the availability times of segments depend on the position of the Segment in the Media Presentation timeline, i.e. the Segments get available over time. Whereas static Media Presentations are suitable to offer On-Demand content, dynamic Media Presentations are mostly suitable to offer live services.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 7, 9-10.



See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 3.



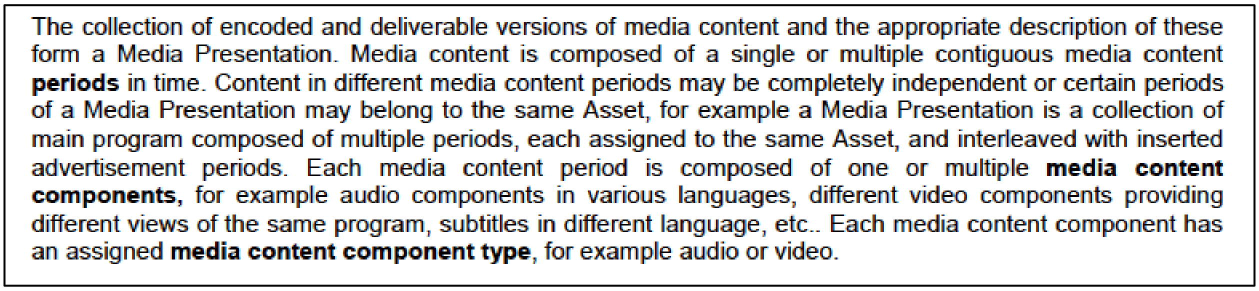
4.1 System description

Dynamic Adaptive Streaming over HTTP (DASH) specifies XML and binary formats that enable delivery of media content from standard HTTP servers to HTTP clients and enable caching of content by standard HTTP caches.

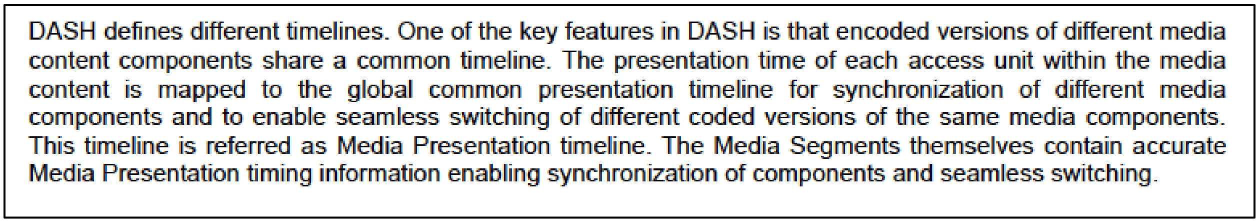
This part of ISO/IEC 23009 primarily defines two formats:

— The Media Presentation Description (MPD) describes a *Media Presentation*, i.e. a bounded or unbounded presentation of media content. In particular, it defines formats to announce resource identifiers for *Segments* and to provide the context for these identified resources within a Media Presentation. These resource identifiers are HTTP-URLs possibly combined with a byte range.

— The Segment formats specify the formats of the entity body of the HTTP response to an HTTP GET request or a partial HTTP GET with the indicated byte range using HTTP/1.1 as defined in RFC 2616 to a resource identified in the MPD. Segments typically contain efficiently coded media data and metadata conforming to or at least closely aligned with common media formats.

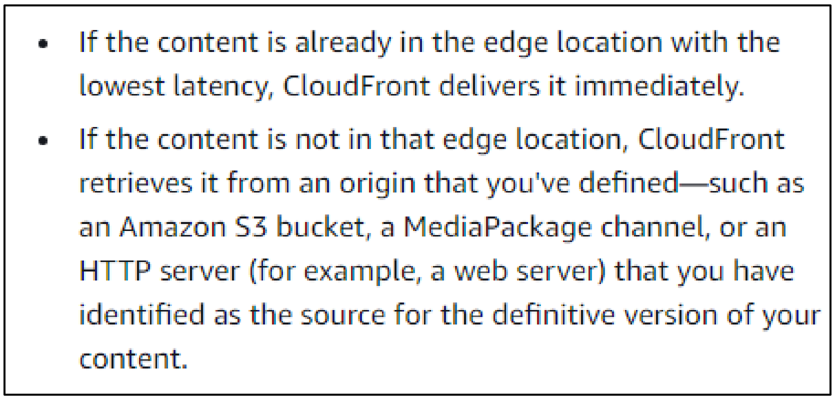


The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent 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.

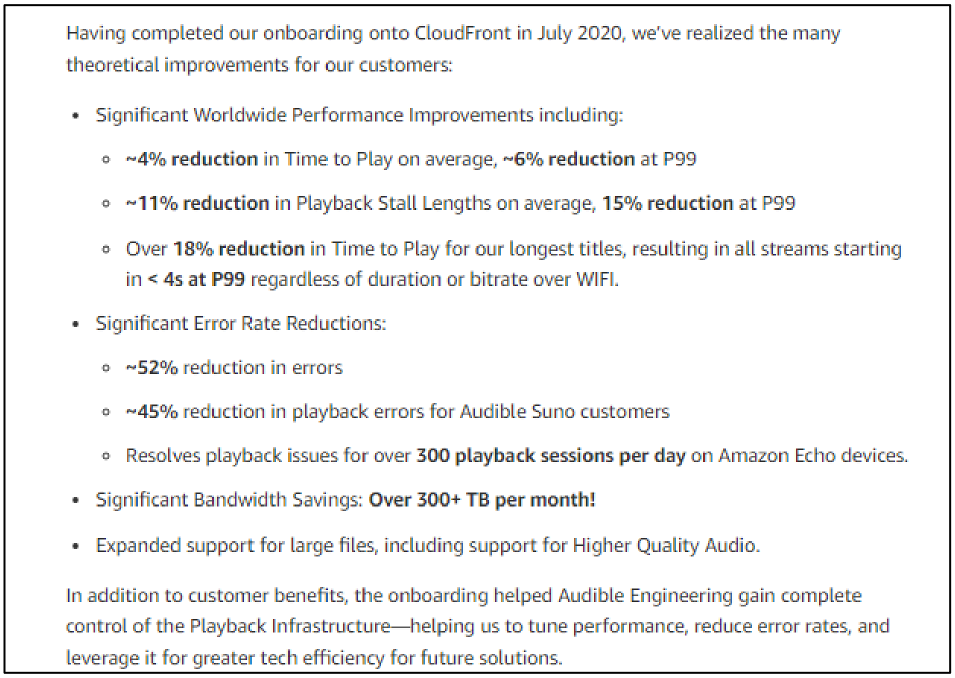


DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different 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/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 7, 9, 10.



- If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
- If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined—such as an Amazon S3 bucket, a MediaPackage channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.



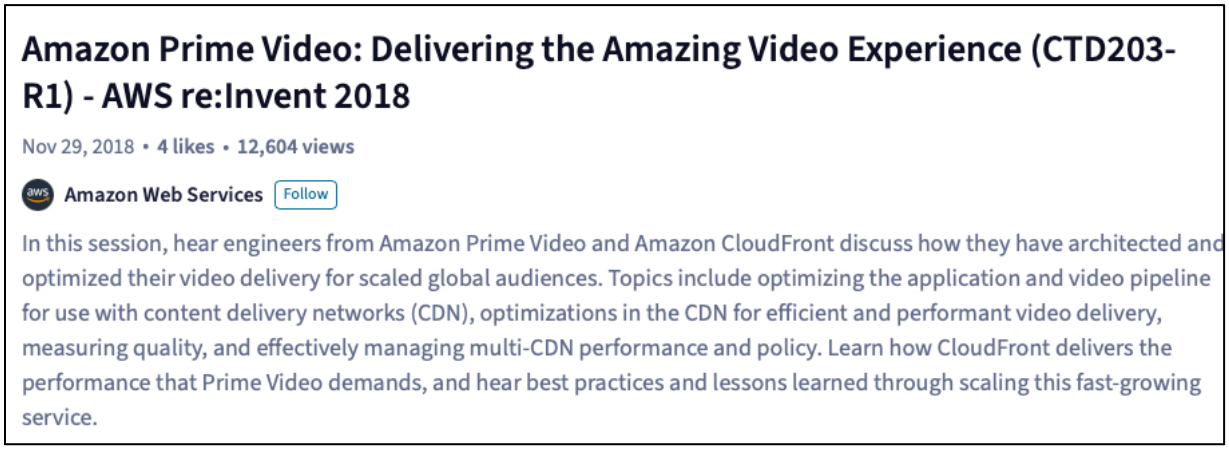
Having completed our onboarding onto CloudFront in July 2020, we've realized the many theoretical improvements for our customers:

- Significant Worldwide Performance Improvements including:
  - ~**4% reduction** in Time to Play on average, **~6% reduction** at P99
  - ~**11% reduction** in Playback Stall Lengths on average, **15% reduction** at P99
  - Over **18% reduction** in Time to Play for our longest titles, resulting in all streams starting in **< 4s at P99** regardless of duration or bitrate over WIFI.
- Significant Error Rate Reductions:
  - ~**52%** reduction in errors
  - ~**45%** reduction in playback errors for Audible Suno customers
  - Resolves playback issues for over **300 playback sessions per day** on Amazon Echo devices.
- Significant Bandwidth Savings: **Over 300+ TB per month!**
- Expanded support for large files, including support for Higher Quality Audio.

In addition to customer benefits, the onboarding helped Audible Engineering gain complete control of the Playback Infrastructure—helping us to tune performance, reduce error rates, and leverage it for greater tech efficiency for future solutions.

See https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html.

54. On information and belief, one or more component of Amazon Cloudfront Products and Services provide a method comprising creating a descriptor file (e.g., Media Presentation Description ("MPD")) for synchronizing a plurality of digital media streams (e.g., audio streams, video streams, and/or mixed media streams), wherein the plurality of digital media streams each contain digital media content (e.g., digital data of the media streams) corresponding to a same originating work (e.g., an artistic work (e.g., a visual work, written work, or audio recording)), wherein the plurality of digital media streams includes a first digital media stream (e.g., a first audio stream, video stream, and/or mixed media stream) containing a digital audio narration of the originating work (e.g., digital audio data corresponding to the artistic work) and one or more other digital media streams (e.g., other audio streams, video streams, and/or mixed media streams), and wherein the descriptor file is external to the first digital media stream (e.g., the MPD is distinct from the streams (e.g., "actual multimedia bitstreams")).



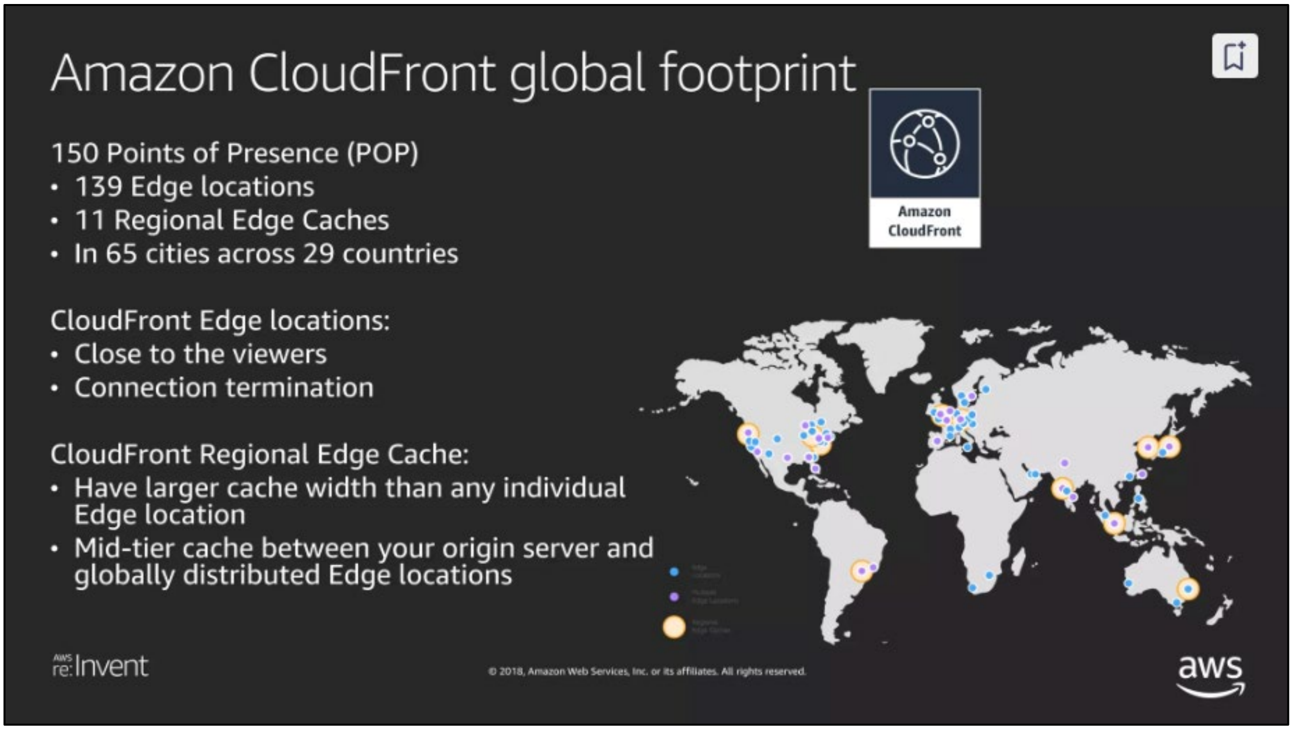
Amazon Prime Video: Delivering the Amazing Video Experience (CTD203-R1) - AWS re:Invent 2018

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/slideshow/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018/124440303.



Amazon CloudFront global footprint

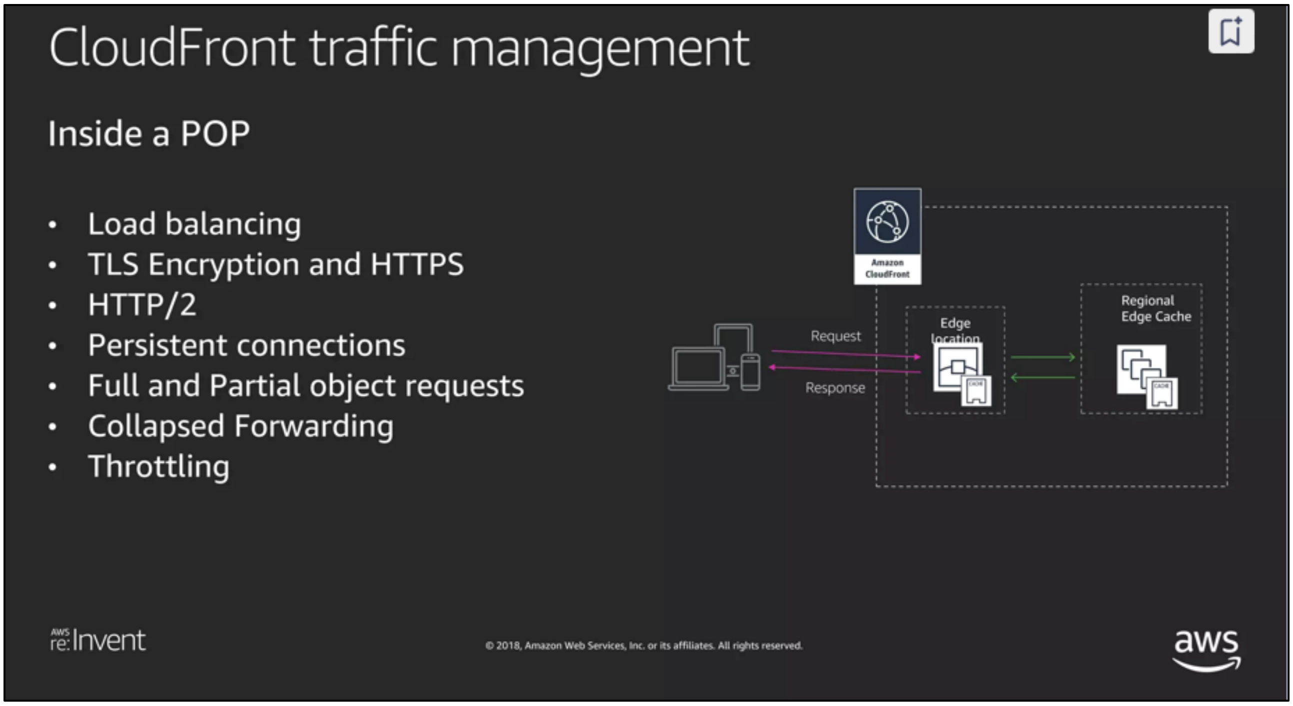
150 Points of Presence (POP)
- 139 Edge locations
- 11 Regional Edge Caches
- In 65 cities across 29 countries

CloudFront Edge locations:
- Close to the viewers
- Connection termination

CloudFront Regional Edge Cache:
- Have larger cache width than any individual Edge location
- Mid-tier cache between your origin server and globally distributed Edge locations

re:Invent

aws



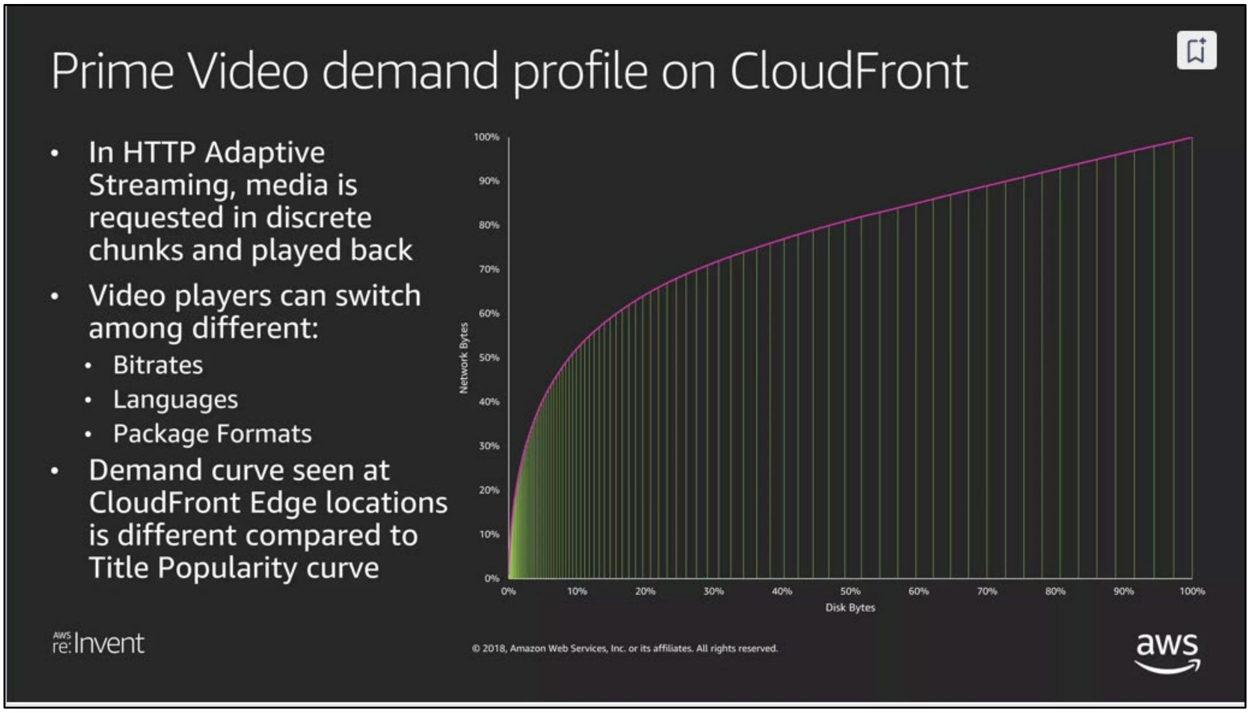
CloudFront traffic management

Inside a POP

- Load balancing
- TLS Encryption and HTTPS
- HTTP/2
- Persistent connections
- Full and Partial object requests
- Collapsed Forwarding
- Throttling

re:Invent

aws

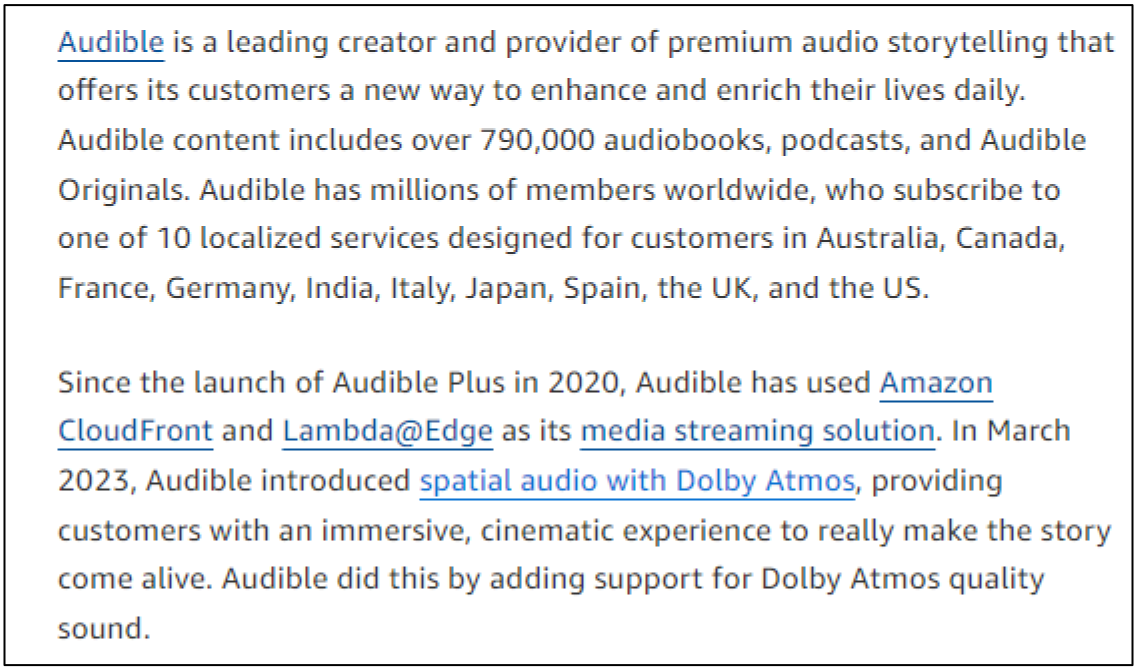


Prime Video demand profile on CloudFront

- In HTTP Adaptive Streaming, media is requested in discrete chunks and played back
- Video players can switch among different:
  - Bitrates
  - Languages
  - Package Formats
- Demand curve seen at CloudFront Edge locations is different compared to Title Popularity curve

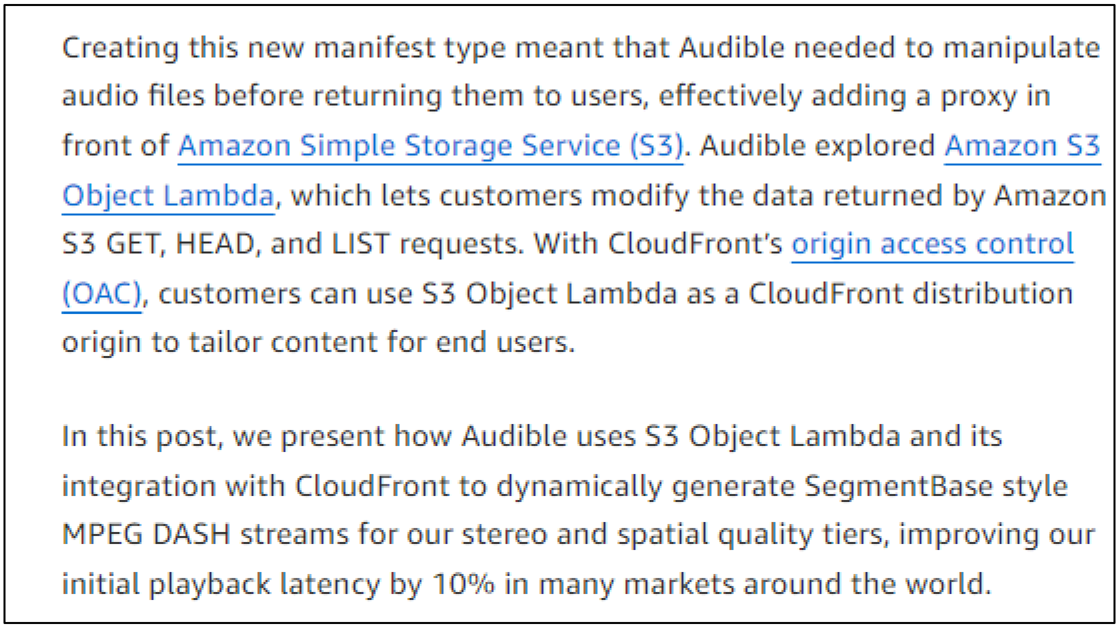
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

See https://www.slideshare.net/AmazonWebServices/amazon-prime-video-delivering-the-amazing-video-experience-ctd203r1-aws-reinvent-2018#19, slides 11, 13, and 18.



Audible is a leading creator and provider of premium audio storytelling that offers its customers a new way to enhance and enrich their lives daily. Audible content includes over 790,000 audiobooks, podcasts, and Audible Originals. Audible has millions of members worldwide, who subscribe to one of 10 localized services designed for customers in Australia, Canada, France, Germany, India, Italy, Japan, Spain, the UK, and the US.

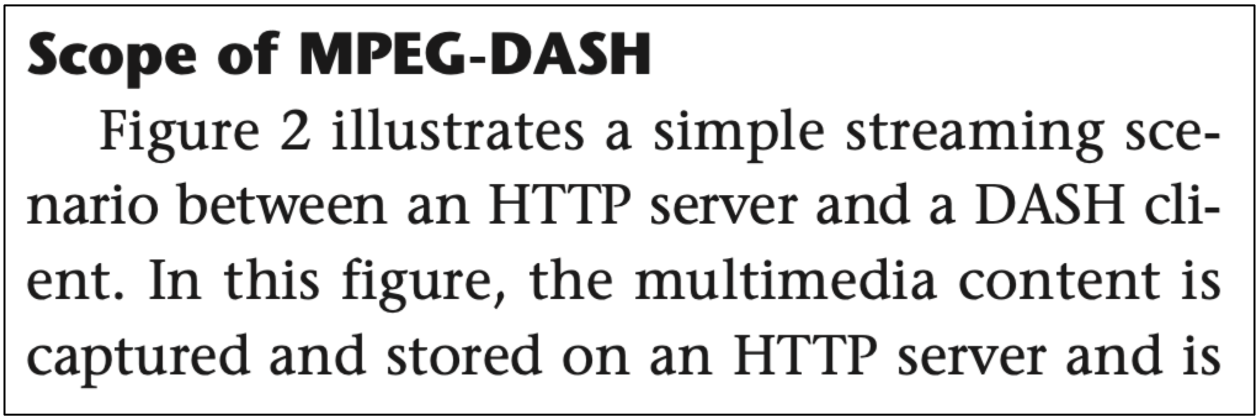
Since the launch of Audible Plus in 2020, Audible has used Amazon CloudFront and Lambda@Edge as its media streaming solution. In March 2023, Audible introduced spatial audio with Dolby Atmos, providing customers with an immersive, cinematic experience to really make the story come alive. Audible did this by adding support for Dolby Atmos quality sound.



Creating this new manifest type meant that Audible needed to manipulate audio files before returning them to users, effectively adding a proxy in front of Amazon Simple Storage Service (S3). Audible explored Amazon S3 Object Lambda, which lets customers modify the data returned by Amazon S3 GET, HEAD, and LIST requests. With CloudFront's origin access control (OAC), customers can use S3 Object Lambda as a CloudFront distribution origin to tailor content for end users.

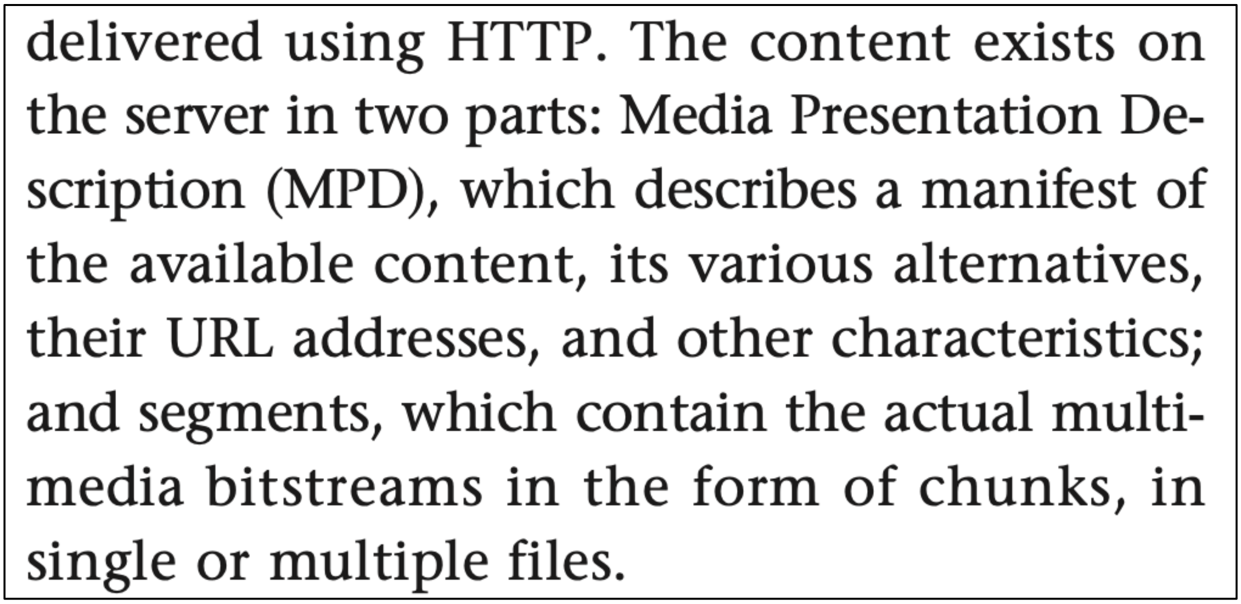
In this post, we present how Audible uses S3 Object Lambda and its integration with CloudFront to dynamically generate SegmentBase style MPEG DASH streams for our stereo and spatial quality tiers, improving our initial playback latency by 10% in many markets around the world.

See, https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



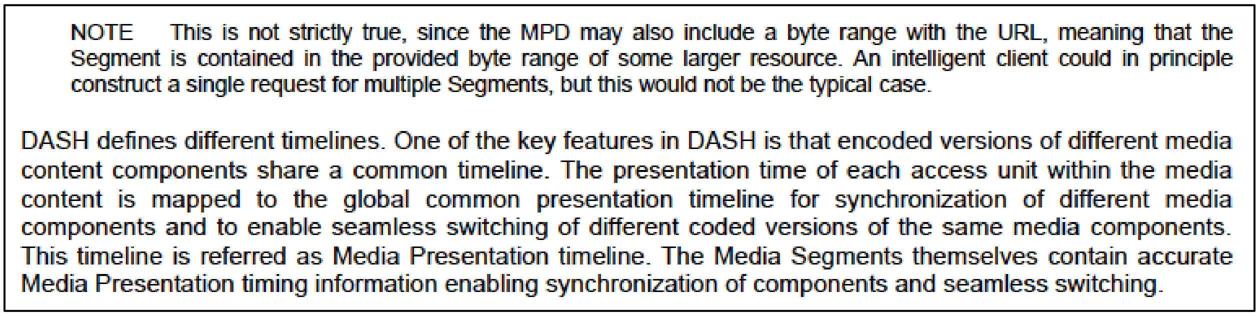
Scope of MPEG-DASH

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.

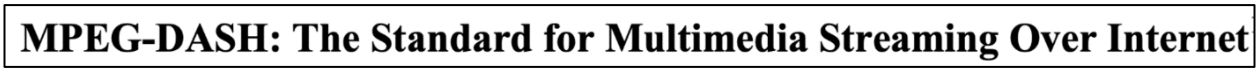
See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/T_MM1_TheMPEGDASHStandard.pdf at 64.



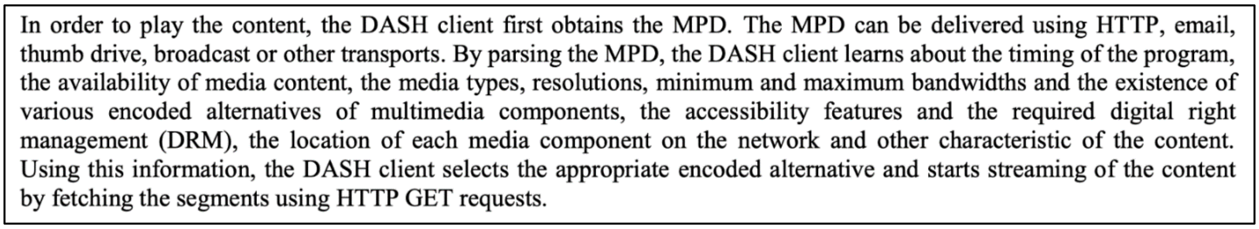
NOTE This is not strictly true, since the MPD may also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client could in principle construct a single request for multiple Segments, but this would not be the typical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

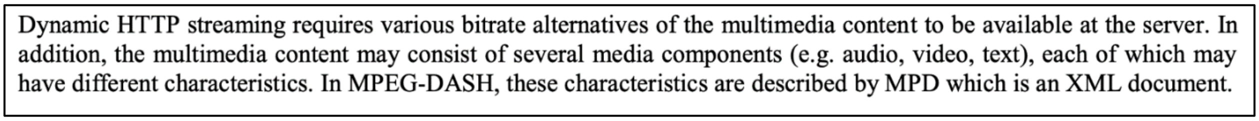
See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 10.



MPEG-DASH: The Standard for Multimedia Streaming Over Internet

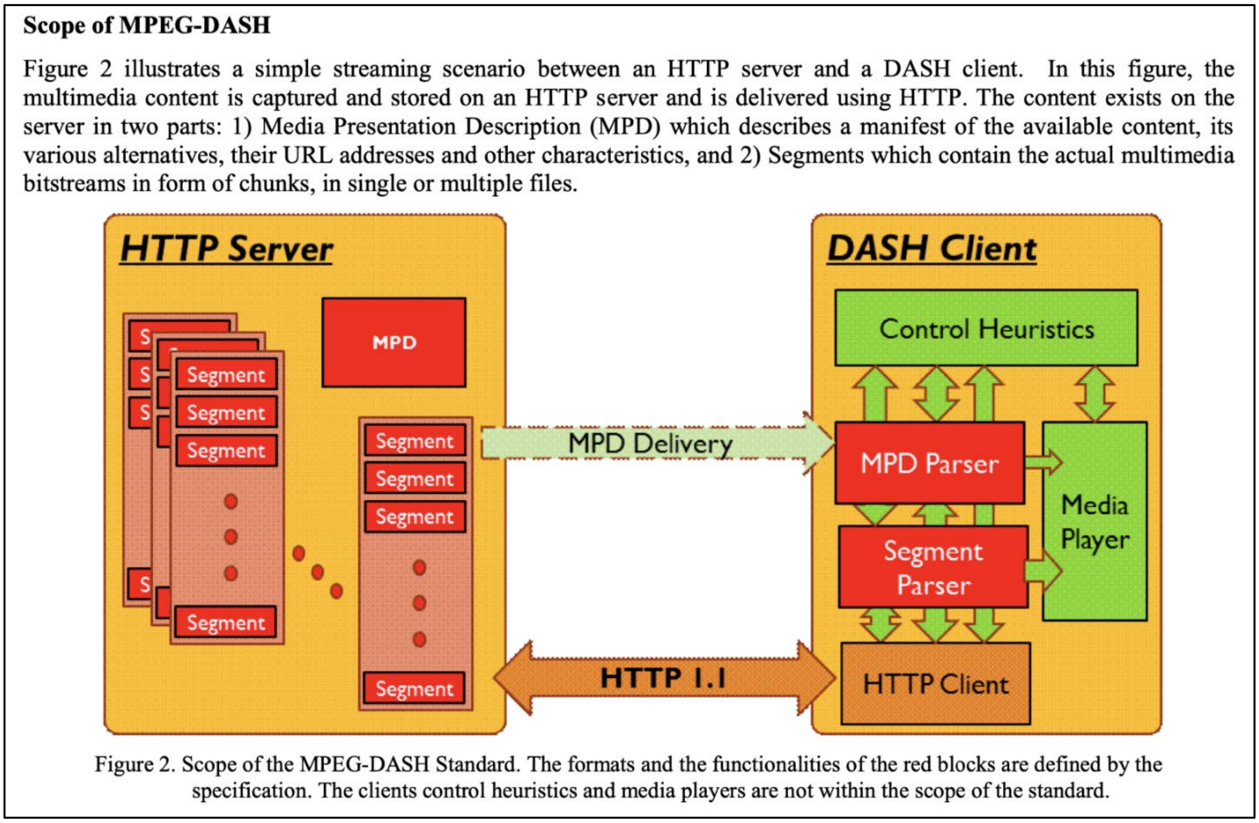


In order to play the content, the DASH client first obtains the MPD. The MPD can be delivered using HTTP, email, thumb drive, broadcast or other transports. By parsing the MPD, the DASH client learns about the timing of the program, the availability of media content, the media types, resolutions, minimum 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 at 1, 3, 6.

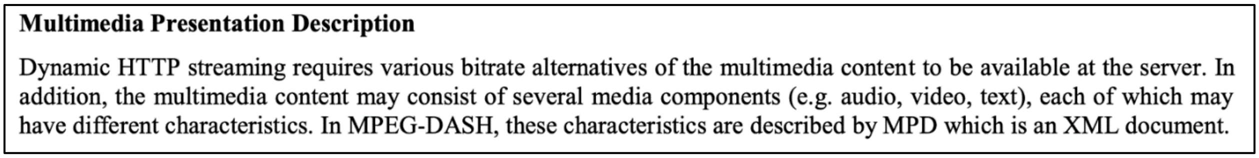


Scope of MPEG-DASH

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.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 3.



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.

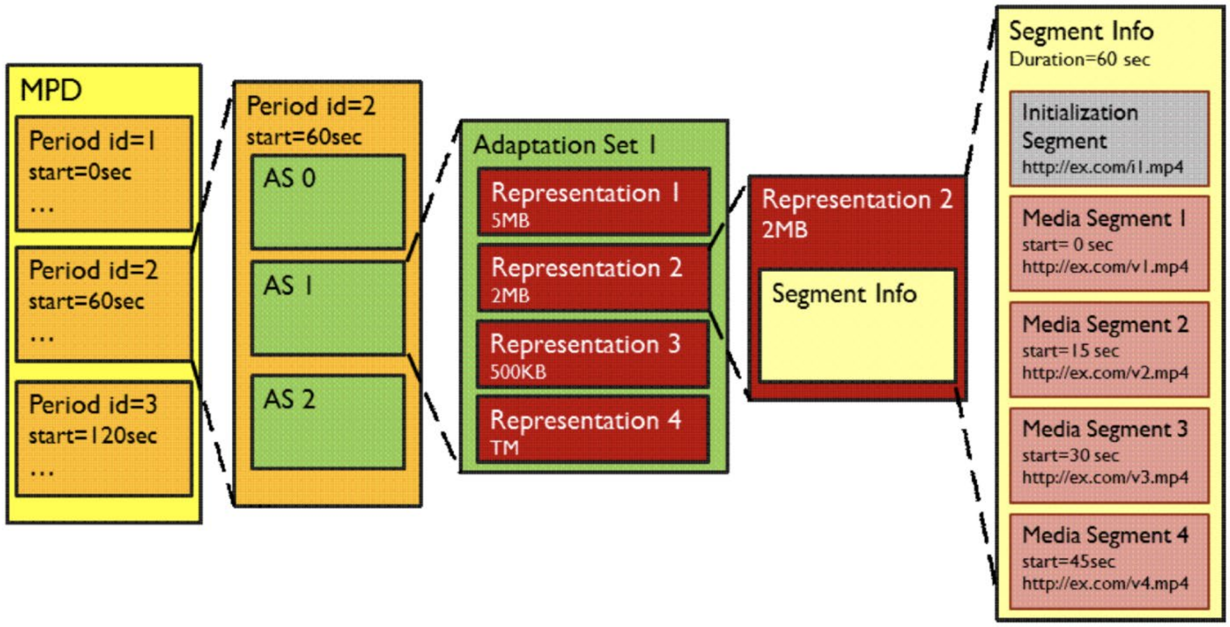
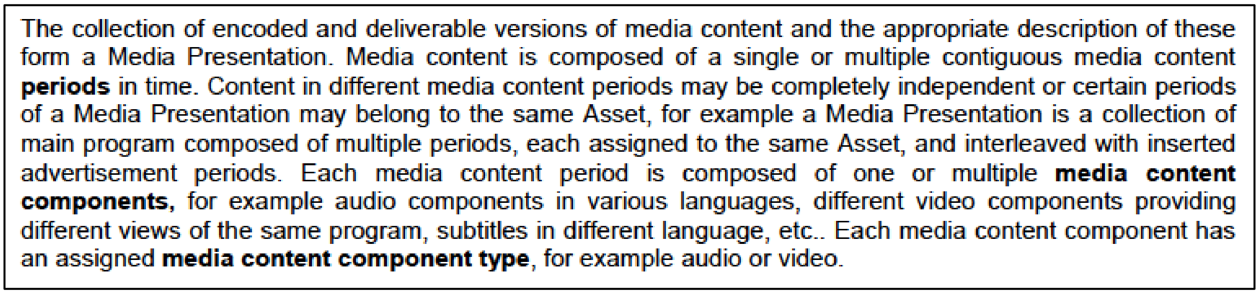


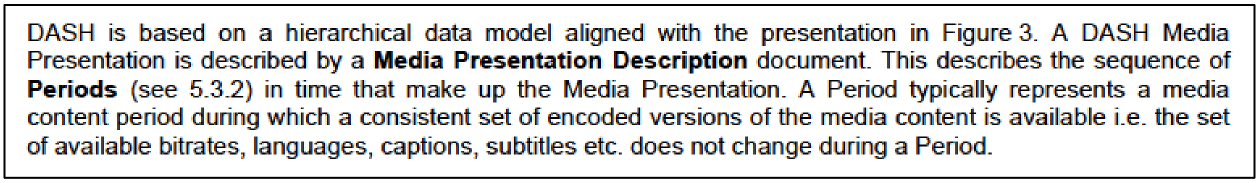
Figure 3. The MPD hierarchical data model.

See id. at 3-4.

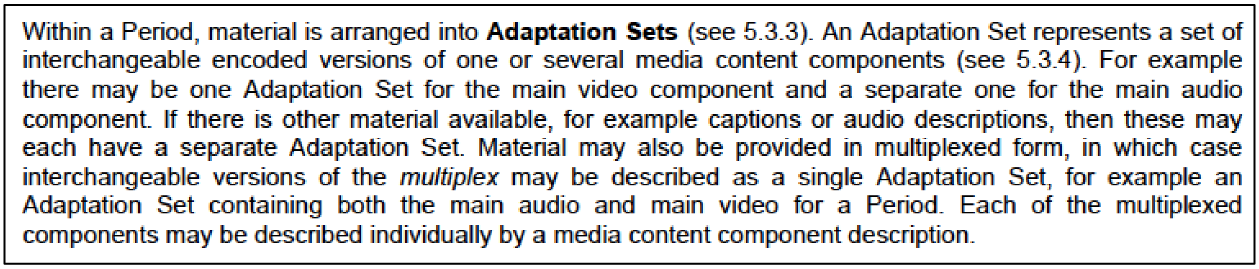
4.3 DASH data model overview



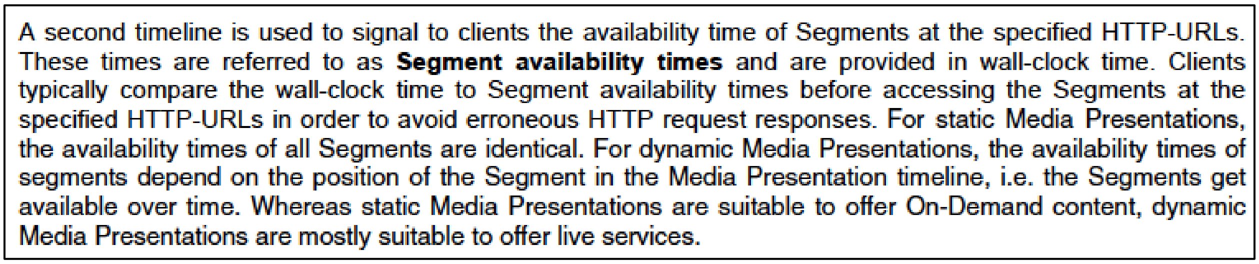
The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent 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.

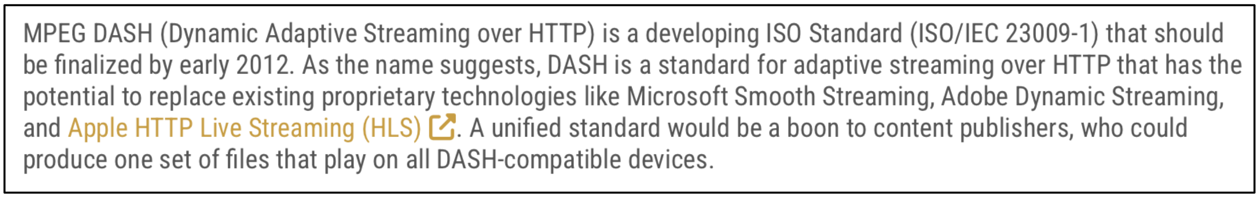


Within a Period, material is arranged into **Adaptation Sets** (see 5.3.3). An Adaptation Set represents a set of interchangeable encoded versions of one or several media content components (see 5.3.4). For example there may be one Adaptation Set for the main video component and a separate one for the main audio component. If there is other material available, for example captions or audio descriptions, then these may each have a separate Adaptation Set. Material may also be provided in multiplexed form, in which case interchangeable versions of the *multiplex* may be described as a single Adaptation Set, for example an Adaptation Set containing both the main audio and main video for a Period. Each of the multiplexed components may be described individually by a media content component description.

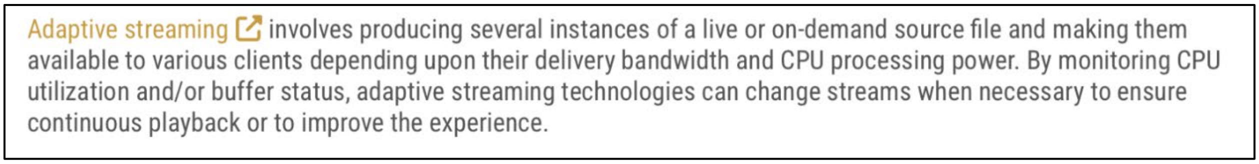


A second timeline is used to signal to clients the availability time of Segments at the specified HTTP-URLs. These times are referred to as **Segment availability times** and are provided in wall-clock time. Clients typically compare the wall-clock time to Segment availability times before accessing the Segments at the specified HTTP-URLs in order to avoid erroneous HTTP request responses. For static Media Presentations, the availability times of all Segments are identical. For dynamic Media Presentations, the availability times of segments depend on the position of the Segment in the Media Presentation timeline, i.e. the Segments get available over time. Whereas static Media Presentations are suitable to offer On-Demand content, dynamic Media Presentations are mostly suitable to offer live services.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9, 10.

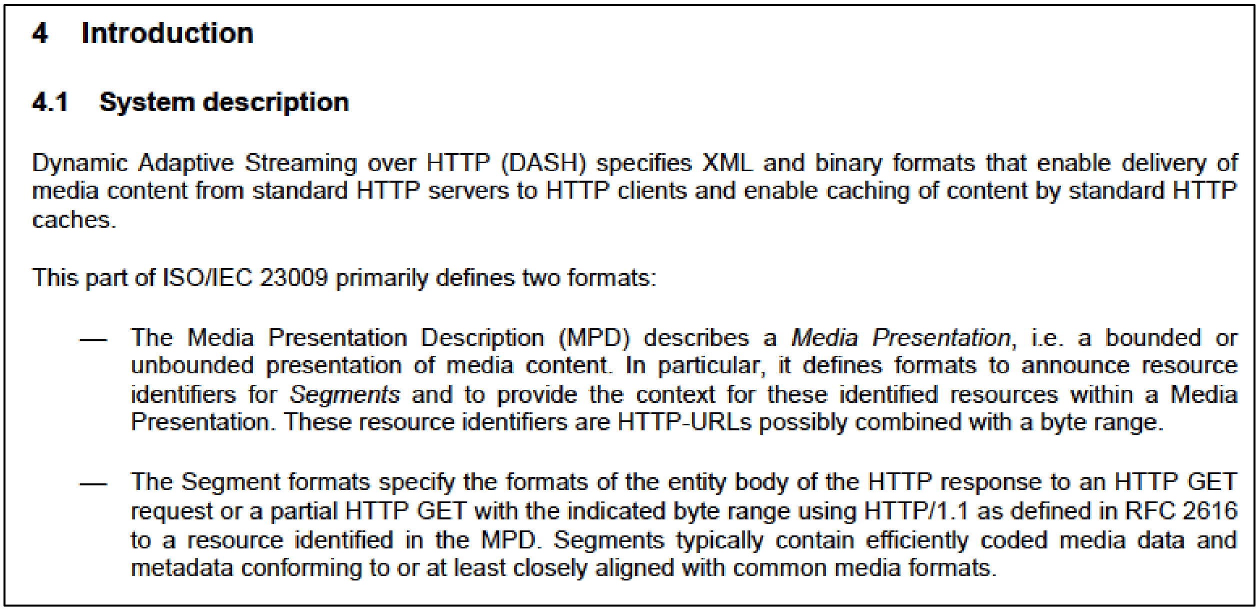


MPEG DASH (Dynamic Adaptive Streaming over HTTP) is a developing ISO Standard (ISO/IEC 23009-1) that should be finalized by early 2012. As the name suggests, DASH is a standard for adaptive streaming over HTTP that has the potential to replace existing proprietary technologies like Microsoft Smooth Streaming, Adobe Dynamic Streaming, and Apple HTTP Live Streaming (HLS). A unified standard would be a boon to content publishers, who could produce one set of files that play on all DASH-compatible devices.



Adaptive streaming involves producing several instances of a live or on-demand source file and making them available to various clients depending upon their delivery bandwidth and CPU processing power. By monitoring CPU utilization and/or buffer status, adaptive streaming technologies can change streams when necessary to ensure continuous playback or to improve the experience.

See https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=79041.



4 Introduction

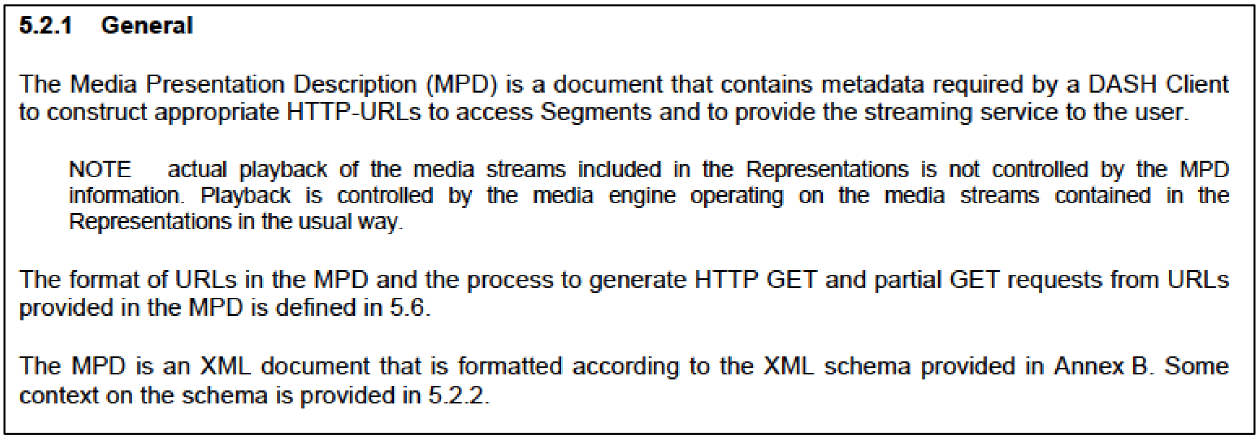
4.1 System description

Dynamic Adaptive Streaming over HTTP (DASH) specifies XML and binary formats that enable delivery of media content from standard HTTP servers to HTTP clients and enable caching of content by standard HTTP caches.

This part of ISO/IEC 23009 primarily defines two formats:

— The Media Presentation Description (MPD) describes a *Media Presentation*, i.e. a bounded or unbounded presentation of media content. In particular, it defines formats to announce resource identifiers for *Segments* and to provide the context for these identified resources within a Media Presentation. These resource identifiers are HTTP-URLs possibly combined with a byte range.

— The Segment formats specify the formats of the entity body of the HTTP response to an HTTP GET request or a partial HTTP GET with the indicated byte range using HTTP/1.1 as defined in RFC 2616 to a resource identified in the MPD. Segments typically contain efficiently coded media data and metadata conforming to or at least closely aligned with common media formats.



5.2.1 General

The Media Presentation Description (MPD) is a document that contains metadata required by a DASH Client to construct appropriate HTTP-URLs to access Segments and to provide the streaming service to the user.

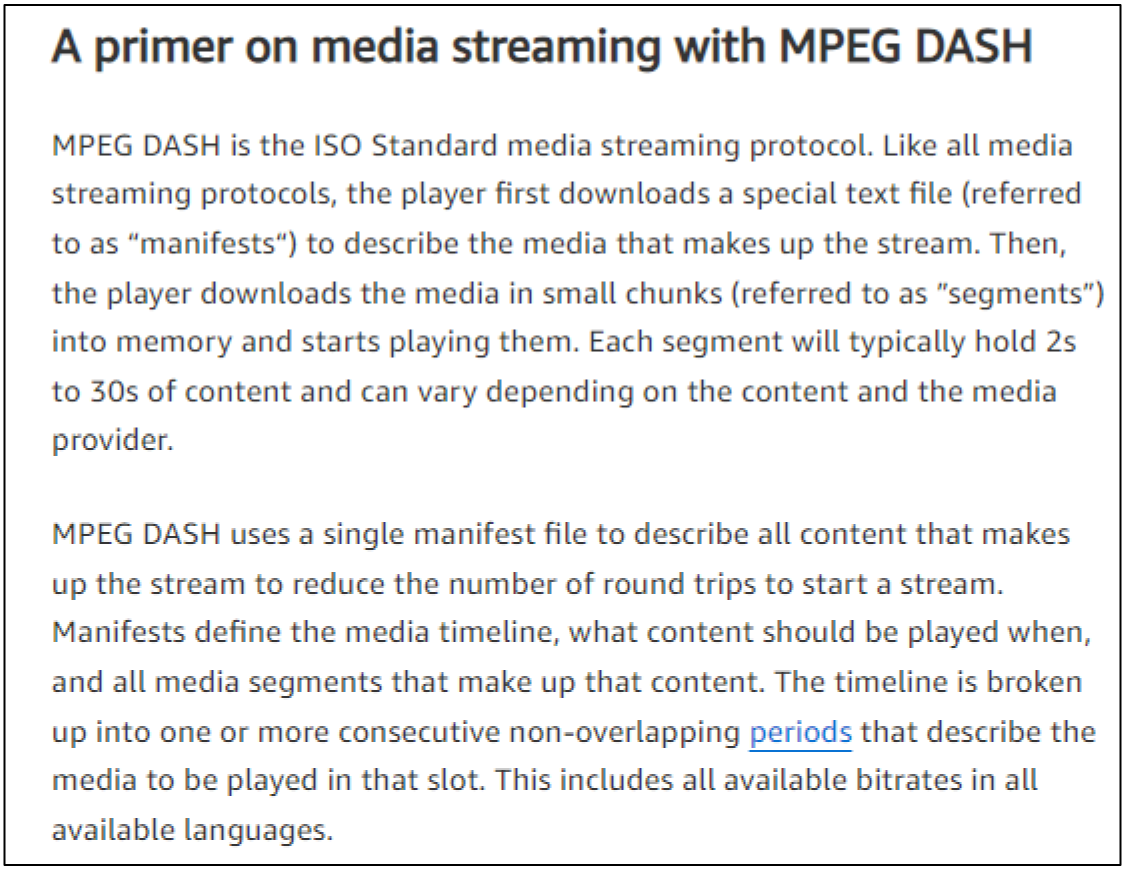
NOTE actual playback of the media streams included in the Representations is not controlled by the MPD information. Playback is controlled by the media engine operating on the media streams contained in the Representations in the usual way.

The format of URLs in the MPD and the process to generate HTTP GET and partial GET requests from URLs provided in the MPD is defined in 5.6.

The MPD is an XML document that is formatted according to the XML schema provided in Annex B. Some context on the schema is provided in 5.2.2.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 7, 16.

55. On information and belief, one or more components of Amazon Cloudfront Products and Services provide a method comprising storing location information (e.g., storing SegmentURL element and/or byte range properties in a SegmentList) for the plurality of digital media streams in the descriptor file.

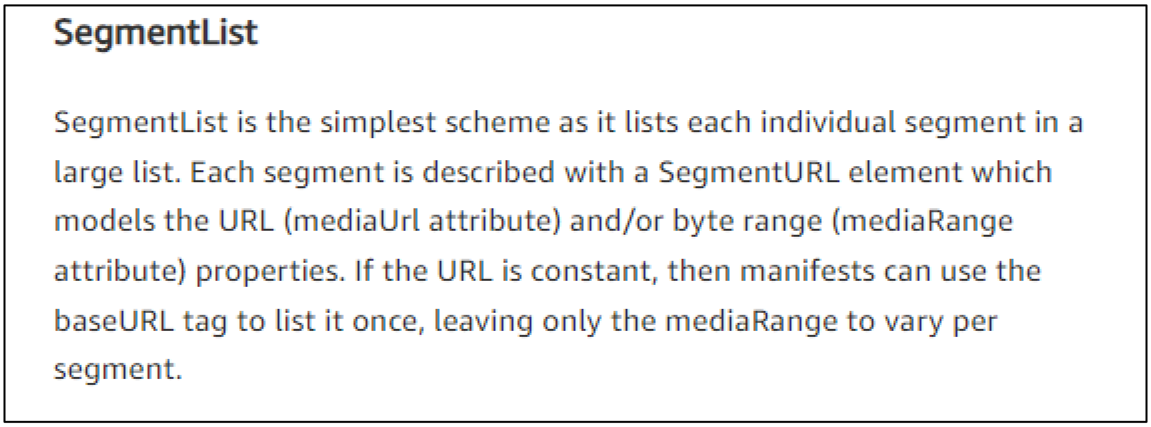


A primer on media streaming with MPEG DASH

MPEG DASH is the ISO Standard media streaming protocol. Like all media streaming protocols, the player first downloads a special text file (referred to as "manifests") to describe the media that makes up the stream. Then, the player downloads the media in small chunks (referred to as "segments") into memory and starts playing them. Each segment will typically hold 2s to 30s of content and can vary depending on the content and the media provider.

MPEG DASH uses a single manifest file to describe all content that makes up the stream to reduce the number of round trips to start a stream. Manifests define the media timeline, what content should be played when, and all media segments that make up that content. The timeline is broken up into one or more consecutive non-overlapping periods that describe the media to be played in that slot. This includes all available bitrates in all available languages.

See, https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.

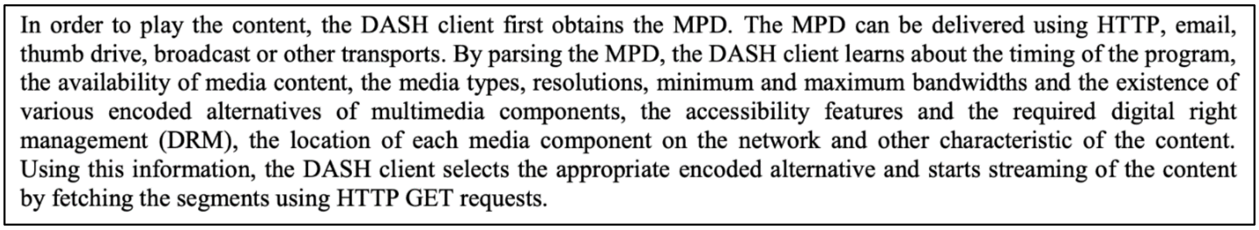


SegmentList

SegmentList is the simplest scheme as it lists each individual segment in a large list. Each segment is described with a SegmentURL element which models the URL (mediaUrl attribute) and/or byte range (mediaRange attribute) properties. If the URL is constant, then manifests can use the baseURL tag to list it once, leaving only the mediaRange to vary per segment.

See, https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.

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.

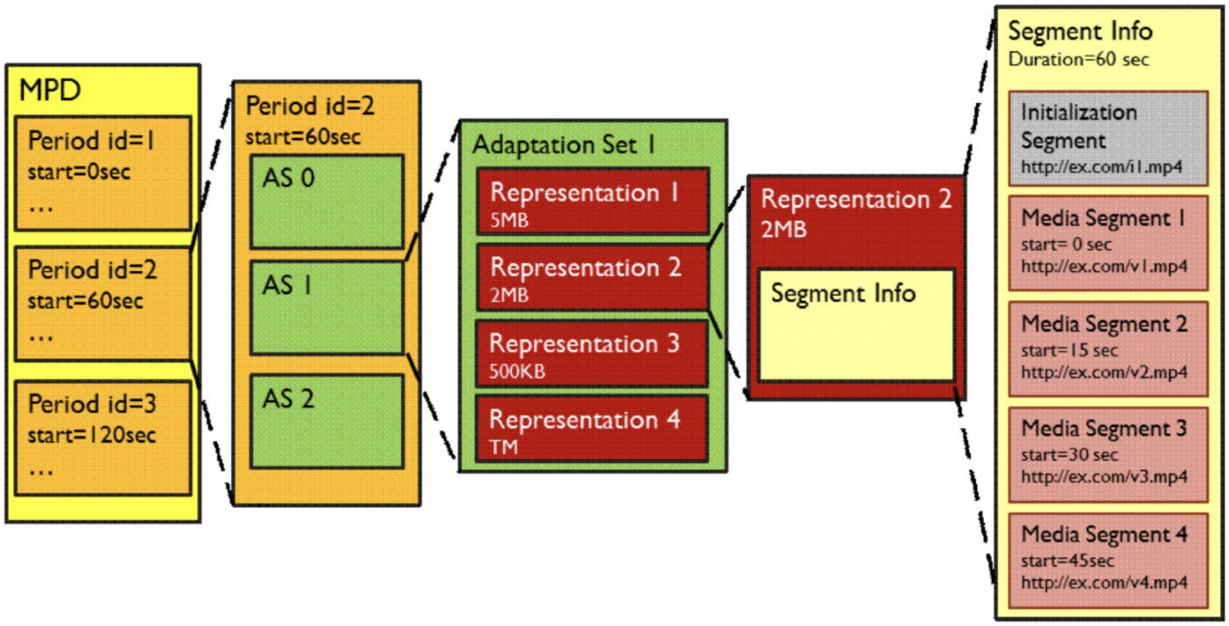
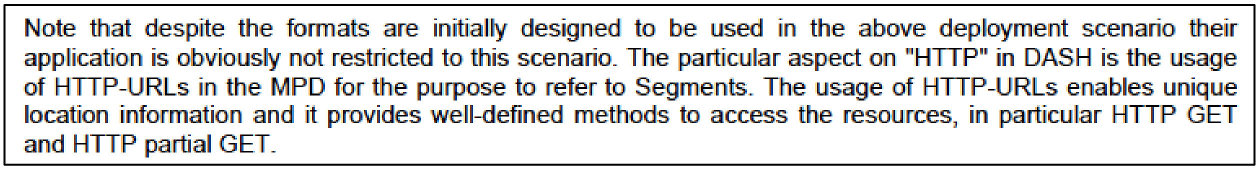
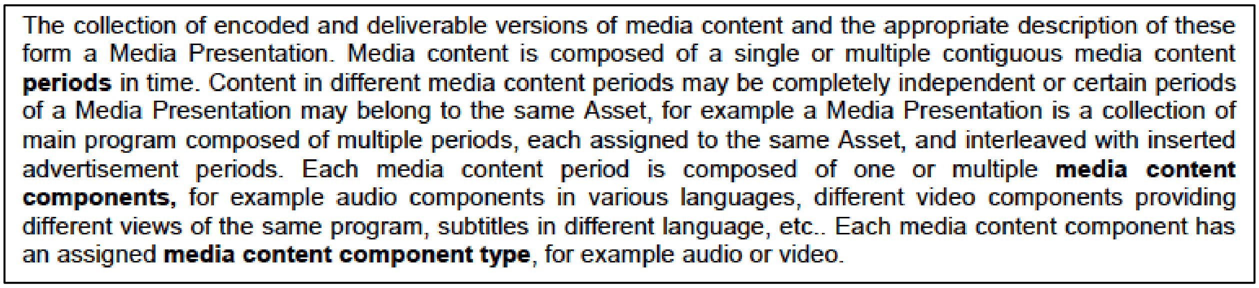


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 1, 3, 4.

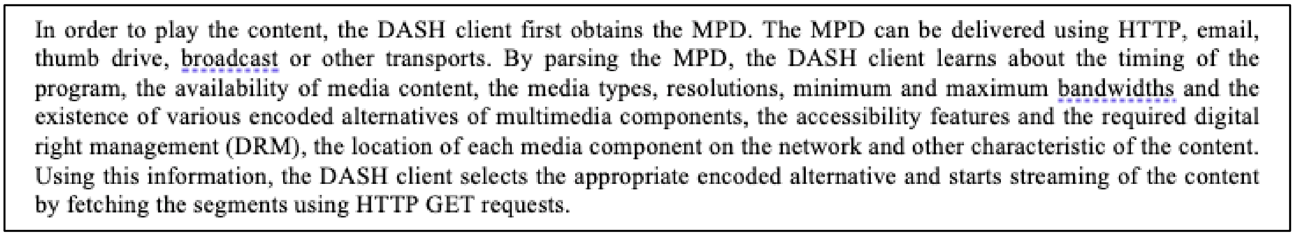


Note that despite the formats are initially designed to be used in the above deployment scenario their application is obviously not restricted to this scenario. The particular aspect on "HTTP" in DASH is the usage of HTTP-URLs in the MPD for the purpose to refer to Segments. The usage of HTTP-URLs enables unique location information and it provides well-defined methods to access the resources, in particular HTTP GET and HTTP partial GET.



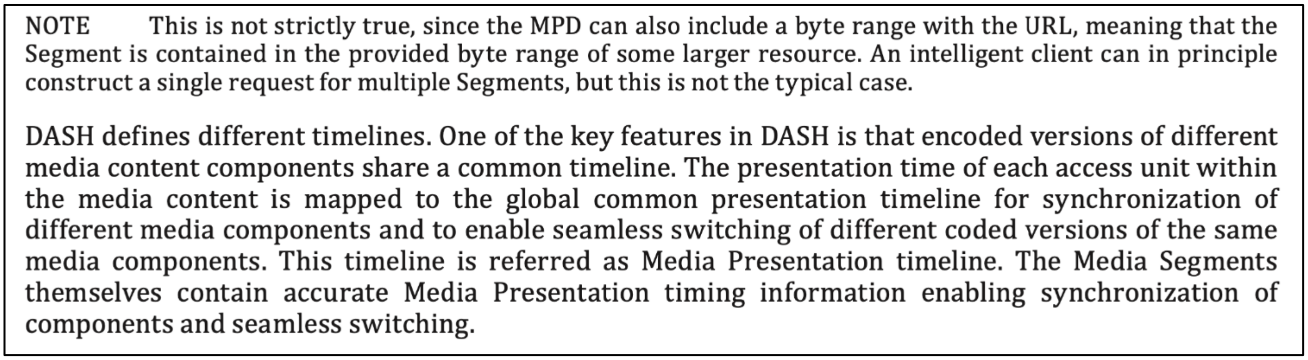
The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent 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.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 8, 9.



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.

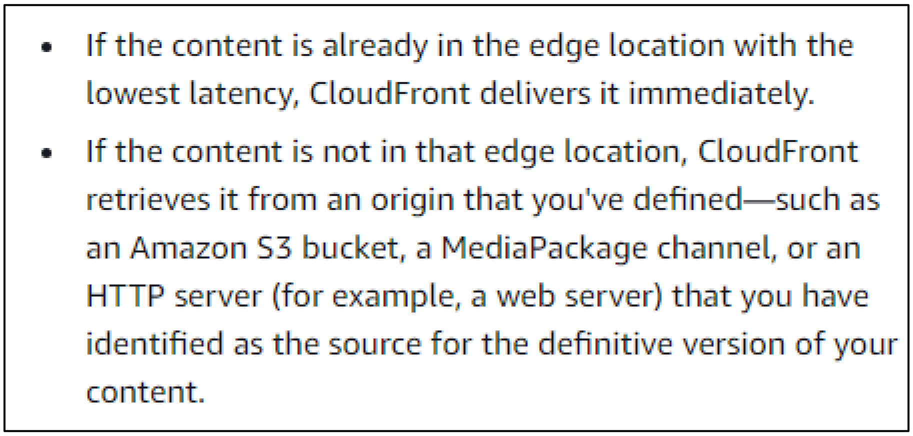
See id. at 3.



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.

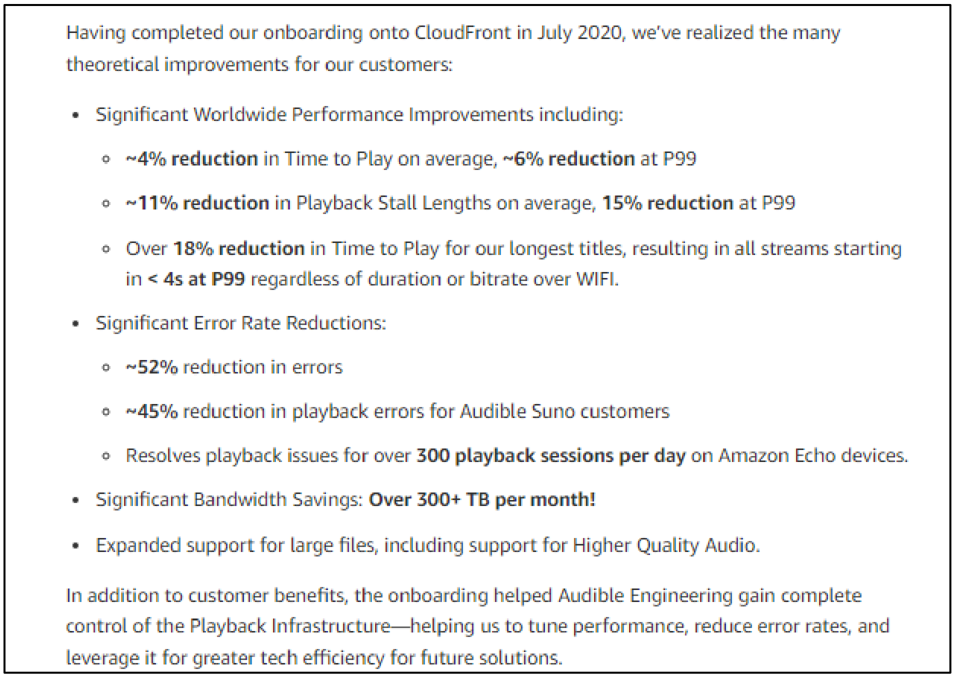
DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different 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 https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9-10.



- If the content is already in the edge location with the lowest latency, CloudFront delivers it immediately.
- If the content is not in that edge location, CloudFront retrieves it from an origin that you've defined—such as an Amazon S3 bucket, a MediaPackage channel, or an HTTP server (for example, a web server) that you have identified as the source for the definitive version of your content.

See e.g.,
https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html.



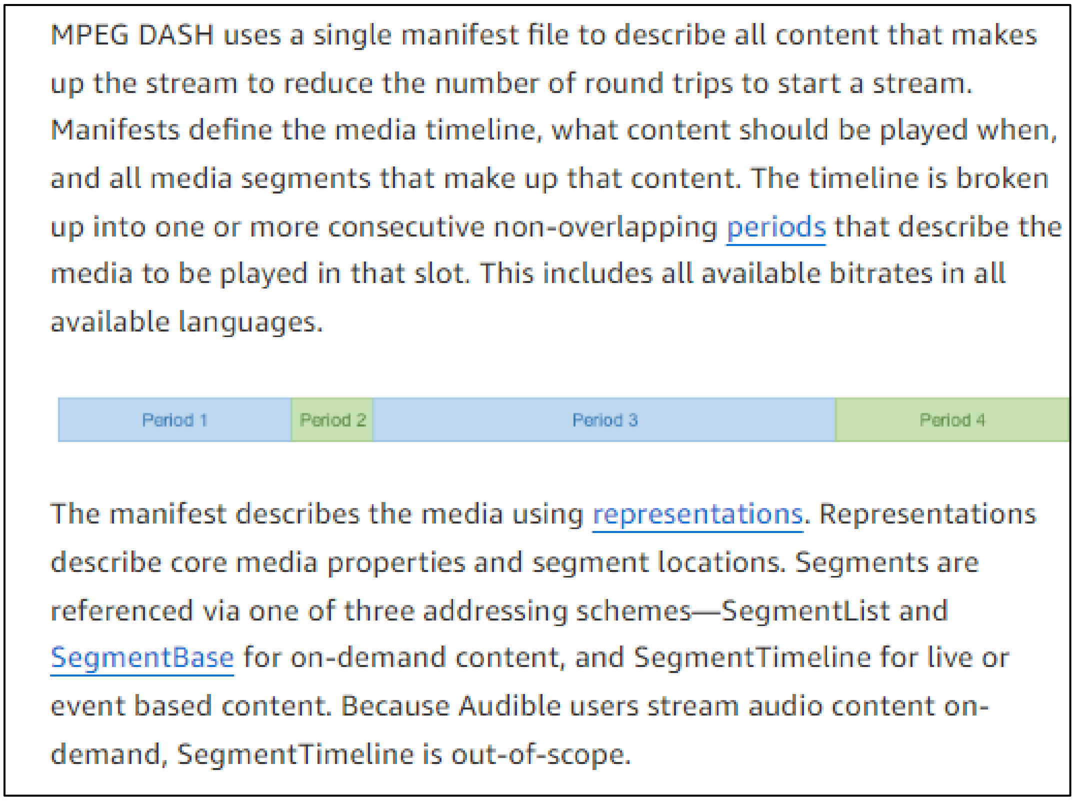
Having completed our onboarding onto CloudFront in July 2020, we've realized the many theoretical improvements for our customers:

- Significant Worldwide Performance Improvements including:
  - **~4% reduction** in Time to Play on average, **~6% reduction** at P99
  - **~11% reduction** in Playback Stall Lengths on average, **15% reduction** at P99
  - Over **18% reduction** in Time to Play for our longest titles, resulting in all streams starting in **< 4s at P99** regardless of duration or bitrate over WIFI.
- Significant Error Rate Reductions:
  - ~**52%** reduction in errors
  - ~**45%** reduction in playback errors for Audible Suno customers
  - Resolves playback issues for over **300 playback sessions per day** on Amazon Echo devices.
- Significant Bandwidth Savings: **Over 300+ TB per month!**
- Expanded support for large files, including support for Higher Quality Audio.

In addition to customer benefits, the onboarding helped Audible Engineering gain complete control of the Playback Infrastructure—helping us to tune performance, reduce error rates, and leverage it for greater tech efficiency for future solutions.

See https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html.

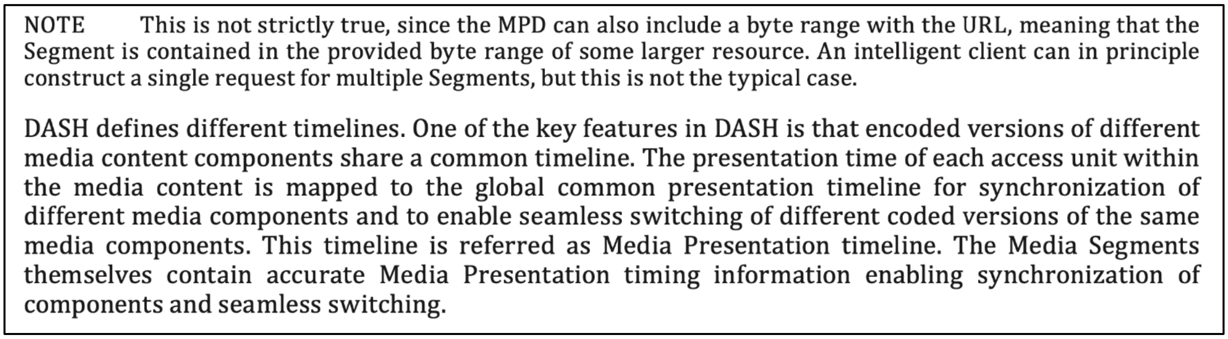
56. On information and belief, one or more components of Amazon Cloudfront Products and Services provide a method comprising identifying a plurality of time offsets (e.g., segment start times and/or end times) in a timeline of the digital audio narration of the first digital media stream, wherein the plurality of time offsets correspond to a plurality of content points in the digital audio narration (e.g., time divisions of content on a timeline of a media stream).



MPEG DASH uses a single manifest file to describe all content that makes up the stream to reduce the number of round trips to start a stream. Manifests define the media timeline, what content should be played when, and all media segments that make up that content. The timeline is broken up into one or more consecutive non-overlapping periods that describe the media to be played in that slot. This includes all available bitrates in all available languages.

The manifest describes the media using representations. Representations describe core media properties and segment locations. Segments are referenced via one of three addressing schemes—SegmentList and SegmentBase for on-demand content, and SegmentTimeline for live or event based content. Because Audible users stream audio content on-demand, SegmentTimeline is out-of-scope.

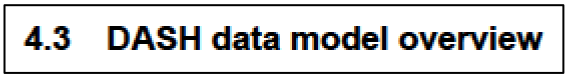
See https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



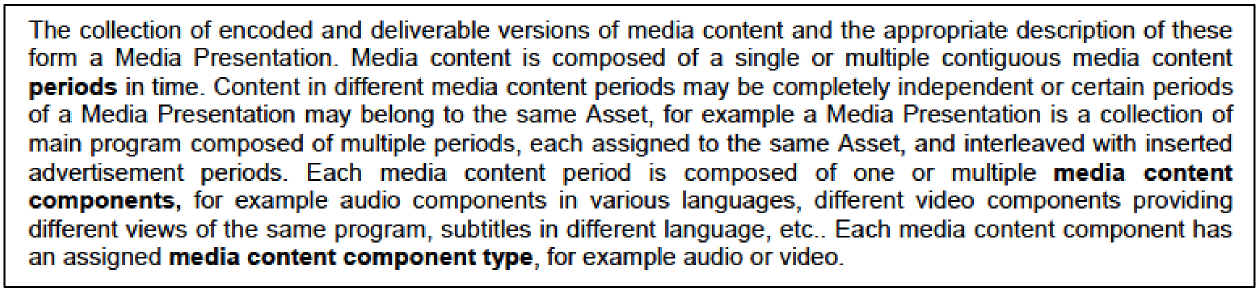
NOTE This is not strictly true, since the MPD can also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client can in principle construct a single request for multiple Segments, but this is not the typical case.

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.

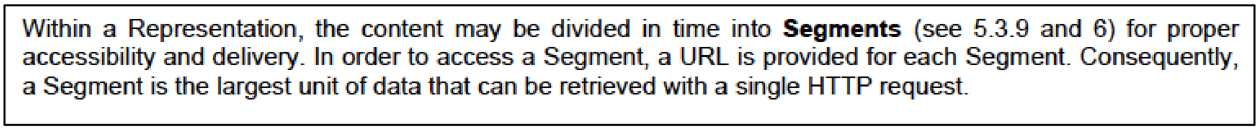
See ISO/IEC 230009-1, Third ed. 2019-08 at 12.



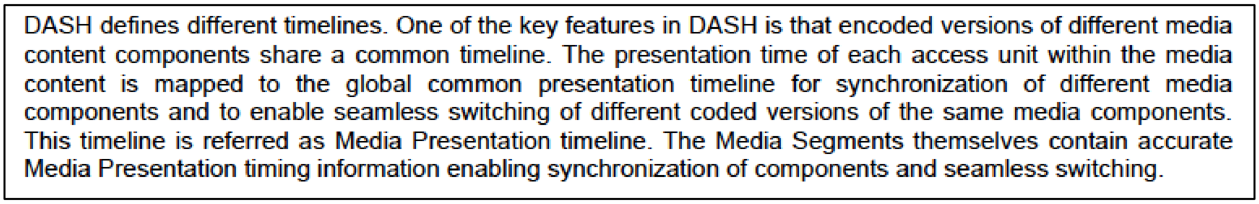
4.3 DASH data model overview



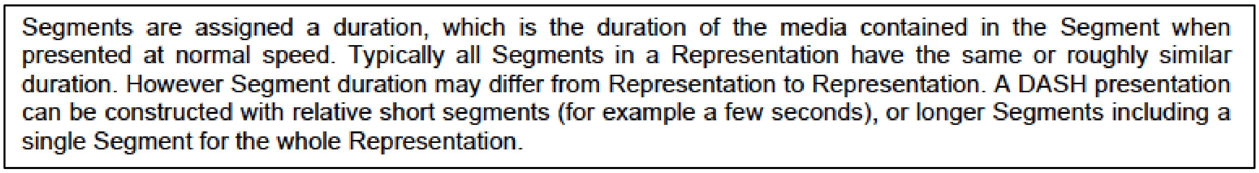
The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent 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.



Within a Representation, the content may be divided in time into **Segments** (see 5.3.9 and 6) for proper accessibility and delivery. In order to access a Segment, a URL is provided for each Segment. Consequently, a Segment is the largest unit of data that can be retrieved with a single HTTP request.



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.



Segments are assigned a duration, which is the duration of the media contained in the Segment when presented at normal speed. Typically all Segments in a Representation have the same or roughly similar duration. However Segment duration may differ from Representation to Representation. A DASH presentation can be constructed with relative short segments (for example a few seconds), or longer Segments including a single Segment for the whole Representation.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9-10.

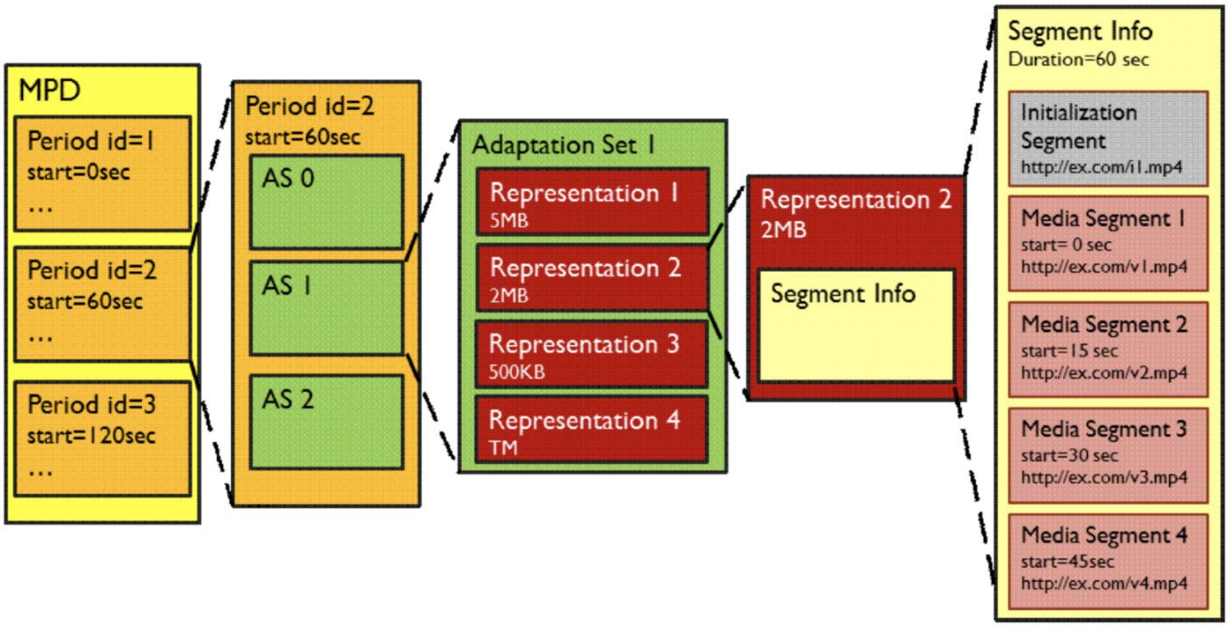
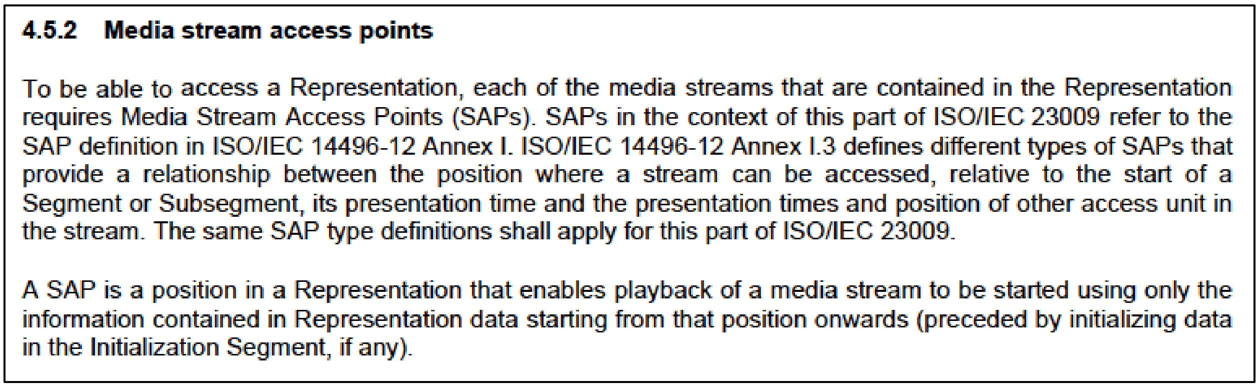


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 3.



4.5.2 Media stream access points

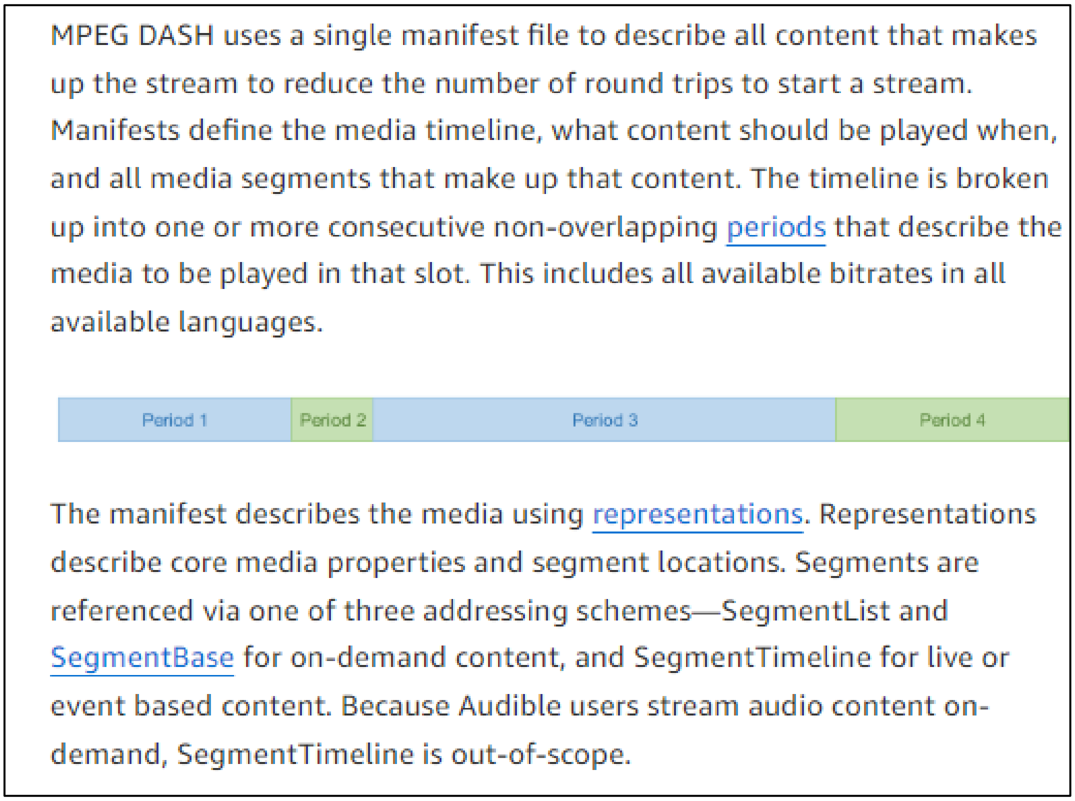
To be able to access a Representation, each of the media streams that are contained in the Representation requires Media Stream Access Points (SAPs). SAPs in the context of this part of ISO/IEC 23009 refer to the SAP definition in ISO/IEC 14496-12 Annex I. ISO/IEC 14496-12 Annex I.3 defines different types of SAPs that provide a relationship between the position where a stream can be accessed, relative to the start of a Segment or Subsegment, its presentation time and the presentation times and position of other access unit in the stream. The same SAP type definitions shall apply for this part of ISO/IEC 23009.

A SAP is a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initializing data in the Initialization Segment, if any).

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 12.

57. On information and belief, one or more components of Amazon CloudFront Products and Services provide a method comprising storing the plurality of time offsets and the plurality of content points in the descriptor file in a manner indicating a correlation between the plurality of time offsets and the plurality of content points (e.g., the segment start times and/or

end times are stored in a MPD, and correspond to the time divisions of content on a timeline of a media stream).

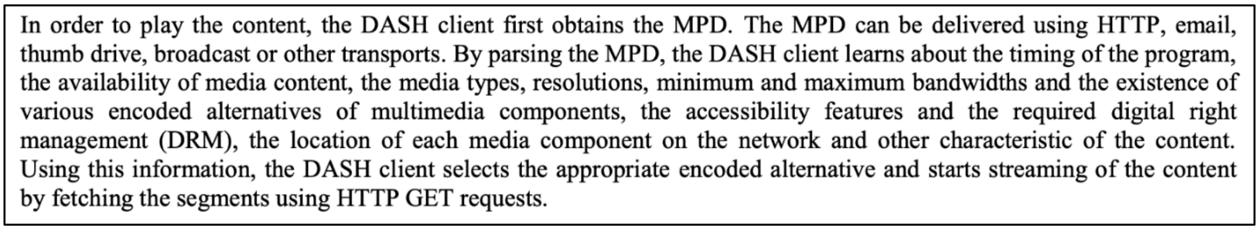


MPEG DASH uses a single manifest file to describe all content that makes up the stream to reduce the number of round trips to start a stream. Manifests define the media timeline, what content should be played when, and all media segments that make up that content. The timeline is broken up into one or more consecutive non-overlapping periods that describe the media to be played in that slot. This includes all available bitrates in all available languages.

The manifest describes the media using representations. Representations describe core media properties and segment locations. Segments are referenced via one of three addressing schemes—SegmentList and SegmentBase for on-demand content, and SegmentTimeline for live or event based content. Because Audible users stream audio content on-demand, SegmentTimeline is out-of-scope.

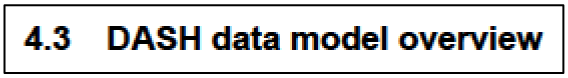
See e.g. https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.

MPEG-DASH: The Standard for Multimedia Streaming Over Internet

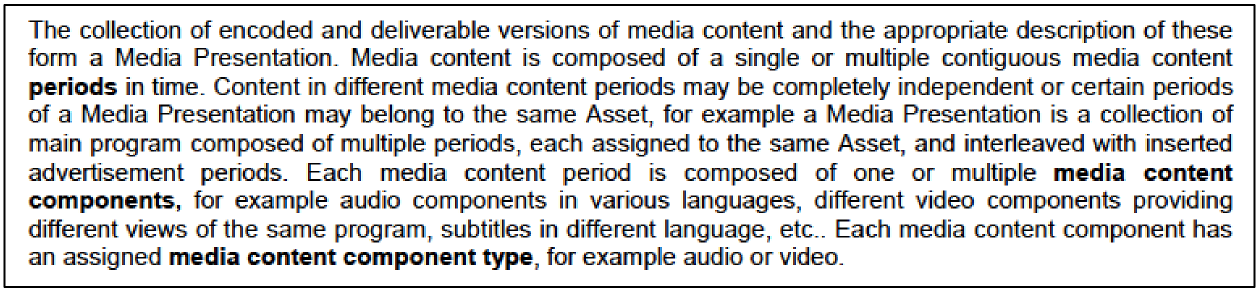


In order to play the content, the DASH client first obtains the MPD. The MPD can be delivered using HTTP, email, thumb drive, broadcast or other transports. By parsing the MPD, the DASH client learns about the timing of the program, the availability of media content, the media types, resolutions, minimum 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.

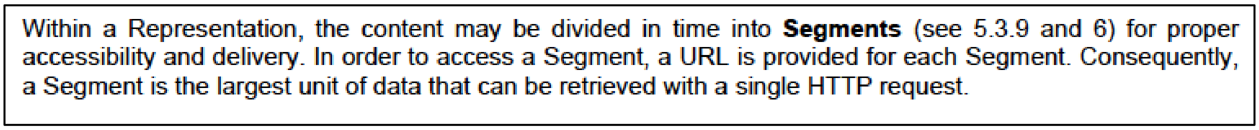
See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 1, 3.



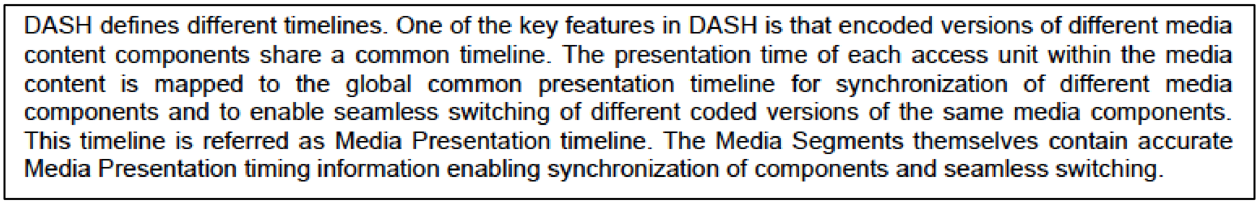
4.3 DASH data model overview



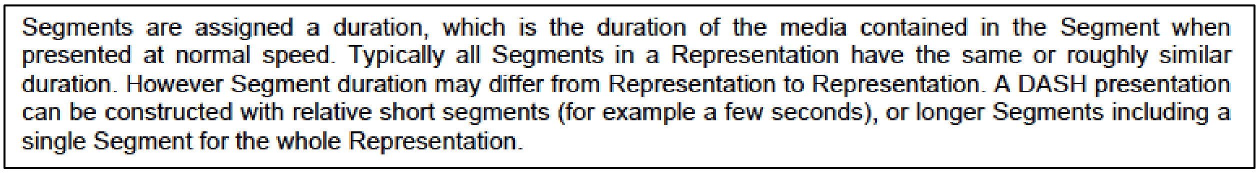
The collection of encoded and deliverable versions of media content and the appropriate description of these form a Media Presentation. Media content is composed of a single or multiple contiguous media content **periods** in time. Content in different media content periods may be completely independent 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.



Within a Representation, the content may be divided in time into **Segments** (see 5.3.9 and 6) for proper accessibility and delivery. In order to access a Segment, a URL is provided for each Segment. Consequently, a Segment is the largest unit of data that can be retrieved with a single HTTP request.

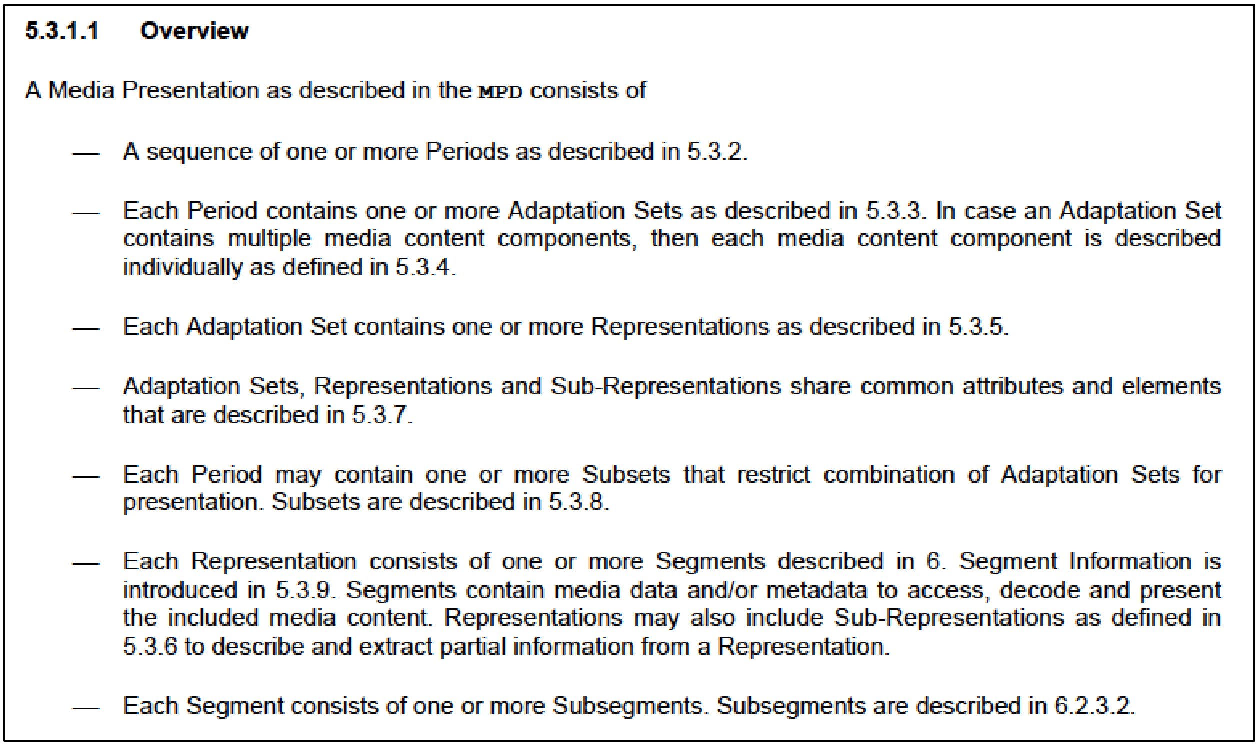


DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different 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.



Segments are assigned a duration, which is the duration of the media contained in the Segment when presented at normal speed. Typically all Segments in a Representation have the same or roughly similar duration. However Segment duration may differ from Representation to Representation. A DASH presentation can be constructed with relative short segments (for example a few seconds), or longer Segments including a single Segment for the whole Representation.

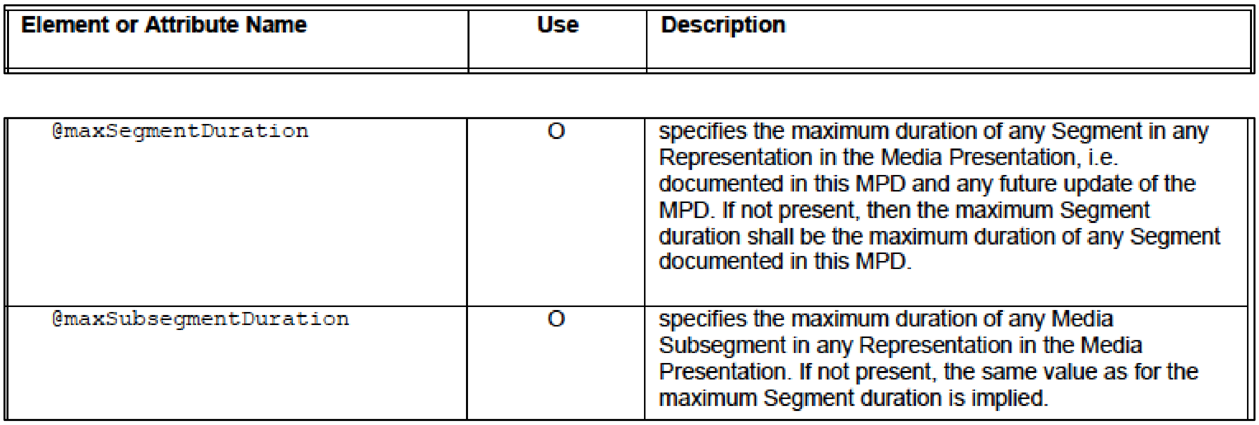
See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9-10.



5.3.1.1 Overview

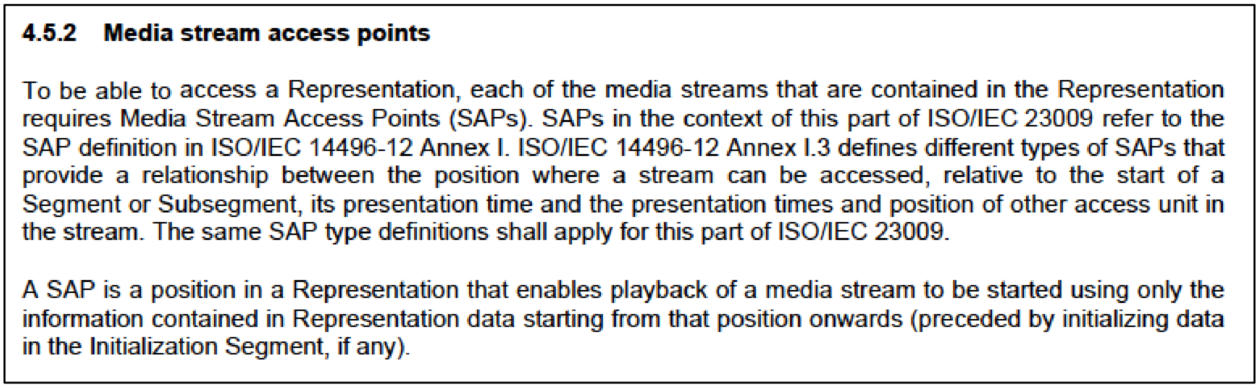
A Media Presentation as described in the MPD consists of

- A sequence of one or more Periods as described in 5.3.2.
- Each Period contains one or more Adaptation Sets as described in 5.3.3. In case an Adaptation Set contains multiple media content components, then each media content component is described individually as defined in 5.3.4.
- Each Adaptation Set contains one or more Representations as described in 5.3.5.
- Adaptation Sets, Representations and Sub-Representations share common attributes and elements that are described in 5.3.7.
- Each Period may contain one or more Subsets that restrict combination of Adaptation Sets for presentation. Subsets are described in 5.3.8.
- Each Representation consists of one or more Segments described in 6. Segment Information is introduced in 5.3.9. Segments contain media data and/or metadata to access, decode and present the included media content. Representations may also include Sub-Representations as defined in 5.3.6 to describe and extract partial information from a Representation.
- Each Segment consists of one or more Subsegments. Subsegments are described in 6.2.3.2.



Element or Attribute Name	Use	Description
@maxSegmentDuration	O	specifies the maximum duration of any Segment in any Representation in the Media Presentation, i.e. documented in this MPD and any future update of the MPD. If not present, then the maximum Segment duration shall be the maximum duration of any Segment documented in this MPD.
@maxSubsegmentDuration	O	specifies the maximum duration of any Media Subsegment in any Representation in the Media Presentation. If not present, the same value as for the maximum Segment duration is implied.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 18, 20.



4.5.2 Media stream access points

To be able to access a Representation, each of the media streams that are contained in the Representation requires Media Stream Access Points (SAPs). SAPs in the context of this part of ISO/IEC 23009 refer to the SAP definition in ISO/IEC 14496-12 Annex I. ISO/IEC 14496-12 Annex I.3 defines different types of SAPs that provide a relationship between the position where a stream can be accessed, relative to the start of a Segment or Subsegment, its presentation time and the presentation times and position of other access unit in the stream. The same SAP type definitions shall apply for this part of ISO/IEC 23009.

A SAP is a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initializing data in the Initialization Segment, if any).

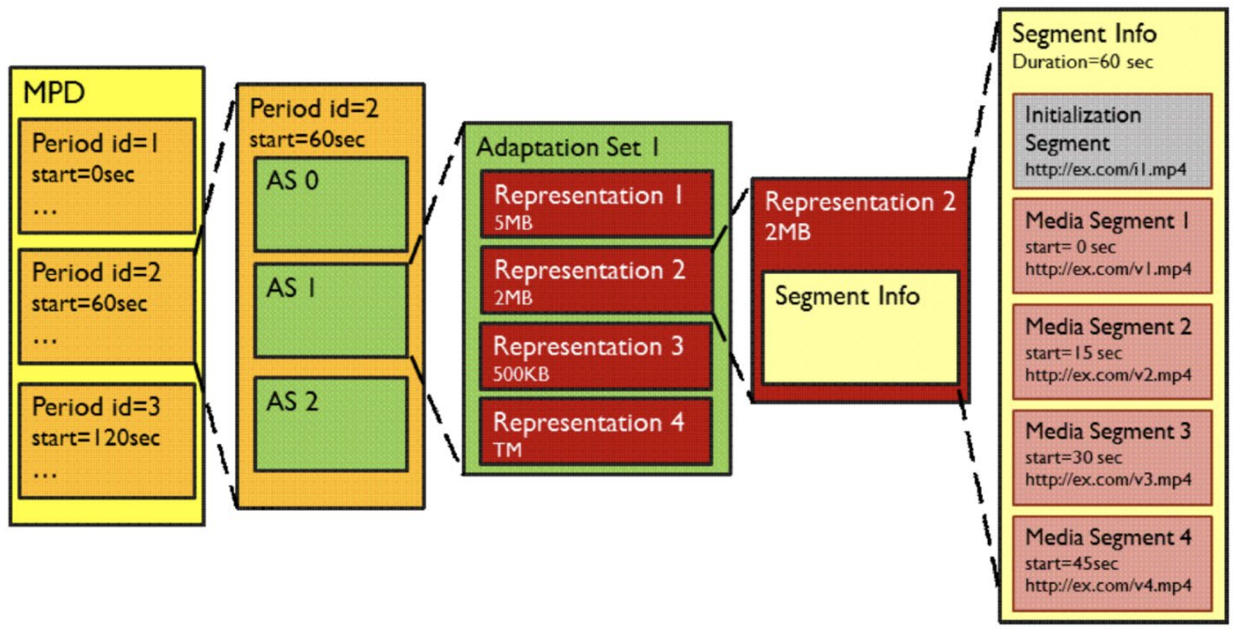
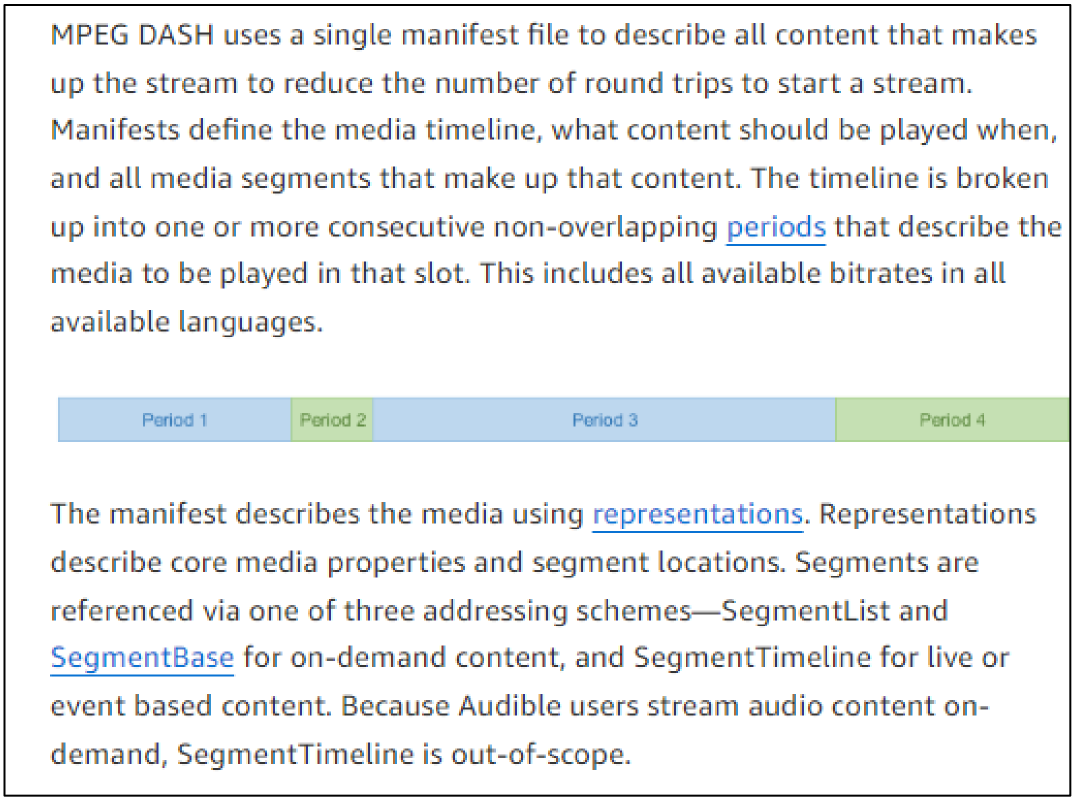


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 3.

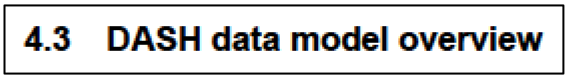
58. On information and belief, one or more components of Amazon CloudFront Products and Services provide a method comprising identifying synchronization points (e.g., media stream access points) in the digital media content of the one or more other digital media streams.



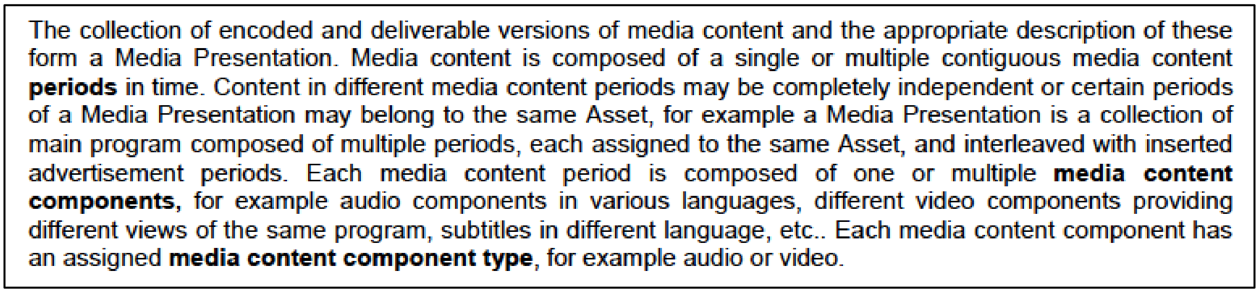
MPEG DASH uses a single manifest file to describe all content that makes up the stream to reduce the number of round trips to start a stream. Manifests define the media timeline, what content should be played when, and all media segments that make up that content. The timeline is broken up into one or more consecutive non-overlapping periods that describe the media to be played in that slot. This includes all available bitrates in all available languages.

The manifest describes the media using representations. Representations describe core media properties and segment locations. Segments are referenced via one of three addressing schemes—SegmentList and SegmentBase for on-demand content, and SegmentTimeline for live or event based content. Because Audible users stream audio content on-demand, SegmentTimeline is out-of-scope.

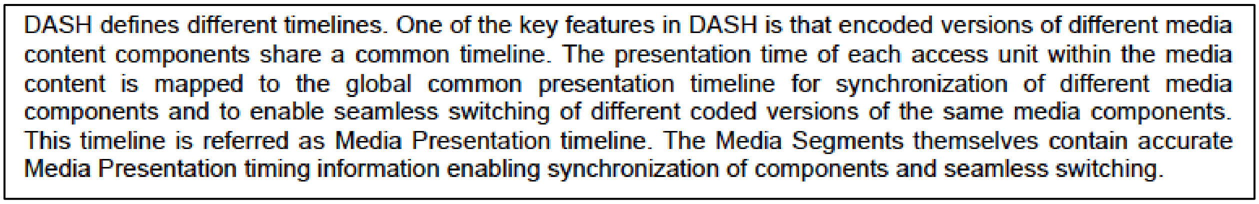
See https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



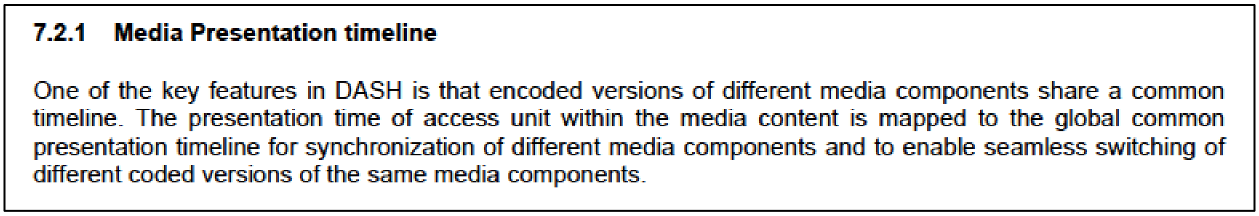
4.3 DASH data model overview



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.



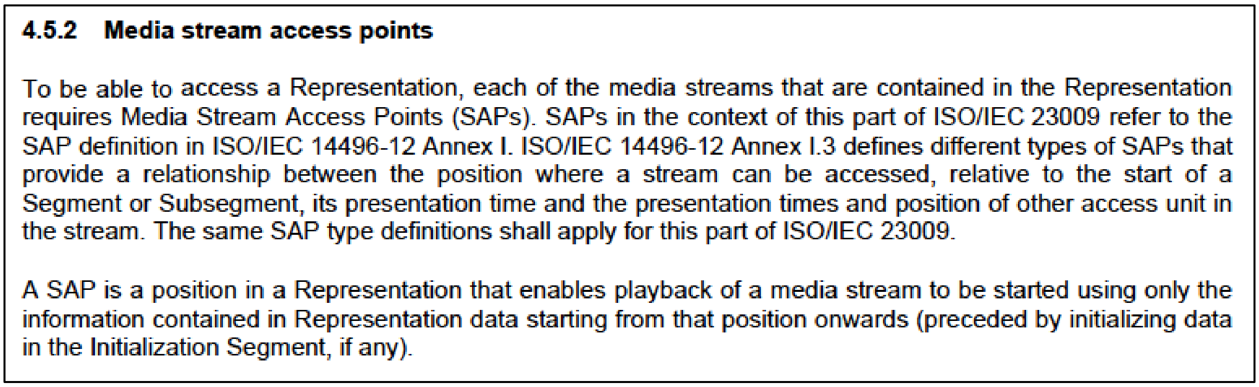
DASH defines different timelines. One of the key features in DASH is that encoded versions of different media content components share a common timeline. The presentation time of each access unit within the media content is mapped to the global common presentation timeline for synchronization of different 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.



7.2.1 Media Presentation timeline

One of the key features in DASH is that encoded versions of different media components share a common timeline. The presentation time of 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.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9, 95.

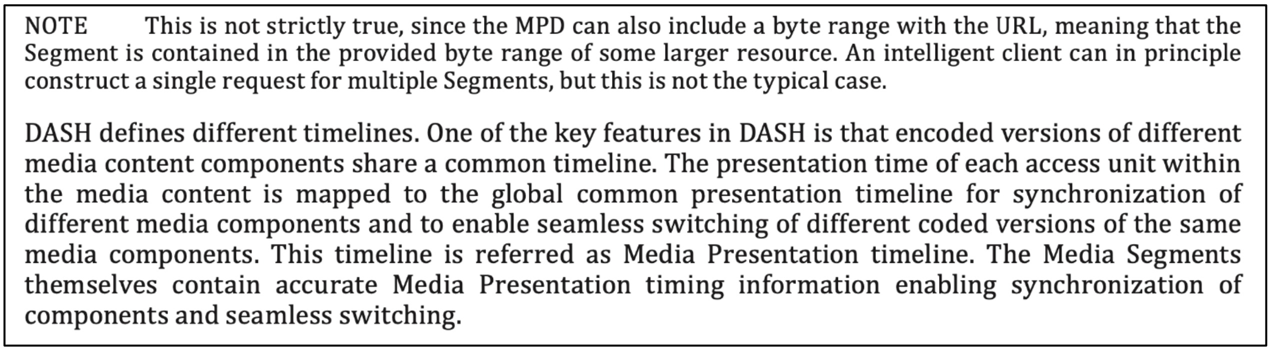


4.5.2 Media stream access points

To be able to access a Representation, each of the media streams that are contained in the Representation requires Media Stream Access Points (SAPs). SAPs in the context of this part of ISO/IEC 23009 refer to the SAP definition in ISO/IEC 14496-12 Annex I. ISO/IEC 14496-12 Annex I.3 defines different types of SAPs that provide a relationship between the position where a stream can be accessed, relative to the start of a Segment or Subsegment, its presentation time and the presentation times and position of other access unit in the stream. The same SAP type definitions shall apply for this part of ISO/IEC 23009.

A SAP is a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initializing data in the Initialization Segment, if any).

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 12.

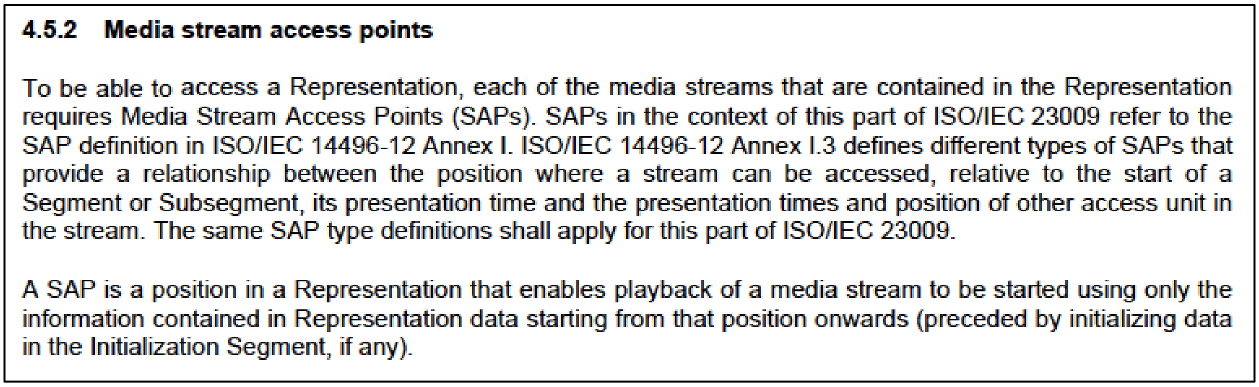


NOTE This is not strictly true, since the MPD can also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client can in principle construct a single request for multiple Segments, but this is not the typical case.

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 https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9-10.

59. On information and belief, one or more components of Amazon CloudFront Products and Services provide a method comprising selecting synchronization time offsets (e.g., a time relationship between a media stream access point and various other time information (e.g., a segment start time, a presentation time, and other access points in the stream)) that correspond to the synchronization points from the plurality of time offsets.

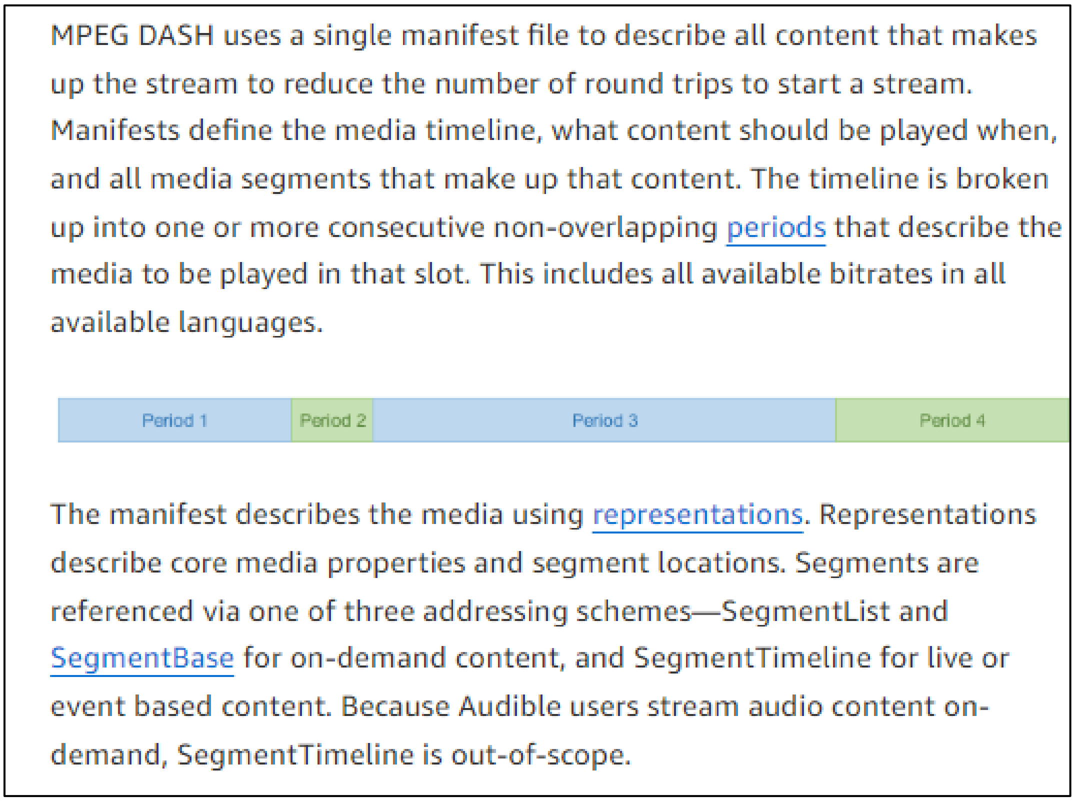


4.5.2 Media stream access points

To be able to access a Representation, each of the media streams that are contained in the Representation requires Media Stream Access Points (SAPs). SAPs in the context of this part of ISO/IEC 23009 refer to the SAP definition in ISO/IEC 14496-12 Annex I. ISO/IEC 14496-12 Annex I.3 defines different types of SAPs that provide a relationship between the position where a stream can be accessed, relative to the start of a Segment or Subsegment, its presentation time and the presentation times and position of other access unit in the stream. The same SAP type definitions shall apply for this part of ISO/IEC 23009.

A SAP is a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initializing data in the Initialization Segment, if any).

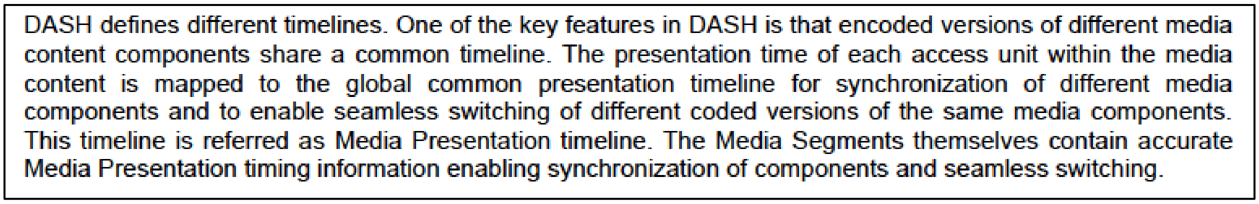
See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 12.



MPEG DASH uses a single manifest file to describe all content that makes up the stream to reduce the number of round trips to start a stream. Manifests define the media timeline, what content should be played when, and all media segments that make up that content. The timeline is broken up into one or more consecutive non-overlapping periods that describe the media to be played in that slot. This includes all available bitrates in all available languages.

The manifest describes the media using representations. Representations describe core media properties and segment locations. Segments are referenced via one of three addressing schemes—SegmentList and SegmentBase for on-demand content, and SegmentTimeline for live or event based content. Because Audible users stream audio content on-demand, SegmentTimeline is out-of-scope.

See https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



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/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf at 9.

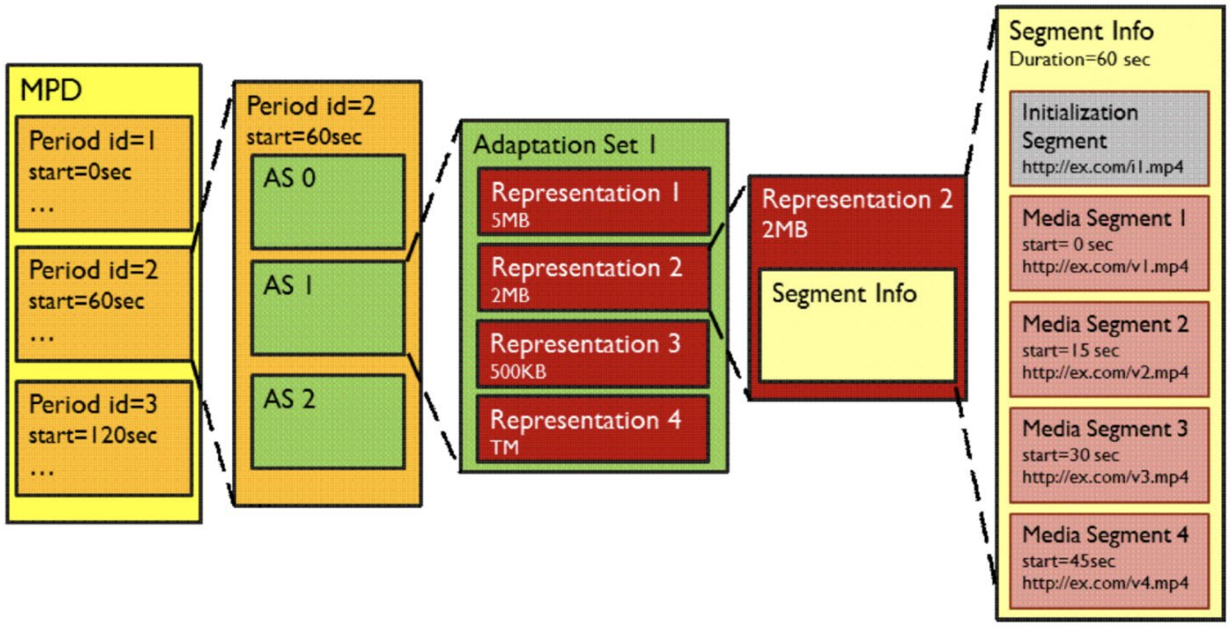
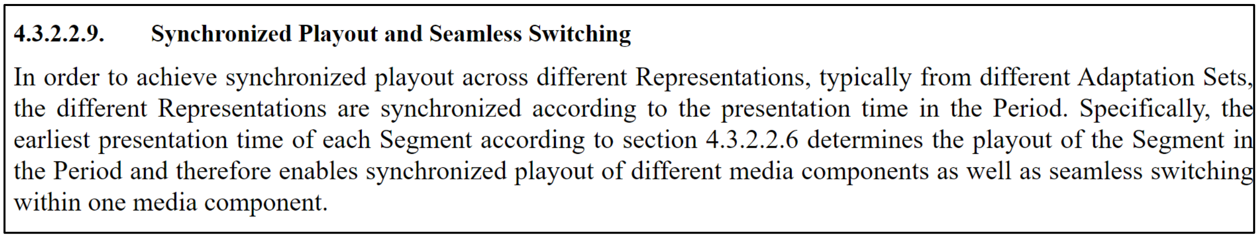


Figure 3. The MPD hierarchical data model.

See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 3.



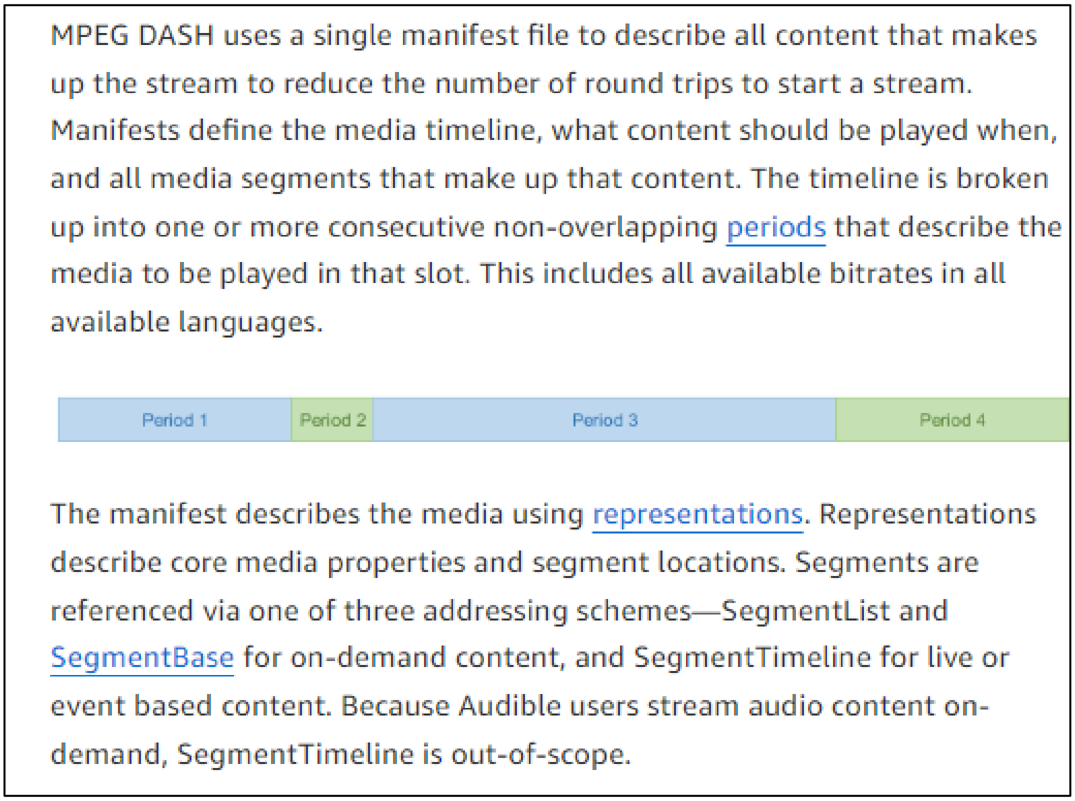
4.3.2.2.9. Synchronized Playout and Seamless Switching

In order to achieve synchronized playout across different Representations, typically from different Adaptation Sets, the different Representations are synchronized according to the presentation time in the Period. Specifically, the earliest presentation time of each Segment according to section 4.3.2.2.6 determines the playout of the Segment in the Period and therefore enables synchronized playout of different media components as well as seamless switching within one media component.

See https://dashif.org/docs/DASH-IF-IOP-v4.2-clean.htm#_Toc511040760.

60. On information and belief, one or more components of Amazon CloudFront Products and Services provide a method comprising storing the synchronization time offsets and the synchronization points in the descriptor file in a manner indicating a correlation between the synchronization time offsets and the synchronization points, such that the descriptor file allows a synchronized rendering of the plurality of digital media streams on a client device (e.g., the (1)

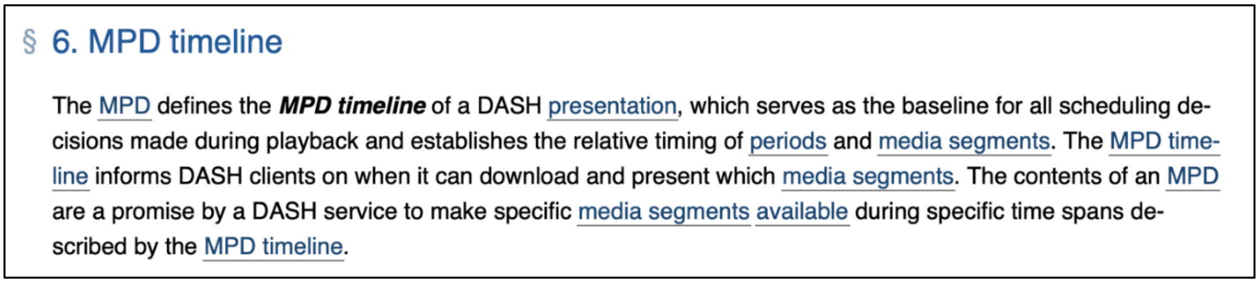
time relationship between a media stream access point and various other time information and (2) media stream access points, are stored in a MPD).



MPEG DASH uses a single manifest file to describe all content that makes up the stream to reduce the number of round trips to start a stream. Manifests define the media timeline, what content should be played when, and all media segments that make up that content. The timeline is broken up into one or more consecutive non-overlapping periods that describe the media to be played in that slot. This includes all available bitrates in all available languages.

The manifest describes the media using representations. Representations describe core media properties and segment locations. Segments are referenced via one of three addressing schemes—SegmentList and SegmentBase for on-demand content, and SegmentTimeline for live or event based content. Because Audible users stream audio content on-demand, SegmentTimeline is out-of-scope.

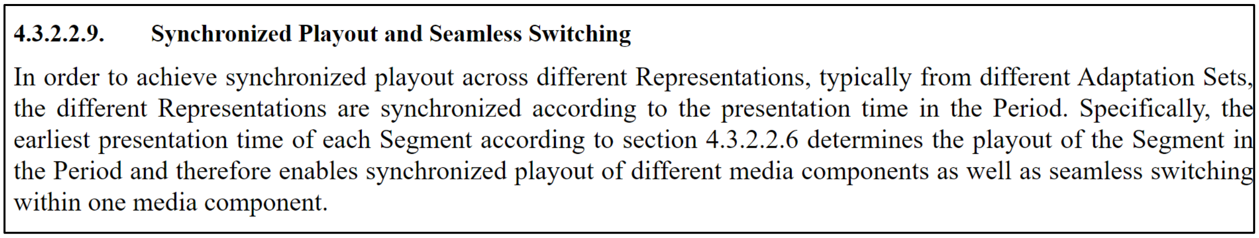
See https://aws.amazon.com/blogs/storage/how-audible-uses-amazon-s3-object-lambda-to-improve-streaming-playback-performance/.



§ 6. MPD timeline

The MPD defines the ***MPD timeline*** of a DASH presentation, which serves as the baseline for all scheduling decisions made during playback and establishes the relative timing of periods and media segments. The MPD timeline informs DASH clients on when it can download and present which media segments. The contents of an MPD are a promise by a DASH service to make specific media segments available during specific time spans described by the MPD timeline.

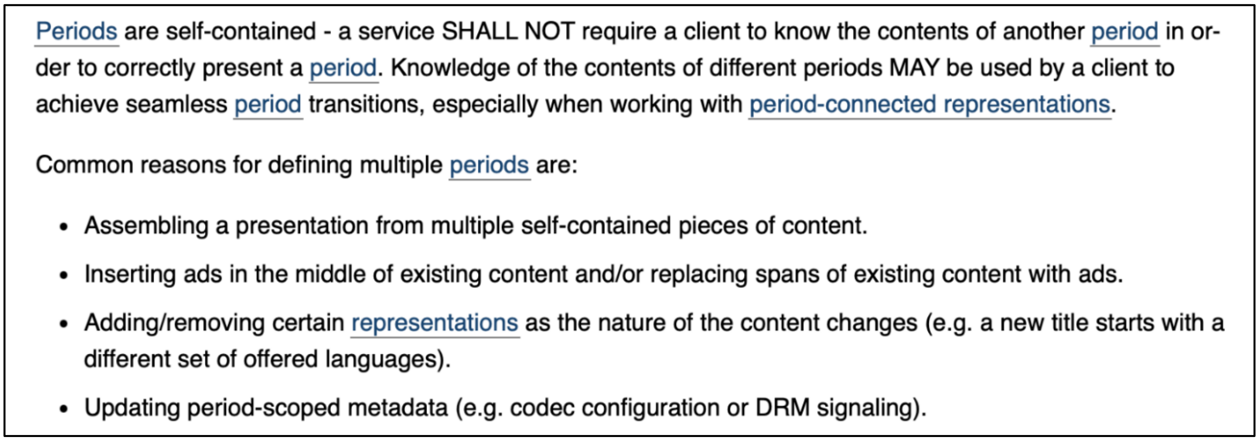
See https://dashif-documents.azurewebsites.net/Guidelines-TimingModel/master/Guidelines-TimingModel.html#mpd-general-timeline.



4.3.2.2.9. Synchronized Playout and Seamless Switching

In order to achieve synchronized playout across different Representations, typically from different Adaptation Sets, the different Representations are synchronized according to the presentation time in the Period. Specifically, the earliest presentation time of each Segment according to section 4.3.2.2.6 determines the playout of the Segment in the Period and therefore enables synchronized playout of different media components as well as seamless switching within one media component.

See https://dashif.org/docs/DASH-IF-IOP-v4.2-clean.htm#_Toc511040760.

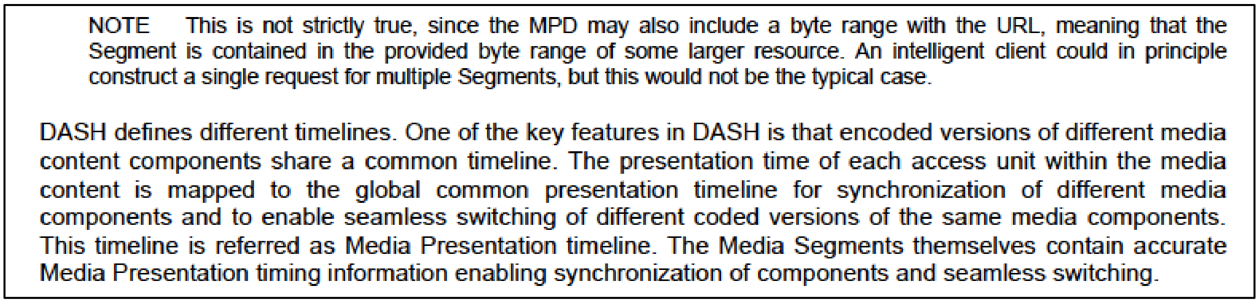


Periods are self-contained - a service SHALL NOT require a client to know the contents of another period in order to correctly present a period. Knowledge of the contents of different periods MAY be used by a client to achieve seamless period transitions, especially when working with period-connected representations.

Common reasons for defining multiple periods are:

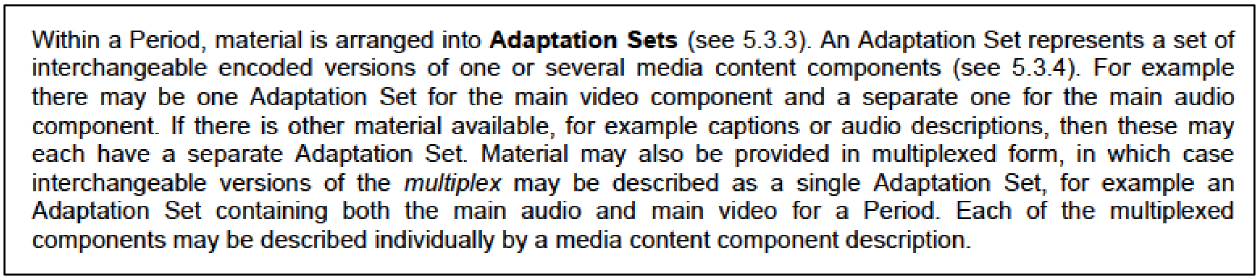
- Assembling a presentation from multiple self-contained pieces of content.
- Inserting ads in the middle of existing content and/or replacing spans of existing content with ads.
- Adding/removing certain representations as the nature of the content changes (e.g. a new title starts with a different set of offered languages).
- Updating period-scoped metadata (e.g. codec configuration or DRM signaling).

See https://dashif-documents.azurewebsites.net/Guidelines-TimingModel/master/Guidelines-TimingModel.html#mpd-general-timeline.



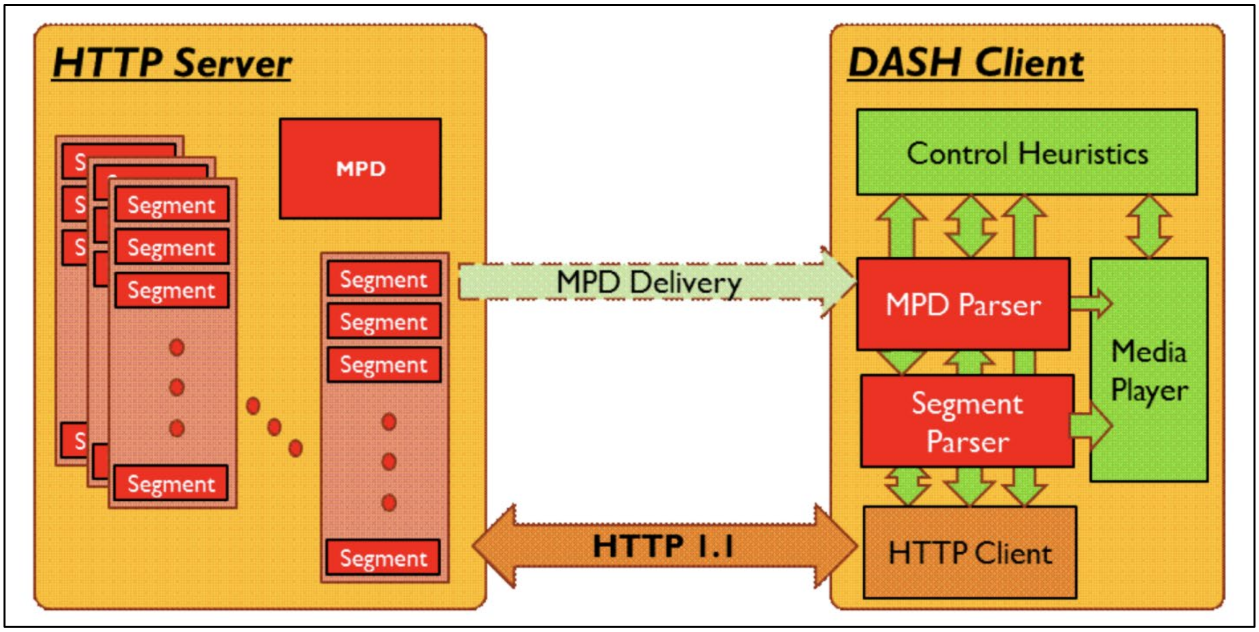
NOTE This is not strictly true, since the MPD may also include a byte range with the URL, meaning that the Segment is contained in the provided byte range of some larger resource. An intelligent client could in principle construct a single request for multiple Segments, but this would not be the typical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



Within a Period, material is arranged into **Adaptation Sets** (see 5.3.3). An Adaptation Set represents a set of interchangeable encoded versions of one or several media content components (see 5.3.4). For example there may be one Adaptation Set for the main video component and a separate one for the main audio component. If there is other material available, for example captions or audio descriptions, then these may each have a separate Adaptation Set. Material may also be provided in multiplexed form, in which case interchangeable versions of the *multiplex* may be described as a single Adaptation Set, for example an Adaptation Set containing both the main audio and main video for a Period. Each of the multiplexed components may be described individually by a media content component description.

See https://github.com/liwf616/awesome-live-stream/blob/master/Ebook/ISO_IEC_23009-1_2014.pdf.



See https://www.bogotobogo.com/VideoStreaming/images/mpeg_dash/DASH-IEEE-multimedia-preprint.pdf at 3.

61. On information and belief, Defendants directly infringe at least claim 1 of the ’907 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 Amazon Cloudfront Products and Services.

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

JURY DEMANDED

63. 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 ’907 patent under at least 35 U.S.C. § 271(a);

b. 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;

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

d. 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

e. 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*