IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

VIDEOLABS, INC., and VL COLLECTIVE IP LLC	
Plaintiffs,	Civil Action No.
v.	JURY TRIAL DEMANDED
NETFLIX, INC.	
Defendant.	

COMPLAINT

Plaintiffs VideoLabs, Inc. ("VL") and VL Collective IP LLC ("VL IP") (collectively "VideoLabs" or "Plaintiffs") file this Complaint against Defendant Netflix Inc. ("Netflix" or "Defendant"), and in support thereof alleges as follows:

NATURE OF THE ACTION

1. Digital video has become fundamental to how society interacts, communicates, educates, and entertains. In fact, video consumption now accounts for more than 82% of all Internet traffic.[1] The ability to reliably provide high-quality video drives the growth of digital platforms that are increasingly integral to the global economy.

2. The advent of high-quality video as a staple of digital consumption did not happen instantaneously. As with any complex technology, digital video presented implementation challenges. Many companies spent many years and resources to develop new and innovative technologies that guide how video is created, streamed, secured, managed, and consumed.

[1] *See* Ex. 1, *The Sustainable Future of Video Entertainment*, INTERDIGITAL (Aug. 2020), https://www.interdigital.com/white_papers/the-sustainable-future-of-video-entertainment?submit_success=true (last visited Jan. 20, 2022).

3. Various inventions and technological advances have transformed digital video. Some of these technologies, such as techniques to efficiently compress video file size, address central challenges to storing and transmitting video. Others enable video content to be efficiently and securely streamed to the many user devices that exist today. Yet others involve managing and organizing videos to provide viewers easier access to content and address how they interact with content. Successful video streaming thus requires myriad technologies that necessarily coordinate with one another.

4. Because various companies played roles in developing the foundational technology for today's digital video, no single company can provide the high-quality video experiences that consumers have come to expect without using technology owned by other companies.

5. The founders of VideoLabs recognized this problem and understood that collective action was needed to address it. If the companies that developed critical video technologies worked together, everyone could benefit: innovators could receive fair compensation for their contributions, companies deploying video technology could respect the innovators' patents and license them on affordable and predictable terms, and consumers could experience better and more affordable video technology.

6. In 2019, with support from widely-recognized industry leaders, VideoLabs launched a platform to achieve these goals. VideoLabs spent millions of dollars and thousands of hours analyzing the video space and identifying the patents that reflect the innovations with the highest impact. VideoLabs then compiled a portfolio of these core patents, obtaining them from leading companies, including Hewlett Packard Enterprise, Alcatel-Lucent S.A., Siemens AG, Swisscom AG, 3Com, Panasonic, LG, and Nokia.

7. VideoLabs then opened-up membership on its platform to all willing companies.

In exchange for low-cost membership or licensing fees, VideoLabs provides access to its patent portfolio and a commitment to seek out the most important patents in the video industry and clear them. Many prominent companies recognized the benefits of the VideoLabs platform and worked with VideoLabs to efficiently and responsibly license its video technology patents.

8. Unfortunately, Netflix has not. Netflix is one of the world's largest users of video technologies and operates the world's most popular streaming TV service with over 213 million subscribers.[2] It is enmeshed in practically every aspect of video, from creation to processing, delivery, and display.[3]

9. VideoLabs contacted Netflix multiple times to offer Netflix the benefit of VideoLabs' platform and to alert it to its use of VideoLabs' patented technology. As an added incentive for engaging in good faith discussion, VideoLabs offered to conduct discussions with Netflix under an NDA that would eliminate any legal risk from participating in the discussions, including a 3-month mutual legal standstill which would provide sufficient time for full and open dialogue. After many months of ignoring VideoLabs' entreaties, Netflix finally responded. But Netflix foreclosed the possibility of good faith discussions by insisting that VideoLabs agree not

[2] *See, e.g.*, Ex. 2 at 3, https://www.cnbc.com/2021/11/10/disney-netflix-and-other-streaming-services-subs-arpu-q3-2021.html#:~:text=Netflix%20continues%20to%20outpace%20the,around%20paying%20customers%20and%20ARPU; Ex. 3 at 6-7 https://screenrant.com/ten-most-popular-streaming-services-ranked-subscriber-numbers/; Ex. 4 at 3-4, https://www.businessofapps.com/data/netflix-statistics/.

[3] *See, e.g.*, Ex. 5 at 1, https://www.statista.com/statistics/883491/netflix-original-content-titles/; Ex. 6 at 1-3, https://netflixtechblog.com/high-quality-video-encoding-at-scale-d159db052746; Ex. 7 at 1-9, https://www.comparitech.com/blog/vpn-privacy/netflix-statistics-facts-figures/; Ex. 8 at 1-9, https://netflixtechblog.com/ava-the-art-and-science-of-image-discovery-at-netflix-a442f163af6; Ex. 9 at 1-6, https://www.webdesignerdepot.com/2020/02/3-lessons-ux-designers-can-take-from-netflix/; Ex. 10 at 1-4, https://uxmag.com/articles/how-netflix-uses-psychology-to-perfect-their-customer-experience; Ex. 11 at 1-12, https://uxplanet.org/the-netflix-conundrum-overcoming-the-paradox-of-choice-a-ux-case-study-95b19acdc28c.

to enforce its patent rights against Netflix for an indefinite period of time. When VideoLabs declined this demand, Netflix ceased responding to any of VideoLabs' communications.

10. Faced with this Hobson's choice, VideoLabs feels that it has no recourse but to file this action to stop Netflix's unauthorized use of VideoLabs' patents. Failure to take action would undermine the viability of VideoLabs' platform and permit further free-riding by Netflix of the significant innovations of VideoLabs' patents.

11. This case is ultimately about ensuring the integrity of the patent system and compensating patent owners for their protected innovations. Respect for intellectual property, as the law requires, is essential to incentivize innovation and promote technological progress. Accordingly, VideoLabs brings this action under the patent laws, 35 U.S.C. § 1 *et seq.*, in order to stop Netflix's willful infringement of U.S. Patent Nos. 8,139,878, 7,440,559, and 7,233,790 (collectively, "patents-in-suit").

THE PARTIES

12. VL was founded in 2018 as part of an industry-sponsored and -funded effort to reduce the cost and risk of technological gridlock associated with diverse patent ownership. VL's leadership has decades of experience in intellectual property licensing, during which they have completed over 1,000 intellectual property transactions worldwide and drawn more than $6 billion in revenue.

13. VL is a corporation organized under the laws of the State of Delaware, with its principal place of business in Palo Alto, California.

14. VL IP was founded in 2019 as a subsidiary of VideoLabs, Inc.

15. VL IP is a corporation organized under the laws of the State of Delaware, with its principal place of business in Palo Alto, California.

16. On information and belief, Netflix is a publicly traded corporation organized and

existing under the laws of the State of Delaware and is registered to do business in the State of Delaware. Netflix's headquarters are located at 100 Winchester Circle, Los Gatos, California 95032.

JURISDICTION AND VENUE

17. This is an action for patent infringement arising under the patent laws of the United States. This Court has jurisdiction over the subject matter of this action under 28 U.S.C. §§ 1331 and 1338(a), 15 U.S.C. § 1121, and 28 U.S.C. § 1367(a).

18. This Court has personal jurisdiction over Netflix because, on information and belief, Netflix conducts business in and has committed acts of patent infringement in this District, and has established minimum contacts with this forum state such that the exercise of jurisdiction over Netflix would not offend traditional notions of fair play and substantial justice. Netflix is incorporated in this District. On information and belief, Netflix offers products and/or services, including those accused herein of infringement, to customers and potential customers located in this District.

19. Venue is proper in this Court under 28 U.S.C. §§ 1391 and 1400(b). Netflix resides in this District. Netflix has chosen to incorporate in the state of Delaware, thereby receiving the benefits offered to Delaware corporations. Netflix must accordingly assume responsibilities to Delaware and its citizens.

20. Further, on information and belief, Netflix has offered and sold, and continues to offer and sell, its infringing products and services in this District. On information and belief, Netflix designs, uses, distributes, sells, and/or offers to sell the infringing products and services to consumers and businesses in this District.

21. On information and belief, Netflix is a corporation with global reach and annual revenue in the billions of dollars. Netflix accordingly cannot reasonably claim it would be

inconvenient to litigate in the forum in which it is incorporated.

22. Moreover, litigating in this District is convenient and would serve the interests of judicial economy because of a related pending lawsuit in this District.[4]

THE VIDEOLABS PATENTS-IN-SUIT

A. U.S. Patent No. 8,139,878

23. U.S. Patent No. 8,139,878 (the "'878 Patent"), titled "Picture Coding Method and Picture Decoding Method," issued on March 20, 2012. VL owns all rights and title to the '878 Patent, as necessary to bring this action. A true and correct copy of the '878 Patent is attached as Exhibit 12.

24. The '878 Patent was developed by engineers at Panasonic, one of the largest consumer electronics companies at the time of the invention and a major innovator in Internet technologies. In 2002, when patent applications were first filed for the '878 Patent, Panasonic was a world leader in digital video technologies.[5] Panasonic developed video coding technologies and designed consumer electronics — including TVs, DVD players, and memory cards — for storing, processing, and displaying video content.[6] The inventions of the '878 Patent are the result of years of research by teams of Panasonic engineers working at the cutting edge of video processing and encoding.

25. Native video files are massive. Modern digital video cameras used by premier television and movie studios capture images at incredibly fast rates (ranging from 30 frames per

[4] *See Starz Entertainment, LLC v. VL Collective IP, LLC*, 1-21-cv-01448 (D. Del. filed Oct. 13, 2021).

[5] *See, e.g.*, Ex. 13 at 6, 10-17, 41, *Annual Report 2002*, National/Panasonic Matsushita Electric, available at https://www.annualreportowl.com/Panasonic/2002/Annual%20Report (last accessed Jan. 20, 2022).

[6] *See id.*

second up to 300 frames per second) and extremely high resolutions (up to "5k," or 5120 x 2880, for a total size of 14,745,600 pixels per frame). Storing just an hour of this raw content requires more than 300 GB of memory.[7] Most modern TVs, laptops, tablets, and smartphones cannot possibly store and play such large files.

26. Even if they could, there would be little point from the perspective of on-demand content delivery: Internet speeds are far too slow to stream such massive video files. The fact is that transmitting high quality audiovisual content is simply not possible without powerful compression technologies. Streaming even just standard high-definition content (720p) requires network bandwidth of approximately 1.5 Gbps,[8] which is about 35 times faster than the average Internet speed in the United States.[9] "Encoding" and "decoding," which respectively refer to the processes of compressing and decompressing content, are thus essential to applications such as video streaming, digital television, and videoconferencing.

27. Encoding video content allows the content to be made small for storage and transmission, while decoding permits the viewer to watch high-quality content on his or her device. In addition to making real-time streaming of content possible, every incremental increase in compression efficiency yields substantial benefits to companies that store, process, transmit, or access video. For example, if a video streaming company can cut the size of each of its movie files in half, then it reasons that it only needs half the numbers of servers to store its movies, half

[7] *See* Ex. 14, *How Many GB Is a 2 Hour 4k Movie?*, https://gamingsection.net/news/how-many-gb-is-a-2-hour-4k-movie/ (last visited Jan. 20, 2022).

[8] *See* Ex. 15, Bryan Samis, *Back to Basics: GOPs Explained*, AWS MEDIA BLOG (May 28, 2020), https://aws.amazon.com/blogs/media/part-1-back-to-basics-gops-explained/ (last visited Jan. 20, 2022).

[9] *See* Ex. 16, *Average U.S. Internet Speed is 42.86 Mbps*, ETI (Feb. 2, 2021), https://etisoftware.com/resources/blog/report-average-u-s-internet-speed-is-42-86-mbps/ (last visited Jan. 20, 2022).

the network bandwidth to transmit its movies, and half of all other related expenses, such as energy costs and staffing resources.

28. The ’878 Patent describes breakthrough techniques for encoding and decoding audiovisual content so that it can be transmitted and stored with fewer resources. The patent vastly improves upon existing methods, and the core technology it describes has been used throughout the industry for years as the gold standard for coding video.

1. Background On Coding Technology

29. Video “coding” refers to both the encoding and decoding of video content. Video compression techniques minimize the size of the data that is sent between the encoder and the decoder by removing redundancies and imperceivable changes and then efficiently representing the remaining data for transmission.

30. Video is comprised of a series of frames. These frames are successively output to create the moving pictures that we recognize as video.

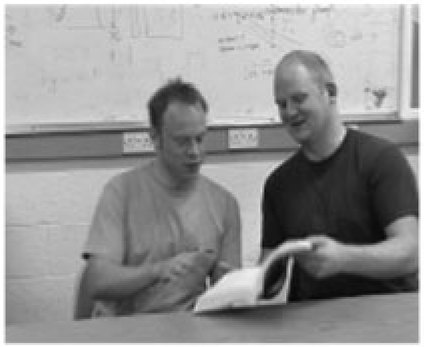


Figure 3.10 Frame 1

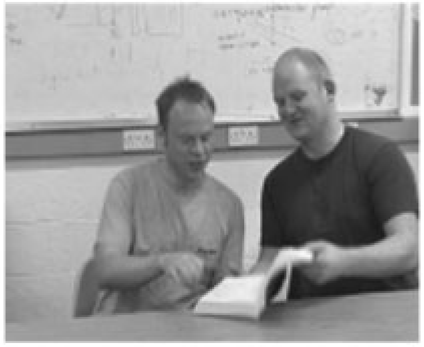


Figure 3.11 Frame 2

Ex. 17, Iain E. Richardson, *The H.264 Advanced Video Compression Standard* (2d. ed. 2010) (hereinafter “Richardson”), at 33.

31. In the early 2000s, certain techniques existed to reduce the amount of data needed to describe each frame without any loss in picture quality. For example, if there are a series of 50 white pixels in a row followed by 75 green pixels, then it is more efficient to store the fact that

there are 50 white pixels followed by 75 green pixels than to store the value of all 125 pixels. This algorithm, which reduces the redundancy stemming from repeating pixels within a frame, yielded substantial benefits.

32. Video engineers also realized that, very often, not much changes between successive frames. In the images shown above, for example, the changes between frames 1 and 2 are largely concentrated in the area near the book. As a result, it is not necessary to send the complete data for every frame of a video. Instead, frames can be sent periodically at strategic points, such as when there is a scene change that creates major differences between successive frames. Those strategic frames — called "key frames" — could be used to "predict" other frames nearby in time by analyzing each frame and storing the differences from one frame to the next.

33. Further research yielded additional advances in what became known as predictive coding. Video engineers realized that it was advantageous to divide each frame into blocks, as shown below.

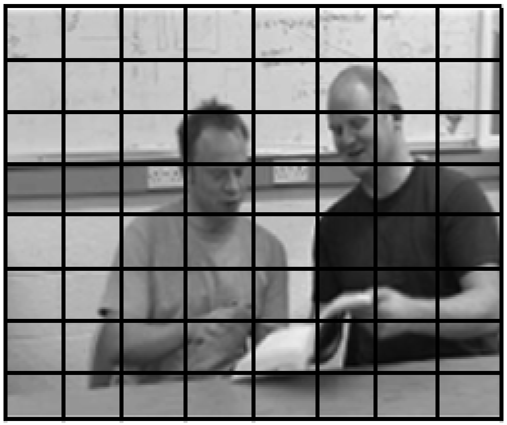


Figure 3.10 Frame 1

34. These blocks could be analyzed and used to predict the pixels in the same block in *surrounding* frames ("*inter*-picture prediction," also called "temporal compression"). Additionally, these blocks could be analyzed to predict the pixels in surrounding blocks in the

same frame ("*intra*-picture prediction," also called "spatial compression"). While predictive coding does not always recreate frames that are identical to the original frames, the differences are so minor as to be imperceptible. For example, in the middle of an intense action sequence, a frame might display a pixel as blue even though it should be green because doing so enables the image to be represented more efficiently. This minor alteration from the original content will go unnoticed by the viewer, who is distracted by all the activity.

35. Once redundancy in the video content has been minimized and imperceptible details have been streamlined, a process called "entropy encoding" further compresses the data by using as few bits to represent the data as possible, while still ensuring fidelity to the original visual content. This is achieved by allocating the fewest bits to commonly appearing bit sequences, and the most bits to infrequently occurring bit sequences. By way of analogy, when training your dog, the commands you use most frequently are likely the shortest, single-word commands, like "sit" and "no." But commands that you need less frequently may be longer, such as "wait for it" and "roll over." In this way, over the course of a week, you expend fewer (verbal) resources. Entropy encoding applies this same principle to the bits of data that comprise video content.

36. There are standardized ways to represent sequences of bits, and depending on the type of entropy coding, these sequences are stored in either "coding tables" or "probability tables." Entropy coding involves selecting the optimal table for the data being transmitted and ensuring that the decoder knows the proper table to use when decoding the data.

37. It was in this context that the inventors of the '878 Patent made their contributions.

2. The '878 Patent

38. The '878 Patent is directed to encoding audio and video content. With respect to video, the '878 Patent describes a type of coding called "Context-based Adaptive Variable Length Coding," or "CAVLC." *See, e.g.*, Ex. 12, '878 Patent at col. 1, ll. 49-52. Content encoded using

the techniques of the ’878 Patent would then be stored or transmitted before ultimately being decoded for playback.

39. When encoded, the image data in a particular image block is represented by, among other things, its “coefficients.” *Id.* at col. 1, ll. 63-67; col. 7, ll. 38-43; col. 21, ll. 60-66; col. 25, ll. 29-36. Roughly speaking, larger coefficients for a block indicate a larger amount of changes in that block as compared with a reference block. *See id.* For many blocks, there are no such changes, and so all the coefficients have a value of zero. *See id.* at col. 21, ll. 60-66. The inventors of the ’878 Patent recognized that these “zero-coefficient” blocks presented an opportunity for further compression. *See*, *e.g.*, *id.* at col. 1, ll. 49-52.

40. They realized that the decoder did not need to know every single time a zero-coefficient block existed; rather, the decoder needs to know only when blocks have *non-zero* coefficients. They devised a technique wherein data about zero-coefficient blocks are effectively not encoded at all, and only non-zero coefficient block data is stored and transmitted. *See*, *e.g.*, *id.* at col. 1, ll.49-52, 56-62; col. 1, l. 65 – col. 2, l. 10. The inventors thereby achieved nearly perfect compression for these zero-coefficient blocks by communicating them practically without sending any information whatsoever. *See id.* at col. 2, ll.11-14.

41. The inventors also made a substantial contribution to the efficiency of entropy coding. They recognized that the coefficients in neighboring blocks were a good predictor of the coefficients in the block being analyzed, and so could be used to select the optimal coding table for the block, yielding enhanced compression. *See, e.g.*, *id.* at col. 9, ll. 34-37; col. 13, ll. 4-11. Prior techniques lacked this level of sophistication. They did not take advantage of the predictive power provided by analyzing the coefficients of the surrounding blocks. They would also use the same coding table for both inter- and intra-predictive coding, which was inefficient because there

could be significant differences between neighboring blocks in the current frame and blocks in subsequent frames. *See, e.g.*, *id.* at col. 1, ll. 33-38. Due to these limitations in the use of coding tables, compression efficiency in previously known entropy coding techniques would vary significantly between different types of content, and generally decreased as the quality of content increased. *Id.* at col. 1, ll.39-44. These problems (and others) were overcome by the inventors of the ’878 Patent.

42. The innovations of the ’878 Patent provided a significant advance in compression that was recognized throughout the industry. In fact, the compression techniques of the ’878 Patent are used in the ubiquitous video codec, H.264. H.264 was revolutionary in the video industry, as it provided a quantum leap of improvement over the video codecs that had previously been commonly used, such as Motion JPEG video and MPEG-2. In particular, H.264 "has an 80% lower bitrate than Motion JPEG video" and "the bitrate savings can be as much as 50% or more compared to MPEG-2."[10]

B. U.S. Patent No. 7,440,559

43. U.S. Patent No. 7,440,559 (the "’559 Patent"), titled "System and Associated Terminal, Method and Computer Program Product for Controlling the Flow of Content," issued on October 21, 2008. VL IP owns all rights and title to the ’559 Patent, as necessary to bring this action. A true and correct copy of the ’559 Patent is attached as Exhibit 18.

44. The original assignee of the ’559 Patent is Nokia Corporation, one of the largest consumer electronics and information technology companies in the world at the time of the invention and a major innovator of digital communications technologies. In 2003, the year in

[10] *See* Ex. 19, *What is H264 Encoding?*, BlackBox, https://www.blackbox.co.uk/gb-gb/page/38313/Resources/Technical-Resources/Black-Box-Explains/Multimedia/What-is-H264-video-encoding/, at 2 (last visited Jan. 20, 2022).

which Nokia first filed for patent protection for the innovations of the ’559 Patent, Nokia was a world leader in mobile device sales and technology. That year, Nokia launched its first media device, the Nokia 7700, and invested nearly one billion euros in research and development.[11]

45. Customers are consuming more content via streaming services, commonly referred to in the industry as OTT (Over-The-Top) services, than ever before. At the same time, competition among video services is increasing. The number of OTT providers is constantly growing, and consumer confusion is mounting. Consumers expect the same level of innovation and development for OTT video as they do for other online services, and broadcasters and content providers are under constant pressure to distinguish their offerings through personalization and availability of innovative apps that entice and retain customers. The management, curation and optimization of audience viewing experiences across screens is becoming a core customer need, and at the same time an opportunity for service differentiation.

46. In the early 2000s, the deployment of high bit-rate mobile networks such as 3G enabled the delivery of new digital services, including video calling and streaming. *See, e.g.*, Ex.18, ’559 Patent, col. 1, ll. 17-40. While audio could be delivered adequately using the bit rates available at the time, the limited transfer rates made it difficult to handle data-intensive tasks like delivering high quality full-motion video. *See, e.g.*, *id.* For this and other reasons, alternative broadband delivery techniques were being investigated to support the delivery of data-intensive content. As digital broadband data broadcast networks evolved, there was increasing interest in combining use of mobile telecommunications with a broadband delivery technique to achieve efficient delivery of digital services to users on the move. But this led to new technical challenges

[11] *See* Ex. 20, Press Release, Nokia, Nokia Closes 2003 With Excellent Fourth Quarter, (Jan. 24, 2004), at 6, 9, available at https://www.nokia.com/system/files/files/q4-2003-earnings-release-pdf.pdf.

for content providers as they had to learn new techniques to efficiently deliver content to the myriad mobile devices that could consume broadband content over mobile networks.

47. At the time, mobile terminals would typically download content by "pulling" it from a server. *See, e.g.*, *id.* at col. 2, ll. 25-39. This is because content providers tended to use content flow policies that had been used in non-mobile networks. *See id.* In those cases, the content provider typically maintained control over the content flow policy to the mobile terminal to enforce content access rights requirements. *See id.* The "pull" technique was thus rooted in the industry's established habits, which ignored input from the devices consuming the content that might otherwise affect an operator's content flow policy. Such outdated content flow policies were inefficient and undesirable as broadband content became accessible to mobile users everywhere and with myriad devices. When controlling content sent to a mobile device, they did not take into account, for example, the user preferences, terminal capabilities, previous content downloads, and/or use of previous content for that device. *See id.* at col. 2, ll. 40-53.

48. The '559 Patent addresses these problems, among others, by giving a network entity control of the flow of content to the terminal based, in part, on status information from the terminal. *See* '559 Patent, col. 2, ln. 57 – col. 3, ln. 9. Content flow is controlled, for example, by instructing the terminal to perform actions, such as downloading pieces of content from an origin server, or other content related actions based, in part, on the status information provided to the network entity from the terminal. *See id.* at col. 3, ll. 20-51. For example, the content provider can control the downloading and storage of content, as well as the deletion of content, at the terminal based upon status information regarding the terminal, and if so desired, further based upon status information regarding a source of content, such as the digital broadcast receiver, an origin server, or the like. *See id.* at col. 11, ll. 6-30. In that way, the flow of content to the terminal is more efficient since

the flow of new content to the terminal is affected by aspects of the terminal itself. *See id.* at col. 10, ll. 45-59.

C. U.S. Patent No. 7,233,790

49. U.S. Patent No. 7,233,790 (the "'790 Patent"), titled "Device Capability Based Discovery, Packaging and Provisioning of Content for Wireless Mobile Devices," issued on June 19, 2007. VL owns all rights and title to the '790 Patent, as necessary to bring this action. A true and correct copy of the '790 Patent is attached as Exhibit 21.

50. The original assignee of the '790 Patent is Openwave Systems, Inc. ("Openwave"), a leading developer of software applications for mobile devices. In the early 2000s, when the inventions of the '790 Patent were in development, Openwave's operating system and web browser software was being installed on billions of mobile phones.[12] This provided Openwave with a front seat to the many new products and services available to consumers on mobile devices.

51. The TV industry has been heavily affected by the rise of video on demand ("VOD") and over-the-top ("OTT") services, which allow users to conveniently stream over the Internet their favorite video content and watch it at any time, in any place, and in the format that best fits their needs. Today, digital video content is available from myriad streaming services and Pay TV operators and can be consumed on an ever-growing number of different connected consumer devices.

52. In the early 2000s, when digital video delivery over the Internet was in its nascent period, delivering media to large numbers of mobile users presented challenges due to the stringent

[12] *See* Ex. 22, *Openwave Announces Mobile Browser Integration for Qualcomm's Brew Solution*, INTERNET ARCHIVE WAYBACK MACHINE, (Sept. 12, 2006), *https*://web.archive.org/web/20061127222501/http://www.openwave.com/us/news_room/press_releases/2006/20060912_opwv_brew_0912.htm, at 1 (last visited Jan. 11, 2022).

requirements of streaming media, mobility, wireless, and scaling to support large numbers of users. While advances in next-generation cellular networks and wireless networks were bringing higher bandwidths to mobile users, these higher bandwidths naturally created the demand for media-rich content, which in turn created requirements for a media delivery infrastructure that could handle the challenges of streaming media, user mobility, and scaling to large numbers of users accessing content with different types of devices. Traditional content delivery techniques that had previously served the market reasonably well at the time were no longer capable of meeting current needs.

53. Indeed, these techniques were rooted in the nature of the old technologies, in which content was prepared and packaged once, for distribution over a traditional broadcast medium and in a singular, conventional broadcast format. From a content supplier's perspective, an impediment to the efficient distribution of digital content was the fact that different connected devices often required different content packaging formats and provisioning protocols. In order for the content supplier to make a given item of digital content available to multiple connected devices supporting different provisioning models, a digital content supplier would normally have to deploy that item of content multiple times, packaging it differently for each of the provisioning models. Needing to package and provision digital content in a manner that is suitable for all of the connected devices in the marketplace is very burdensome. Moreover, it was a challenge for content suppliers to keep up with the constant changes in device capabilities for the many connected devices in the marketplace. As such, there was often a gap between the interoperability of a given digital product and how effective it could be used on a certain device.

54. The '790 Patent addresses this problem, among others, providing an efficient way for content providers to distribute digital content to different connected devices without the need to separately package and deploy the content differently for each device. *See* '790 Patent, col. 1,

ln. 60 – col. 2, ln. 17; col. 12, ll. 40-45. Since multiple different implementations of the content are kept in the content library, each associated with certain device capabilities, content can be packaged once, and distributed to the different user devices, leaving to the device the choice of which implementation version of the content to obtain from the service. *See* '790 Patent, col. 2, ll. 50-67; col. 9, ll. 37-50; col. 12, ll. 45-58; col. 11, ll. 42-62; col. 12, ll. 54-58. In the patented invention, the product information is separated from the content itself, the content is separated from how it is packaged for delivery, the packaged content is separated from the delivery mechanism, and the delivery mechanism is separated from the discovery mechanism. *See* '790 Patent, col. 12, ll. 46-54. By virtue of this separation, a content supplier can deploy content only once, targeting a wide range of devices, and trust it will successfully be delivered to those devices over a wide range of provisioning protocols. *See* '790 Patent, col. 12, ll. 54-58. Furthermore, access to different implementations of the content can be managed by the content supplier based on a device's capabilities ensuring the efficient distribution of compatible material. *See* '790 Patent, col. 2, ll. 7-17; col. 7, ll. 39-45; col. 8, ln. 66 – col. 9, ln. 12; col. 9, ll. 37-62.

ACCUSED NETFLIX PRODUCTS AND SERVICES

A. Netflix '878 Accused Products

55. The "Netflix '878 Accused Products" refers to all Netflix products, services, and functionalities that implement, in whole or in part, H.264 entropy encoding using CAVLC. This includes, for example, Netflix's use of the H.264 baseline, main, and high profiles with CAVLC to encode its video content.

56. H.264 is the name for technology described in an industry standard that is widely used to encode and decode streaming video. H.264 reduces the file size of video files without any loss in quality of video, enabling companies to stream video in higher quality given the same network bandwidth.

57. H.264 focuses on the coding of the picture portions of the video content. To this end, H.264 defines a format, or syntax, for compressed video and a method for decoding this syntax to produce a displayable video sequence. An H.264 video encoder carries out prediction, transform, and encoding processes to produce a compressed H.264 bitstream. An H.264 video decoder carries out the complementary processes of decoding, inverse transform, and reconstruction to produce a decoded video sequence.

58. H.264 has been the dominant industry standard for compressing video for applications such as digital television, DVD video, video conferencing, and Internet video streaming. Standardizing video compression made it possible for products from different manufacturers to inter-operate. Recommendation H.264: Advanced Video Coding is a video decoding standard published by the international standards bodies ITU-T (International Telecommunication Union) and ISO/IEC (International Organisation for Standardisation / International Electrotechnical Commission) (attached as Ex. 23). It defines a format (syntax) for compressed video and a method for decoding this syntax to produce a displayable video sequence. Products that support H.264 encoding and decoding are compliant with the H.264 standard.

59. According to Netflix, "given its wide support, our H.264/AVC Main profile family still represents a substantial portion of the members viewing hours and an even larger portion of the traffic."[13] As recently as August 10, 2020, Netflix announced that it was continuing to invest in its use of H.264, stating that "[k]eeping in mind our goal to maintain ubiquitous device support, we leveraged what we learned from innovations implemented during the development of newer encode families and have made a number of improvements to our H.264/AVC Main profile per-

[13] *See* Ex. 24, https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9, at 1-2.

title encodes."[14] For its mobile devices, Netflix uses the high profile of H.264, which "enjoys broad decoder support."[15] Netflix's use of H.264, and the resulting decrease in the bitrate of its content streams, "corresponds to a significant reduction in the overall Netflix traffic as well. These changes also lead to an improvement in Quality-of-Experience (QoE) metrics that affect the end user experience, such as play delays (i.e. how long it takes for the video to start playing), rebuffer rates, etc., as a result of the reduction in average bitrates. In addition, footprint savings will allow more content to be stored in edge caches, thus contributing to an improved experience for our members."[16]

60. On information and belief, Netflix performs H.264 encoding using CAVLC to efficiently and seamlessly deliver video to its customers.

B. Netflix '559 Accused Products

61. "Netflix '559 Accused Products" refers to all Netflix products, services, features, and functionalities that control the flow of content to a client based on status information from a client and/or a content server. This includes, for example, all versions and implementations of the Netflix streaming service (including the service and the application).

62. Netflix controls the flow of streaming content to Netflix clients based, in part, on status information from the client and/or a content server.

[14] *See* Ex. 24, https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9, at 2; *see also* Ex. 25, https://www.broadbandtvnews.com/2020/08/11/netflix-works-to-improve-h-264-experience/, at 1-2.

[15] *See* Ex. 26, https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909, at 1-2; Ex. 27, https://www.vdocipher.com/blog/tech-update-netflix-updates-codecs-use-efficient-encoding/, at 1-2.

[16] *See* Ex. 24, https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9.

63. On information and belief, Netflix uses status information from the client and/or a content server to "bookmark" content so that, when a viewer stops viewing content and then returns to view the same content later, the content begins at the time segment where the viewer left off. Without this feature, returning to view content would be significantly more inconvenient and probably cause viewers to view less content.[17]

64. On information and belief, Netflix uses status information from the client and/or a content server to automatically download content based on the subscriber's viewing history. Netflix's automatic download feature operates so that, when a viewer "finish[es] watching a downloaded episode, Smart Downloads will delete it, and then automatically download the next episode."[18] The automatic download feature allows Netflix to "smartly download new episodes of TV shows so that users don't have to worry about doing the same manually."[19]

65. On information and belief, Netflix further uses status information from the client and/or a content server to prompt video features that significantly improve the consumer viewing experience. For example, Netflix provides viewers a "Skip Intro" option shortly after the opening

[17] *See, e.g.*, Ex. 28, https://about.netflix.com/en/news/now-you-can-easily-tidy-up-your-continue-watching-row-on-all-devices, at 1; *See* Ex. 29, https://help.netflix.com/en/node/115312; https://www.addictivetips.com/web/get-continue-watching-on-top-in-netflix/, at 1-3; Ex. 30, https://discussions.apple.com/thread/4365964, at 1-4 (various user complaints about a "bug" where Netflix videos were not updating to the most recent episode).

[18] *See* Ex. 31, *Downloading on Netflix Just Got Smarter*, NETFLIX, https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1, at 1-2 (last visited Feb. 17, 2022); Ex. 32, Amazon Prime Video to Get Netflix-like Smart Downloads Feature, MOBILESCOUT, at 1-2 (Nov. 19, 2019), https://www.mobilescout.com/android/news/n114549/amazonprime-video-auto-downloads-feature-spotted.html (last visited Feb 17, 2022).

[19] *See* Ex. 31, *Downloading on Netflix Just Got Smarter*, NETFLIX, https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1, at 1-2(last visited Feb. 17, 2022).

credits sequence begins.[20] Similarly, when the beginning of an episode includes a summary of previous episodes in a TV series, Netflix provides viewers a "Skip Recap" option.[21] Additionally, immediately after the final scene of an episode, Netflix's "autoplay" feature allows the viewer to automatically play the next episode in a TV series.[22] Without these features, viewing content would be more burdensome and the viewing experience would suffer.[23]

C. Netflix '790 Accused Products

66. "Netflix '790 Accused Products" refers to all Netflix products, services, features, and functionalities that maintain a product catalog of, and provide access to, content with a plurality of different implementations corresponding to device capabilities. This includes, for example, all versions and implementations of the Netflix streaming service (including the service and the application).

67. Netflix maintains a catalog of, and provides access to, streaming content, which has a plurality of different implementations corresponding to device capabilities.

68. On information and belief, Netflix receives and stores content, such as movies and

[20] *See* Ex. 64, *How to Skip Intros on TV Shows*, NETFLIX, https://help.netflix.com/en/node/63402, at 1 (last visited Feb. 17, 2022).

[21] *See* Ex. 33, *Netflix Just Sneakily Added Another Button That Will Make You Binge So Much Better*, HELLOGIGGLES (Apr. 04, 2017) https://hellogiggles.com/reviews-coverage/tv-shows/netflix-just-rolled-new-feature-will-seriously-help-show-facts-straight-binge/; https://help.netflix.com/en/node/2102, at 1-6.

[22] *See* Ex. 35, *How to Autoplay the Next Episode*, NETFLIX, https://help.netflix.com/en/node/121518, at 1.

[23] *See, e.g.*, Ex. 36, *Amazon Prime Video Advert Skip*, DIGITALSPY (July 12, 2021), https://forums.digitalspy.com/discussion/2420457/amazon-prime-video-advert-skip (last visited Jan. 20, 2022) at 6 ("The skip credits and Recap option is very useful as one click is a lot easier than having to forward to the place you want. It is obviously a popular feature as Netflix, Amazon and Disney Plus all have the feature . . . ."); Ex. 37, *Netflix's "Skip Intro" Feature*, MEDIUM, https://medium.com/an-attempt-at-writing/netflixs-skip-intro-feature-how-the-hell-do-they-do-that-7c5db9408f82 at 1-2 ("[Y]ou can skip intros of most shows. . . . [I]t's a huge time saver and makes situations like binge watching especially very fluid.").

TV shows, that will be provided to users.[24] By doing so, delivery of content to users is faster and less expensive.[25]

69. On information and belief, Netflix maintains a plurality of different implementations of this content, where each implementation corresponds to device capabilities of the Netflix client.[26] Maintaining different implementations of content allows the Netflix clients to smoothly and seamlessly stream content without "blips."[27]

70. On information and belief, Netflix maintains a product catalog which contains references to the content.[28] By maintaining a product catalog that includes references to various

[24] *See, e.g.,* Ex. 38, *How Netflix Works with ISPs Around the Globe to Deliver a Great Viewing Experience*, NETFLIX (Mar. 17, 2016), https://about.netflix.com/en/news/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience, at 2-5 (last visited Feb. 21, 2022) ("Netflix Open Connect delivers 100% of our video traffic . . . almost all Netflix content is served from the local OCAs . . .."); *see also* Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) , https://www.theverge.com/22787426/netflix-cdn-open-connect (last visited Feb. 21, 2022), at 3 ("To avoid the traffic and fees, Netflix ships copies of its content to its own servers ahead of time.").

[25] *See, e.g.,* Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) at 3 ("To avoid the traffic and fees, Netflix ships copies of its content to its own servers ahead of time.").

[26] *See, e.g.,* Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021), at 6 ("Netflix effectively ships three copies of each of its titles to its servers, each with a different level of quality."); *see also* Ex. 40, *Cloud Storage: How Does Netflix Store Their Content?*, GLOBAL TECH COUNCIL (Jan. 1, 2021) , https://www.globaltechcouncil.org/big-data/cloud-storage-how-does-netflix-store-their-content/ (last visited Feb. 21, 2022), at 2-4 ("To be available for viewing on a range of devices: from 44' 4K Smart TV to 6-inch handset displays, they need to be encoded and compressed into different formats and qualities. . . . Many versions of the compressed encoded file are copied worldwide to hundreds of servers, forming a larger CDN for Netflix-like people around the world.")

[27] *See, e.g.,* Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) ("'We will adapt the content to the quality of the network and not vice versa . . . [t]hat's the reason why you don't see when your network has a blip – your streaming stays constant.'").

[28] *See, e.g.,* Ex. 38, *How Netflix Works with ISPs Around the Globe to Deliver a Great Viewing Experience*, NETFLIX (Mar. 17, 2016) ("Although the number and size of the files that make up our content library can be staggering, we are able to use sophisticated popularity models

content, Netflix is able to minimize and eliminate the risks of service disruption,[29] and personalize its presentations for a particular user.[30]

ALLEGATIONS OF PATENT INFRINGEMENT

COUNT I
INFRINGEMENT OF U.S. PATENT NO. 8,139,878

71. VideoLabs incorporates by reference the foregoing paragraphs of this Complaint as if fully set forth herein.

72. VL is the assignee and lawful owner of all right, title, and interest in and to the ’878 Patent. The ’878 Patent is valid and enforceable.

73. On information and belief, Netflix has infringed and continues to infringe the ’878 Patent in violation of 35 U.S.C. § 271(a), either literally or through the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States products and/or methods that practice at least claim 1 of the ’878 Patent, including with respect to the Netflix ’878

to make sure the right file is on the right server at the right time.”); *see also* Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) (“[W]e are placing content on all of these servers around the world . . ..”); Ex. 40, *Cloud Storage: How Does Netflix Store Their Content?*, GLOBAL TECH COUNCIL (Jan. 1, 2021) (“It is essential to sift through and organize the petabyte value of content that Netflix has, like a library.”); Ex. 41, *Why Your Netflix Thumbnails Don’t Look Like Mine*, VOX (Nov. 21, 2018), https://www.vox.com/2018/11/21/18106394/why-your-netflix-thumbnail-coverart-changes (last visited Feb. 21, 2022) (“Netflix has a catalog of thousands of videos to watch . . . Netflix doesn’t just use a film or show’s original art; it employs an algorithm with the daunting task of sourcing high-quality images from those videos.”).

[29] *See, e.g.,* Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) (“‘This pre-placement of our films and shows allows us, based on prime time viewing hours, to store 100 percent of our catalog locally. And that basically eliminates the whole risk associated with service disruption.”).

[30] *See, e.g.*, Ex. 42, *Selecting the Best Artwork for Videos Through A/B Testing*, THE NETFLIX TECH BLOG (May 3, 2016), https://netflixtechblog.com/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6 (last visited Feb. 21, 2022) (“Selecting the best artwork has improved the Netflix product experience in material ways. We were able to help our members find and enjoy titles faster.”); .”); *see also e.g.*, Ex. 63, “Learning a Personalized Homepage”, NETFLIX, https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a (last retrieved, Feb. 21, 2022)..

Accused Products.

74. On information and belief, Netflix uses the Netflix ’878 Accused Products for its own business purposes. In addition, Netflix regularly conducts research, testing, and troubleshooting of the Netflix ’878 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix’s subsidiaries) use the Netflix ’878 Accused Products.

75. On information and belief, Netflix’s infringement through its use of H.264 CAVLC entropy encoding, described below, is exemplary of all of Netflix’s infringement with respect to all the Netflix ’878 Accused Products.

76. The Netflix ’878 Accused Products directly infringe at least claim 1 of the ’878 Patent, for example, by performing variable length encoding of blocks of picture data using the block data, inter-, intra-, and context-aware prediction to generate a predictive block, calculating a residual block using orthogonal transformation and quantization, and using the number of non-zero coefficients in the predicted block to encode the picture data at the encoder.

77. Each of the Netflix ’878 Accused Products meet every limitation of claim 1 of the ’878 Patent, which recites:

> 1. A transmitting apparatus which transmits multiplexed data which is obtained by multiplexing coded audio data and coded picture data, said transmitting apparatus comprising:
>
> an audio processing unit configured to code audio data to obtain coded audio data;
>
> a picture coding unit configured to code picture data to obtain coded picture data; and
>
> a multiplexing unit configured to multiplex the coded audio data and the coded picture data to obtain multiplexed data,
>
> wherein said picture coding unit includes a block decoding unit configured to code a block image to obtain coded block data, the block image being obtained by dividing a picture signal into plural

> blocks, generating a residual block image from a block image of the respective blocks and a predictive block image obtained by intra-picture prediction or inter-picture prediction, and coding, on a block basis, coefficients obtained by performing orthogonal transformation and quantization on the residual block image,
>
> wherein said block coding unit includes:
>
> a coefficient number coding unit configured to code a total number of non-zero coefficients included in a current block to be coded, each of the non-zero coefficients being a coefficient having a value other than "0";
>
> wherein said coefficient number coding unit includes:
>
> a determining unit configured to determine a predictive value for the number of non-zero coefficients included in the current block based on the number of non-zero coefficients included in a coded block located on a periphery of the current block;
>
> a selecting unit configured to select a variable length code table based on the determined predictive value; and
>
> a variable length coding unit configured to perform variable length coding on the total number of the non-zero coefficients included in the current block, by using the selected variable length code table.

78. Each of the Netflix '878 Accused Products includes a transmitting apparatus that transmits multiplexed data, which is obtained by multiplexing coded audio data and coded picture data. In one representative example, H.264 carries audio and video multiplexed in a single stream. H.264 is directed to the picture portion of video, so devices containing H.264 encoders and decoders must multiplex/demultiplex audio and picture data in order to provide the H.264 picture data. For example, encoders can multiplex the audio and pictures into a single stream, so that decoders receive the complete video presentation, including sound. Alternatively, the coded audio and coded video can be time-multiplexed and delivered as independent content streams. Regardless, decoders receive the complete video presentation, including sound, and decode the stream to recreate the video. In the ISO Media File Format, which each Netflix '878 Accused Products, and the Netflix streaming service, is capable of processing, for example, a coded stream such as an H.264 video sequence or an audio stream is stored as a track, representing a sequence

of coded data items or samples. Figure 8.32, below, illustrates an example of such multiplexed data, in which coded audio data ("audio track samples") and coded picture data ("video track samples") are multiplexed together.

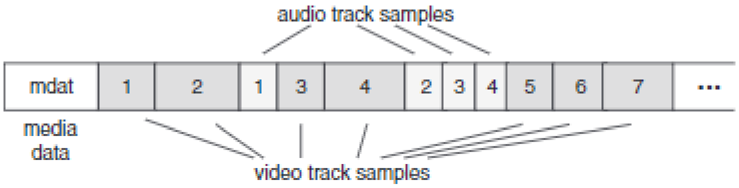


Figure 8.32 ISO Media File

Ex. 17, Richardson, at 247.

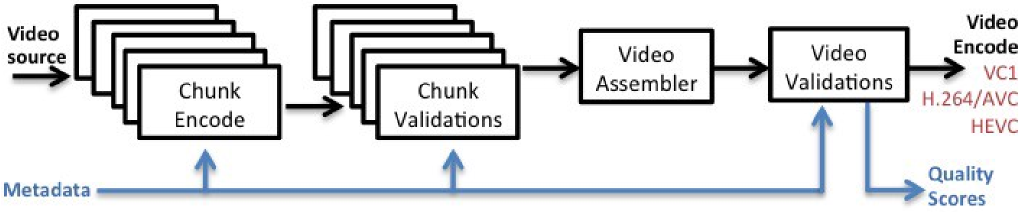
79. The Netflix '878 Accused Products have an audio processing unit configured to code audio data to obtain coded audio data. For example, the Netflix '878 Accused Products incorporate an audio codec that is configured to code audio data according to one of several input formats, including without limitation, Dolby Digital, Dolby Digital Plus, Dolby Atmos, and AAC.[31] The audio data is encoded as coded audio data ("audio track samples").

80. The Netflix '878 Accused Products further include a picture coding unit configured to code picture data to obtain coded picture data. For example, the Netflix '878 Accused Products incorporate a H.264 video codec that is configured to code picture data to generate a H.264-compliant bitstream.[32] The picture data is encoded as coded picture data ("video track samples"). For example, Netflix employs a parallel video encoding process to generate multiple quality

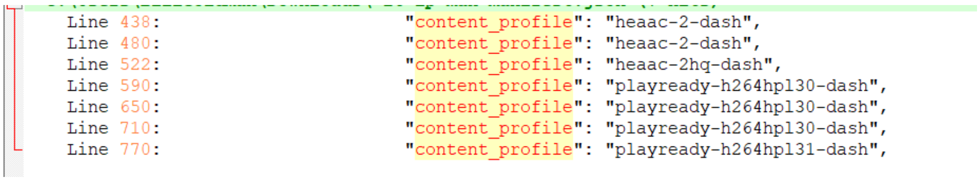
[31] *See* Ex. 43, https://netflixtechblog.com/optimizing-the-aural-experience-on-android-devices-with-xhe-aac-c27714292a33; Ex. 44, https://about.netflix.com/en/news/dolby-atmos-coming-to-netflix;

[32] *See* Ex. 24, https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9; Ex. 26, https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909.

representations of a content program at different bitrates.[33]



A typical Netflix content program is encoded using multiple H.264 encoding profiles for encoding at the different bitrates:[34]



```
Line 438:                         "content_profile": "heaac-2-dash",
Line 480:                         "content_profile": "heaac-2-dash",
Line 522:                         "content_profile": "heaac-2hq-dash",
Line 590:                         "content_profile": "playready-h264hpl30-dash",
Line 650:                         "content_profile": "playready-h264hpl30-dash",
Line 710:                         "content_profile": "playready-h264hpl30-dash",
Line 770:                         "content_profile": "playready-h264hpl31-dash",
```

81. The Netflix '878 Accused Products include a multiplexing unit configured to multiplex the coded audio data and the coded picture data into multiplexed data. Encoders multiplex the audio track samples and video track samples so that decoders receive the complete video presentation, including sound. One non-limiting example of multiplexed data generated by the Netflix '878 Accused Products is shown in Figure 8.32 below, which shows the multiplexed stream of an ISO Media File including both coded audio track data and coded video track data.

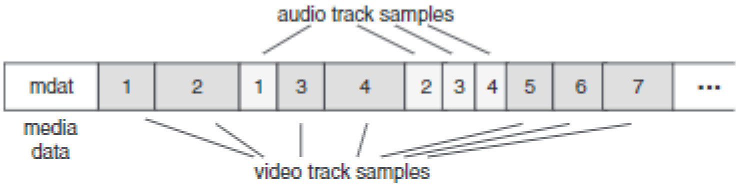


Figure 8.32 ISO Media File

[33] *See* Ex. 6, https://netflixtechblog.com/high-quality-video-encoding-at-scale-d159db052746.

[34] *See e.g.*, filtered JSON manifest for the representative Netflix movie "IP Man" containing four H.264 High profile encodes.

Id. at 247. As noted, the coded audio data and the coded picture data can alternatively be time-multiplexed.

82. The picture coding unit in the Netflix ’878 Accused Products includes a block coding unit configured to code a block image to obtain coded block data, the block image being obtained by dividing a picture signal into plural blocks.

83. In the Netflix ’878 Accused Products picture data is coded by an H.264-compliant encoder by dividing a picture signal into macroblocks, as shown in Figure 6-7:

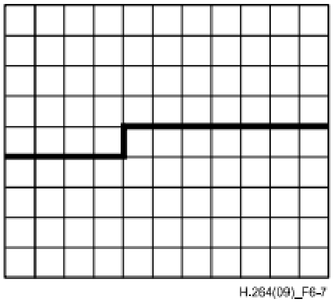


Figure 6-7 – A picture with 11 by 9 macroblocks that is partitioned into two slices

See ITU-T H.264, *Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of moving video, Advanced video coding for generic audiovisual services*, Section 6.3, p. 25 (03/2009), at Figure 6-7.

84. The macroblock consists of a 16 x 16 block of luma samples and two corresponding blocks of chroma samples. A macroblock can be further partitioned for inter-prediction forming segmentations for motion representation as small as a block of 4 x 4 luma samples.[35]

85. In the Netflix ’878 Accused Products a residual block image is generated from the block image of the respective blocks and a predictive block image is obtained by intra-picture

[35] *See generally* Ex. 23, ITU-T H.264*, Series H: Audio Visual and Multimedia Systems, Infrastructure of Audiovisual Services – Coding of Moving Video, Advanced Video Coding for Generic Audio Visual Services* [hereinafter “H.264 Standard”], Section 0.6.3, p. 5 (09/2019).

prediction or inter-picture prediction by an H.264-complaint encoder. For example, Figure 6.6 below shows a picture signal to be coded, with the macroblock being coded highlighted. The macroblock is predicted using neighboring, previously-encoded samples, as shown in Figure 6.7; because this prediction looks only to the other macroblocks of the same picture, this is called intra-picture prediction. The predicted macroblock is shown in Figure 6.7. Figure 6.8 shows the prediction (Figure 6.7) subtracted from the original (Figure 6.6), which is called the residual.

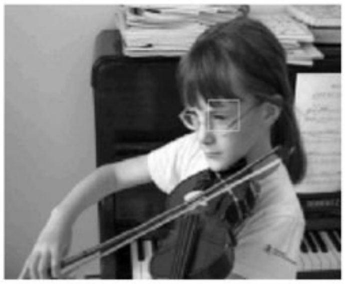


Figure 6.6 QCIF frame with highlighted macroblock

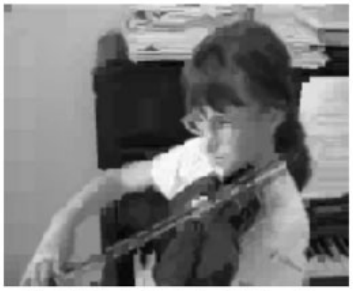


Figure 6.7 Predicted luma frame formed using H.264 intra prediction

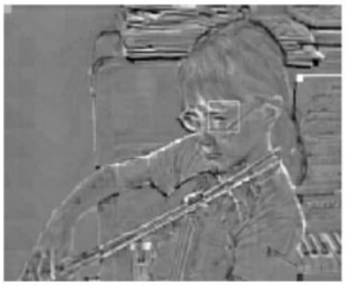


Figure 6.8 Residual after subtracting intra prediction

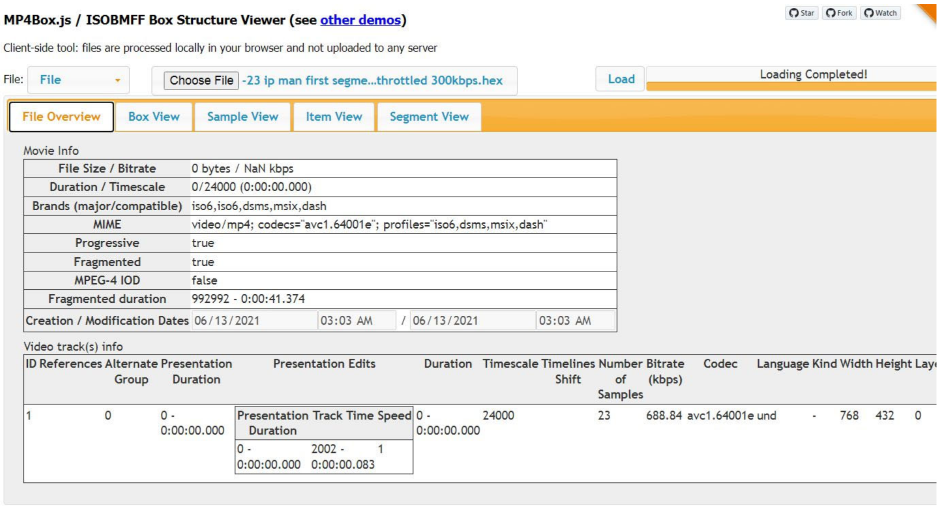
Richardson at 141-143.

86. The Netflix ’878 Accused Products can use intra- and inter- picture coding on the macroblocks of a picture signal. Inter-picture coding predicts the value of the macroblock using temporal statistical dependencies between different pictures. Both types of prediction can be used by the H.264-compliant encoder to calculate the residual.

87. In the Netflix ’878 Accused Products the H.264-compliant encoder codes, on a block basis, coefficients obtained by performing orthogonal transformation and quantization on the residual block image. H.264 specifies an *entropy_coding_mode* flag that dictates the entropy encoding algorithm used to encode the picture data. When this flag is set to “0” the residual block data is coded using a CAVLC scheme.[36] In the case of Netflix’s AVC-Main and AVC-High profile

[36] *See* Ex. 23, H.264 Standard, Section 7.4.2.2, pp. 81-82 (09/2019).

H.264 encodes, the *entropy_coding_mode* flag is set to "0":[37]



MP4Box.js / ISOBMFF Box Structure Viewer (see other demos)

Client-side tool: files are processed locally in your browser and not uploaded to any server

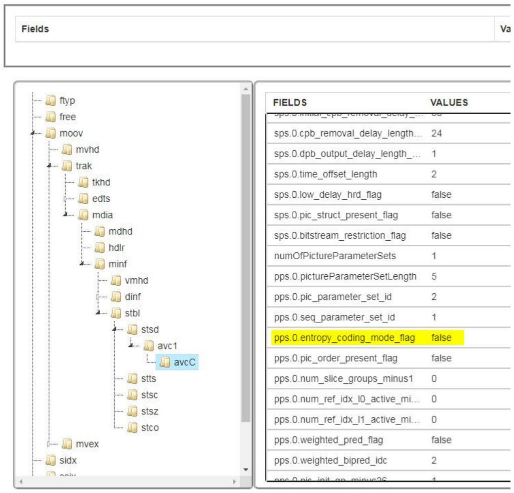
File: File | Choose File -23 ip man first segme...throttled 300kbps.hex | Load | Loading Completed!

Movie Info

File Size / Bitrate	0 bytes / NaN kbps
Duration / Timescale	0/24000 (0:00:00.000)
Brands (major/compatible)	iso6,iso6,dsms,msix,dash
MIME	video/mp4; codecs="avc1.64001e"; profiles="iso6,dsms,msix,dash"
Progressive	true
Fragmented	true
MPEG-4 IOD	false
Fragmented duration	992992 - 0:00:41.374
Creation / Modification Dates	06/13/2021 03:03 AM / 06/13/2021 03:03 AM

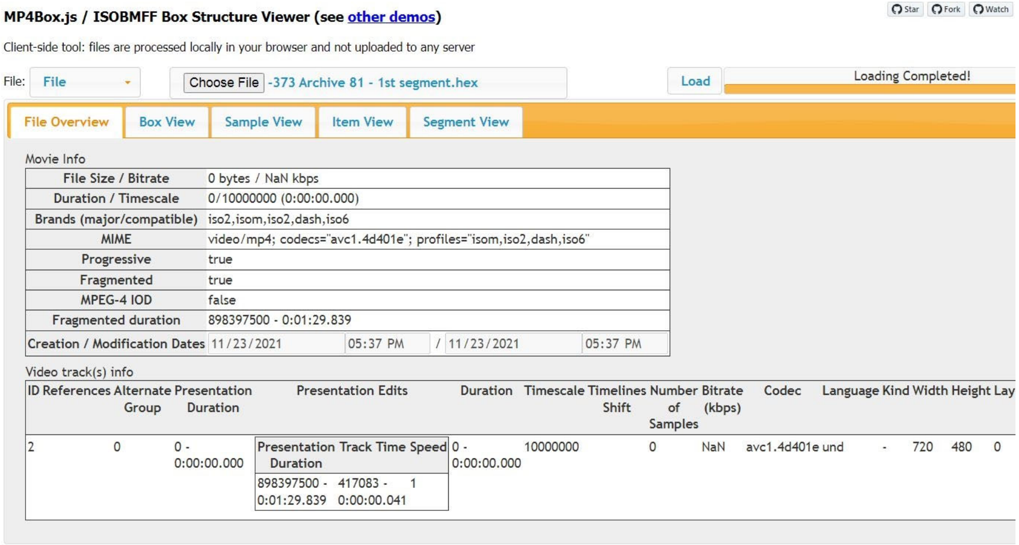
Video track(s) info

ID	References	Alternate Group	Presentation Duration	Presentation Edits	Duration	Timescale	Timelines Shift	Number of Samples	Bitrate (kbps)	Codec	Language	Kind	Width	Height	Lay
1		0	0 - 0:00:00.000	Presentation Duration: 0 - 0:00:00.000 Track Time: 2002 - 0:00:00.083 Speed: 1	0 - 0:00:00.000	24000		23	688.84	avc1.64001e	und	-	768	432	0



FIELDS	VALUES
sps.0.cpb_removal_delay_length...	24
sps.0.dpb_output_delay_length_...	1
sps.0.time_offset_length	2
sps.0.low_delay_hrd_flag	false
sps.0.pic_struct_present_flag	false
sps.0.bitstream_restriction_flag	false
numOfPictureParameterSets	1
pps.0.pictureParameterSetLength	5
pps.0.pic_parameter_set_id	2
pps.0.seq_parameter_set_id	1
pps.0.entropy_coding_mode_flag	false
pps.0.pic_order_present_flag	false
pps.0.num_slice_groups_minus1	0
pps.0.num_ref_idx_l0_active_mi...	0
pps.0.num_ref_idx_l1_active_mi...	0
pps.0.weighted_pred_flag	false
pps.0.weighted_bipred_idc	2
pps.0.pic_init_qp_minus26	1

[37] Picture Parameter Set analysis of two different Netflix content programs encoded using AVC High profile (first two images) and AVC Main profile (second two images).



MP4Box.js / ISOBMFF Box Structure Viewer (see other demos)

Client-side tool: files are processed locally in your browser and not uploaded to any server

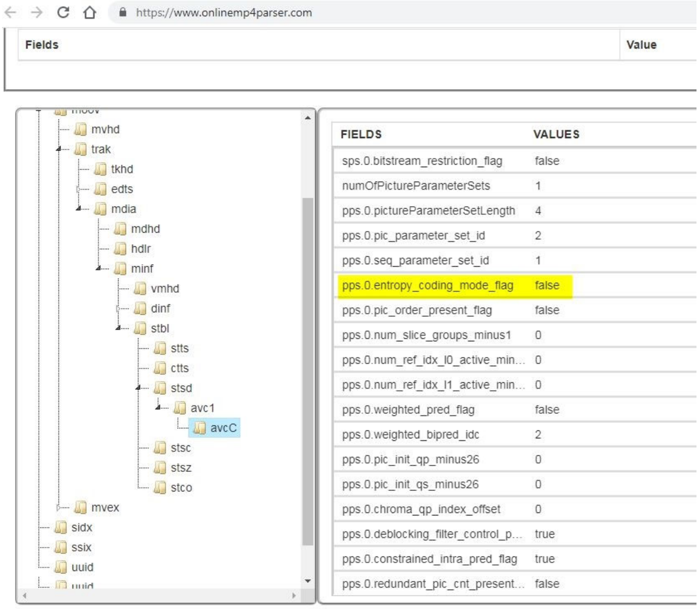
File: File · Choose File -373 Archive 81 - 1st segment.hex · Load · Loading Completed!

Movie Info

File Size / Bitrate	0 bytes / NaN kbps
Duration / Timescale	0/10000000 (0:00:00.000)
Brands (major/compatible)	iso2,isom,iso2,dash,iso6
MIME	video/mp4; codecs="avc1.4d401e"; profiles="isom,iso2,dash,iso6"
Progressive	true
Fragmented	true
MPEG-4 IOD	false
Fragmented duration	898397500 - 0:01:29.839
Creation / Modification Dates	11/23/2021 05:37 PM / 11/23/2021 05:37 PM

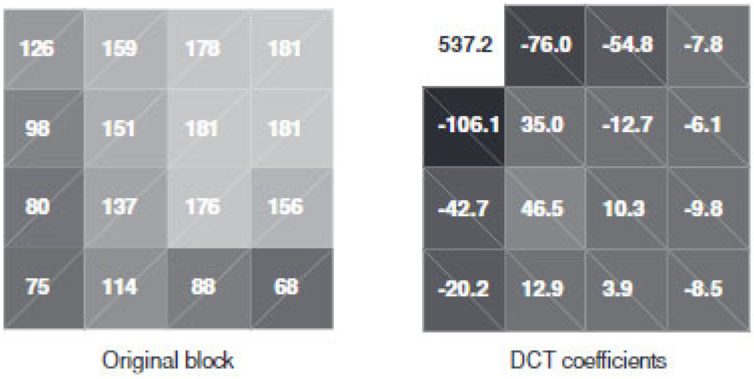
Video track(s) info

ID	References	Alternate Group	Presentation Duration	Presentation Edits	Duration	Timescale	Timelines Shift	Number of Samples	Bitrate (kbps)	Codec	Language	Kind	Width	Height	Lay
2		0	0 - 0:00:00.000	Presentation Duration: 898397500 - 0:01:29.839 Track Time: 417083 - 0:00:00.041 Speed: 1	0 - 0:00:00.000	10000000		0	NaN	avc1.4d401e	und	-	720	480	0



FIELDS	VALUES
sps.0.bitstream_restriction_flag	false
numOfPictureParameterSets	1
pps.0.pictureParameterSetLength	4
pps.0.pic_parameter_set_id	2
pps.0.seq_parameter_set_id	1
pps.0.entropy_coding_mode_flag	false
pps.0.pic_order_present_flag	false
pps.0.num_slice_groups_minus1	0
pps.0.num_ref_idx_l0_active_min...	0
pps.0.num_ref_idx_l1_active_min...	0
pps.0.weighted_pred_flag	false
pps.0.weighted_bipred_idc	2
pps.0.pic_init_qp_minus26	0
pps.0.pic_init_qs_minus26	0
pps.0.chroma_qp_index_offset	0
pps.0.deblocking_filter_control_p...	true
pps.0.constrained_intra_pred_flag	true
pps.0.redundant_pic_cnt_present...	false

88. In such case, the resulting prediction residual is split into 4x4 blocks, and transformation and quantization are applied. An integer transform is applied to the residual, outputting a set of coefficient weighting values. This process is shown in the figures below:



126	159	178	181
98	151	181	181
80	137	176	156
75	114	88	68

Original block

537.2	-76.0	-54.8	-7.8
-106.1	35.0	-12.7	-6.1
-42.7	46.5	10.3	-9.8
-20.2	12.9	3.9	-8.5

DCT coefficients

Figure 3.31 Close-up of 4 × 4 block; DCT coefficients

Ex. 17, Richardson, at 47.

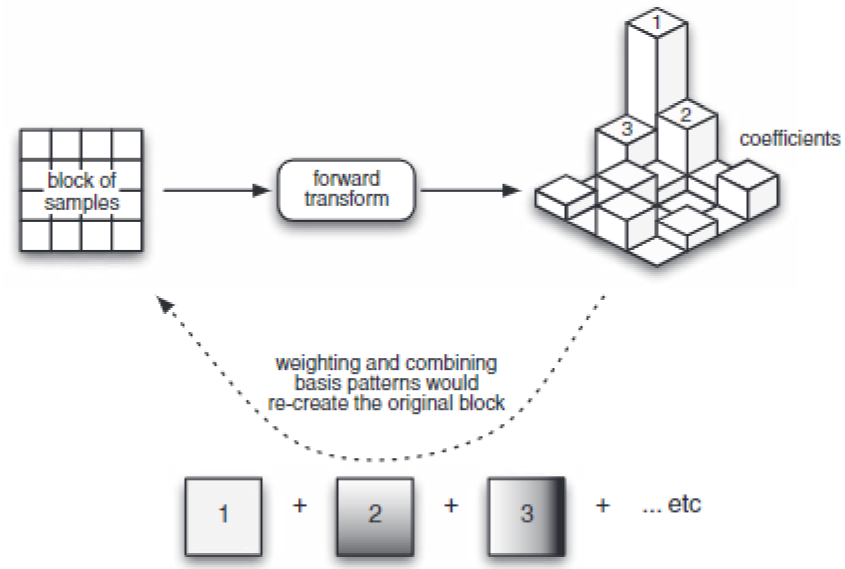

block of samples
forward transform
1
2
3
coefficients
weighting and combining basis patterns would re-create the original block
1
+
2
+
3
+
... etc

Figure 4.11 Forward transform

Ex. 17, Richardson, at 88.

89. The coefficients are then quantized, meaning that insignificant coefficient values are rounded down (for example, to zero), while a small number of significant, non-zero coefficients are retained. The quantization step is shown in Figure 4.12:

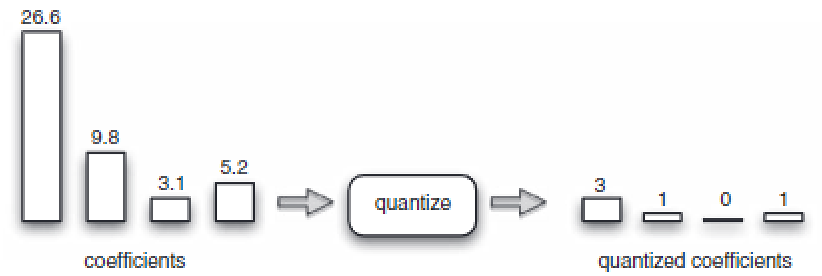


Figure 4.12 Quantization example

Ex. 17, Richardson, at 88.

90. The encoding process is shown in Figure 4.4:

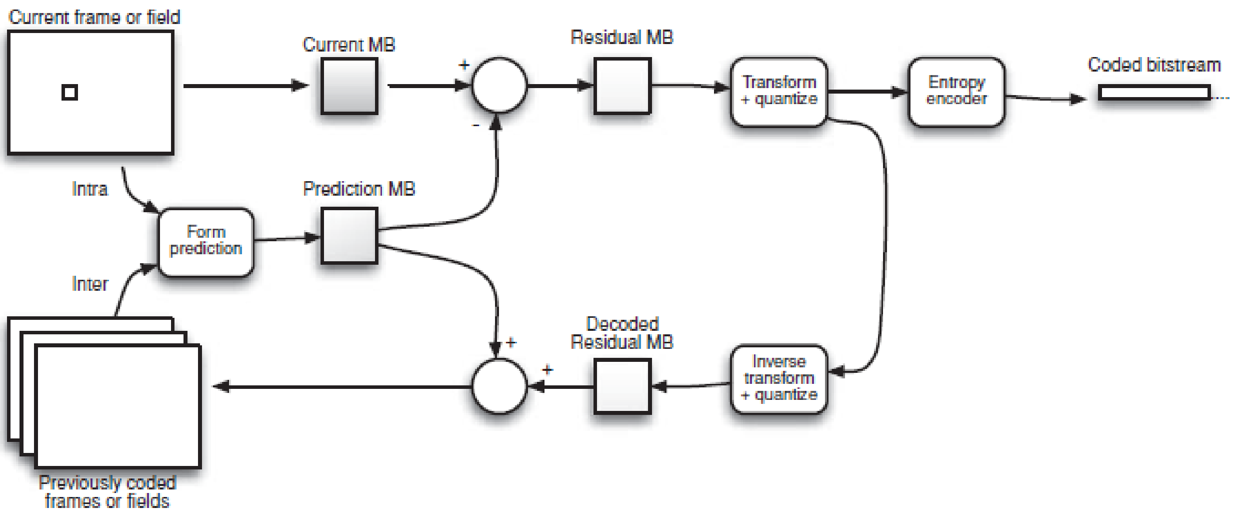


Figure 4.4 Typical H.264 encoder

Ex. 17, Richardson, at 84.

91. In the Netflix '878 Accused Products, the block coding unit includes a coefficient number coding unit configured to code a total number of non-zero coefficients included in a current block to be coded, where each of the non-zero coefficients have a value other than "0". In H.264 CAVLC encoding generally, which the Netflix '878 Accused Products support and perform, the total number of non-zero coefficients included in a current block to be coded is derived in order to generate a H.264-compliant bitstream.

92. To this end, the coefficient number coding unit in the Netflix '878 Accused

Products includes a determining unit configured to determine a predictive value for the number of non-zero coefficients included in the current block based on the number of non-zero coefficients included in a block located on a periphery of the current block. Since CAVLC is a context-adaptive variable length coding technique, the number of non-zero coefficients in neighboring blocks is correlated as part of the entropy coding process. An H.264-compliant encoder uses previously-processed macroblocks to help encode the currently-processed macroblock. The number of non-zero coefficients in previously-processed blocks on the periphery—including the blocks to the left and above the current macroblock—are used to predict the number of non-zero coefficients in the current block. This use of the blocks of the periphery is shown in Figure 6-14:

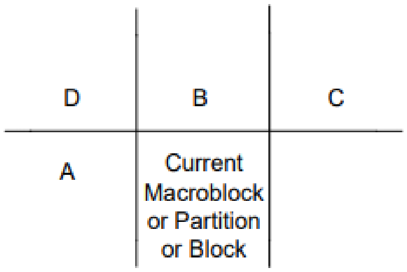


Figure 6-14 – Determination of the neighbouring macroblock, blocks, and partitions (informative)

Ex. 23, H.264 Standard, at 33.

93. The H.264-compliant encoder in the Netflix ’878 Accused Products obtains the number of non-zero coefficients in the left and above blocks to set the variable *nC*, the prediction of the current macroblock’s number of non-zero coefficients based on the neighboring macroblocks’ number of non-zero coefficients.[38]

94. The coefficient number coding unit in the Netflix ’878 Accused Products also includes a selecting unit configured to select a variable length code table based on the determined predictive value. The H.264-compliant encoder in the Netflix ’878 Accused Products uses the

[38] *See* Ex. 23, H.264 Standard, Section 9.2.1, pp. 214-216 (09/2019).

predictive value *nC* to select one of six variable length coding tables specified in Table 9-5 of the H.264 standard in order to generate a H.264-compliant bitstream.[39]

95. This selection of a variable length code table based on a determined predictive value is exemplified in Figure 7.19:

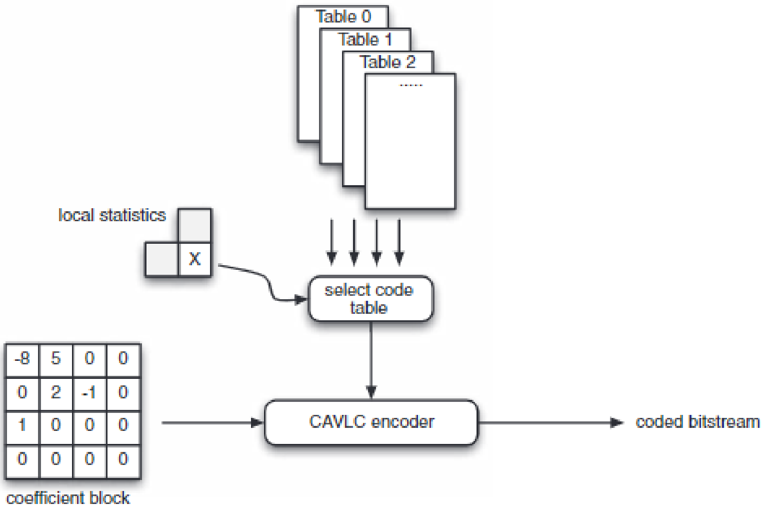


Figure 7.19 CAVLC encoder overview

Richardson at 211.

96. The coefficient number coding unit in the Netflix '878 Accused Products further includes a variable length coding unit configured to perform variable length coding on the number of the non-zero coefficients included in the current block, by using the selected variable length code table. Figure 7.19, *supra*, shows a CAVLC coding unit. The unit uses the selected variable length code table to perform variable length coding on the syntax element *coeff_token* representing the number of non-zero coefficients of the current macroblock to generate a H.264-compliant bitstream.[40]

[39] *See* Ex. 23, H.264 Standard, Section 9.2.1 and Table 9-5, pp. 214-218 (09/2019).
[40] *See generally* H.264 Standard, Section 9.2.1 and Table 9-5, pp. 214-218 (09/2019).

97. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.[41] Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.[42] Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory call on March 19, 2021.[43] Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.[44] After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a "standstill" for an indefinite period of time.[45] Given that nearly two years had already passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs' communications. On February 22, 2022, VideoLabs sent an email to Netflix apprising it of the '878 Patent, identifying the Netflix products that infringe the '878 Patent, and requesting that the parties discuss the terms for Netflix to take a license to the '878 Patent.[46]

98. Netflix of course knows how its products operate, and on information and belief, upon receiving notice of the '878 Patent, has begun investigating the '878 Patent and its infringement of the Netflix '878 Accused Products. Netflix has been given further notice of the '878 Patent and its infringement of the '878 Patent through the filing of this Complaint. On information and belief, Netflix is either knowingly infringing the '878 Patent or is willfully blind to its infringement — including by ignoring VideoLabs' communications and/or using a

[41] *See* Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

[42] *See* Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

[43] *See* Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

[44] *See id.* at 8-10.

[45] *See id.* at 2-7.

[46] *See id.* at 1-2.

disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs and learning more about VideoLabs' patents — and continues to act in wanton disregard of VideoLabs' patent rights.

99. Despite becoming aware of or willfully blinding itself to its infringement of the '878 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '878 Accused Products. On information and belief, Netflix has made no attempts to design around the '878 Patent or otherwise stop its infringing behavior.

100. Netflix's infringement of the '878 Patent therefore has been and remains willful.

101. Netflix also indirectly infringes the '878 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '878 Accused Products in this District and throughout the United States. As described above, on information and belief, Netflix has known about the '878 Patent since at least February 22, 2022.

102. On information and belief, Netflix has actively induced the infringement of the '878 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix '878 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the '878 Patent in violation of 35 U.S.C. § 271(a).

103. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the '878 Patent by, for example, requiring that third parties provide content to Netflix that has been encoded using H.264, working with third parties to ensure that this requirement is met, and otherwise actively encouraging, supporting, and assisting such infringing

acts.[47]

104. On information and belief, Netflix contributorily infringes the ’878 Patent under 35 U.S.C. § 217(c) by importing, selling, and/or offering to sell within the United States the Netflix ’878 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the hardware and/or software for encoding content with H.264 using CAVLC is material, has no insubstantial non-infringing uses, and is known by Netflix to be especially made or adapted for use in a manner that infringes the ’878 Patent.

COUNT II
INFRINGEMENT OF U.S. PATENT NO. 7,440,559

105. VideoLabs incorporates by reference the foregoing paragraphs of this Complaint as if fully set forth herein.

106. VL IP is the assignee and lawful owner of all right, title, and interest in and to the ’559 Patent. The ’559 Patent is valid and enforceable.

107. On information and belief, Netflix has infringed and continues to infringe the ’559 Patent in violation of 35 U.S.C. § 271, either literally or through the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States products and/or methods that practice at least claims 1 and 13 of the ’559 Patent, including with respect to the Netflix ’559 Accused Products.

108. On information and belief, Netflix uses the Netflix ’559 Accused Products for its own business purposes. In addition, Netflix regularly conducts research, testing, and

[47] *See, e.g.*, Ex. 51, https://partnerhelp.netflixstudios.com/hc/en-us/articles/360057627253-VFX-Media-Review-Delivery-Specifications; Ex. 52, https://drive.google.com/file/d/1_bwbUs4NaF7Y-07_NHwQqsL5L6zPVMPN/view; Ex. 53, https://fta-media.konsole-labs.com/pdf_files/Netflix_Full_Technical_Spec_%26_Operators_Manual_v8.1.pdf.

troubleshooting of the Netflix ’559 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix’s subsidiaries) use the Netflix ’559 Accused Products.

109. On information and belief, Netflix’s infringement through certain instrumentalities of the Netflix streaming service, described below, are exemplary of all of Netflix’s infringement with respect to all the Netflix ’559 Accused Products.

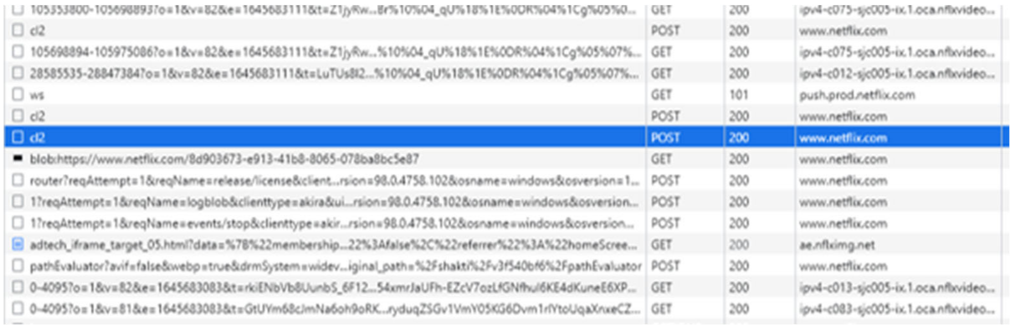
110. The Netflix ’559 Accused Products directly infringe at least claim 1 of the ’559 Patent, for example, by providing one or more servers configured to receive, from a remote terminal accessing the Netflix streaming and content delivery services, content status information, including content bookmark information, and configured to receive server status information regarding available content from a source of content, and in response to the content status information, instructing the terminal to perform actions, such as download control instructions, based on the terminal status information and the server status information, to control the flow of content to the terminal.

111. The Netflix ’559 Accused Products meet every limitation of at least claim 1 of the ‘559 Patent, which recites:

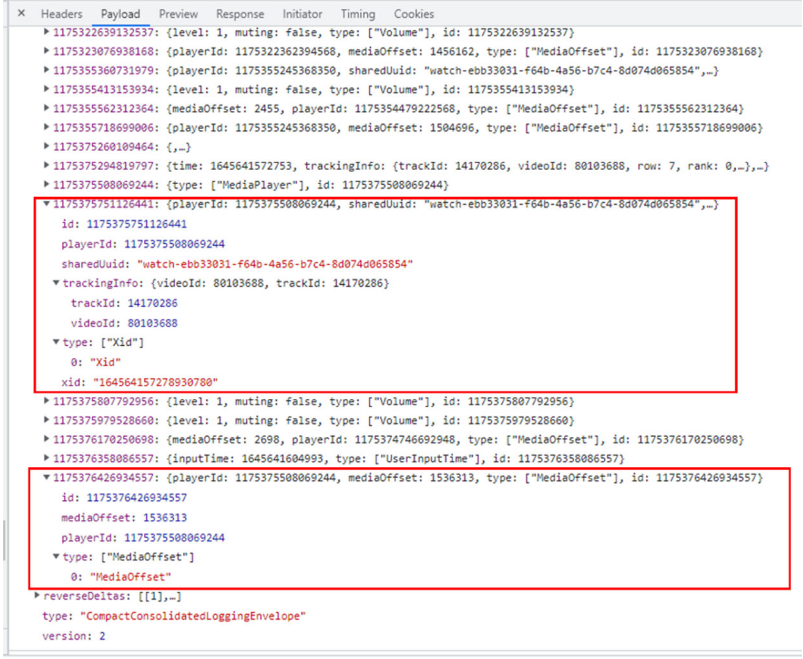
> 1. An apparatus comprising:
>
> a processor configured to receive, from a terminal located remote from the apparatus, a content status including terminal status information, and configured to receive server status information regarding a source of content, wherein the server status information comprises a listing of at least one piece of content available from the source, wherein the processor is configured to send, to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information and the server status information, and
>
> wherein the at least one piece of content available from the source, and the content for which the processor is configured to control the flow, comprise multimedia content.

112. The architecture of the Netflix streaming and content delivery services generally constitutes (among other things) a back-end control plane, which controls and manages the myriad of microservices employed by Netflix to manage and deliver streaming content to Netflix subscribers, a terminal application, accessible directly from a subscriber's browser, or via the Netflix app installed on one or more of a subscriber's connected devices, and a content distribution network in which implementations of Netflix's content library are stored, distributed and managed, and subsequently delivered, to Netflix subscribers on demand.

113. The Netflix back-end control plane incorporates a processor configured to receive, from a terminal located remote from the apparatus, a content status including terminal status information. During a streaming session with a remote terminal accessing the Netflix streaming and content delivery services using, for example, a browser by accessing www.netflix.com, or via the Netflix application installed on the subscriber's connected device, the remote terminal routinely provides content status messages to Netflix servers in the back-end control plane. One such content status message provided in this regard is a content status message ("cl2") provided by the remote terminal to the Netflix domain www.netflix.com, via an API message (highlighted in blue below). This content status message is provided to the Netflix servers in the back-end control plane via a HTTPS POST method and includes, in its message payload, terminal status information about the content the Netflix subscriber is watching or has recently finished watching (red highlighting below).



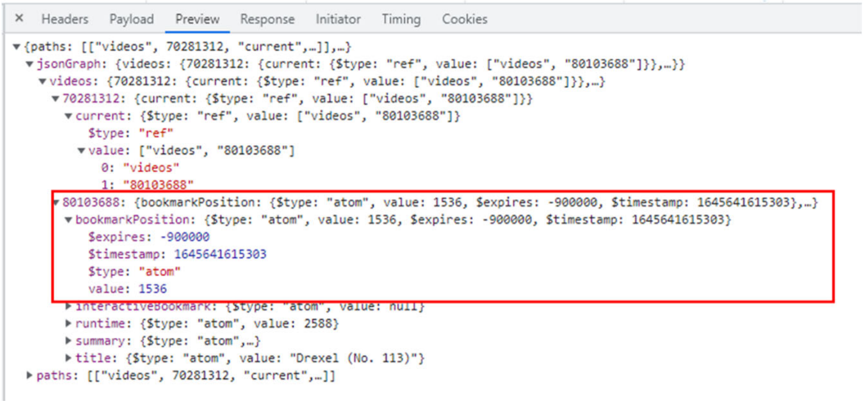
```
□ 105353800-105698893?o=1&v=82&e=1645683111&t=Z1jyRw...8r%10%04_qU%18%1E%0DR%04%1Cg%05%0...  GET   200  ipv4-c075-sjc005-ix.1.oca.nflxvideo...
□ cl2                                                                                        POST  200  www.netflix.com
□ 105698894-105975086?o=1&v=82&e=1645683111&t=Z1jyRw...%10%04_qU%18%1E%0DR%04%1Cg%05%07%...  GET   200  ipv4-c075-sjc005-ix.1.oca.nflxvideo...
□ 28585535-28847384?o=1&v=82&e=1645683111&t=LuTUs8I2...%10%04_qU%18%1E%0DR%04%1Cg%05%07%...  GET   200  ipv4-c012-sjc005-ix.1.oca.nflxvideo...
□ ws                                                                                         GET   101  push.prod.netflix.com
□ cl2                                                                                        POST  200  www.netflix.com
□ cl2                                                                                        POST  200  www.netflix.com
▬ blob:https://www.netflix.com/8d903673-e913-41b8-8065-078ba8bc5e87                          GET   200  www.netflix.com
□ router?reqAttempt=1&reqName=release/license&client...rsion=98.0.4758.102&osname=windows&osversion=1...  POST  200  www.netflix.com
□ 1?reqAttempt=1&reqName=logblob&clienttype=akira&ui...rsion=98.0.4758.102&osname=windows&osversion...   POST  200  www.netflix.com
□ 1?reqAttempt=1&reqName=events/stop&clienttype=akir...rsion=98.0.4758.102&osname=windows&osversion...   POST  200  www.netflix.com
▣ adtech_iframe_target_05.html?data=%7B%22membership...22%3Afalse%2C%22referrer%22%3A%22homeScree...     GET   200  ae.nflximg.net
□ pathEvaluator?avif=false&webp=true&drmSystem=widev...iginal_path=%2Fshakti%2Fv3f540bf6%2FpathEvaluator  POST  200  www.netflix.com
□ 0-4095?o=1&v=82&e=1645683083&t=rkiENbVb8UunbS_6F12...54xmrJaUFh-EZcV7ozLfGNfhul6KE4dKuneE6XP...         GET   200  ipv4-c013-sjc005-ix.1.oca.nflxvideo...
□ 0-4095?o=1&v=81&e=1645683083&t=GtUYm68cJmNa6oh9oRK...ryduqZSGv1VmY05KG6Dvm1rlYtoUqaXnxeCZ...           GET   200  ipv4-c083-sjc005-ix.1.oca.nflxvideo...
```



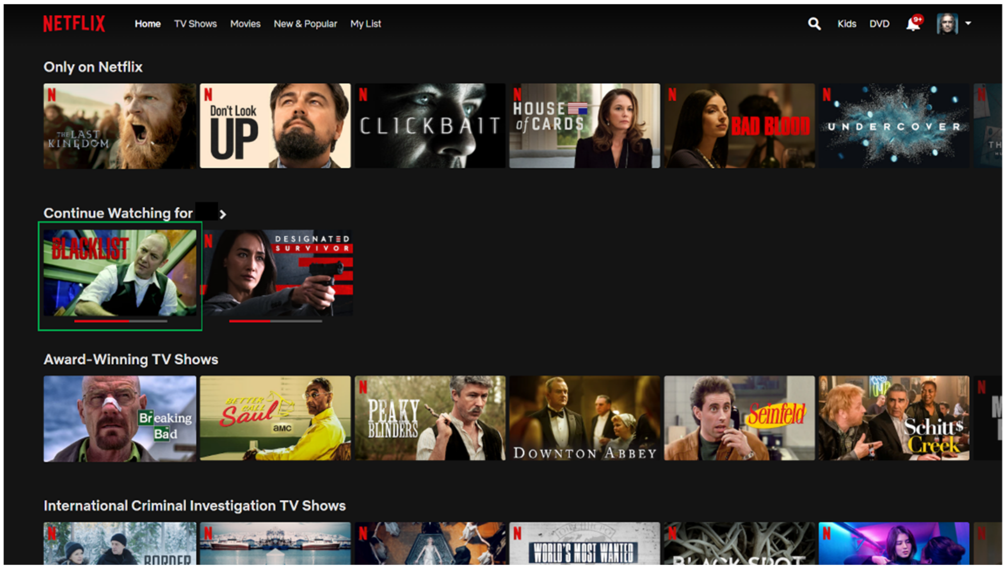
```
  ▸ 1175322639132537: {level: 1, muting: false, type: ["Volume"], id: 1175322639132537}
  ▸ 1175323076938168: {playerId: 1175322362394568, mediaOffset: 1456162, type: ["MediaOffset"], id: 1175323076938168}
  ▸ 1175355360731979: {playerId: 1175355245368350, sharedUuid: "watch-ebb33031-f64b-4a56-b7c4-8d074d065854",…}
  ▸ 1175355413153934: {level: 1, muting: false, type: ["Volume"], id: 1175355413153934}
  ▸ 1175355562312364: {mediaOffset: 2455, playerId: 1175354479222568, type: ["MediaOffset"], id: 1175355562312364}
  ▸ 1175355718699006: {playerId: 1175355245368350, mediaOffset: 1504696, type: ["MediaOffset"], id: 1175355718699006}
  ▸ 1175375260109464: {,…}
  ▸ 1175375294819797: {time: 1645641572753, trackingInfo: {trackId: 14170286, videoId: 80103688, row: 7, rank: 0,…},…}
  ▸ 1175375508069244: {type: ["MediaPlayer"], id: 1175375508069244}
  ▾ 1175375751126441: {playerId: 1175375508069244, sharedUuid: "watch-ebb33031-f64b-4a56-b7c4-8d074d065854",…}
      id: 1175375751126441
      playerId: 1175375508069244
      sharedUuid: "watch-ebb33031-f64b-4a56-b7c4-8d074d065854"
    ▾ trackingInfo: {videoId: 80103688, trackId: 14170286}
        trackId: 14170286
        videoId: 80103688
    ▾ type: ["Xid"]
        0: "Xid"
      xid: "164564157278930780"
  ▸ 1175375807792956: {level: 1, muting: false, type: ["Volume"], id: 1175375807792956}
  ▸ 1175375979528660: {level: 1, muting: false, type: ["Volume"], id: 1175375979528660}
  ▸ 1175376170250698: {mediaOffset: 2698, playerId: 1175374746692948, type: ["MediaOffset"], id: 1175376170250698}
  ▸ 1175376358086557: {inputTime: 1645641604993, type: ["UserInputTime"], id: 1175376358086557}
  ▾ 1175376426934557: {playerId: 1175375508069244, mediaOffset: 1536313, type: ["MediaOffset"], id: 1175376426934557}
      id: 1175376426934557
      mediaOffset: 1536313
      playerId: 1175375508069244
    ▾ type: ["MediaOffset"]
        0: "MediaOffset"
▸ reverseDeltas: [[1],…]
  type: "CompactConsolidatedLoggingEnvelope"
  version: 2
```

114. The content status message received by the Netflix servers in the back-end control plane from the remote terminal specifies, among other information, the view progress within the content the Netflix subscriber has recently watched, but not yet finished watching (in the above example, the content program is represented by videoID 80103688, which correlates to the Netflix show Blacklist: Season 3, Episode 15 – Drexel). For example, the terminal status information relating to the view progress in the above example is reported by the Netflix client player and includes the parameter *mediaOffset*, which represents the offset position within the video program the Netflix client player is/was streaming. This information is passed through to the Netflix back-end control plane and Netflix maintains this terminal status information in its back-end control

plane and uses this information at least to manage the content bookmarking feature in its streaming and content delivery services. For example, Netflix updates the bookmarking information in near real-time based on this received information to reflect the bookmark change, if any, in the client GUI, as shown below:

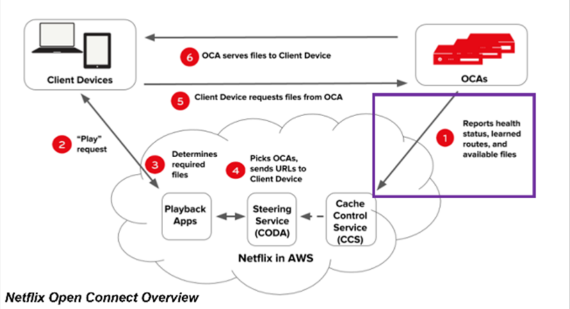


```
▼{paths: [["videos", 70281312, "current",…]],…}
  ▼jsonGraph: {videos: {70281312: {current: {$type: "ref", value: ["videos", "80103688"]}},…}}
    ▼videos: {70281312: {current: {$type: "ref", value: ["videos", "80103688"]}},…}
      ▼70281312: {current: {$type: "ref", value: ["videos", "80103688"]}}
        ▼current: {$type: "ref", value: ["videos", "80103688"]}
            $type: "ref"
          ▼value: ["videos", "80103688"]
              0: "videos"
              1: "80103688"
      ▼80103688: {bookmarkPosition: {$type: "atom", value: 1536, $expires: -900000, $timestamp: 1645641615303},…}
        ▼bookmarkPosition: {$type: "atom", value: 1536, $expires: -900000, $timestamp: 1645641615303}
            $expires: -900000
            $timestamp: 1645641615303
            $type: "atom"
            value: 1536
        ▶interactiveBookmark: {$type: "atom", value: null}
        ▶runtime: {$type: "atom", value: 2588}
        ▶summary: {$type: "atom",…}
        ▶title: {$type: "atom", value: "Drexel (No. 113)"}
  ▶paths: [["videos", 70281312, "current",…]]
```

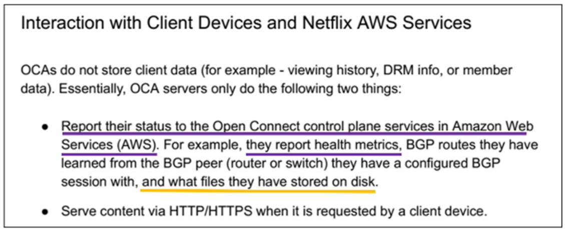


115. The Netflix back-end control plane also incorporates a processor configured to receive server status information regarding a source of content, wherein the server status information comprises a listing of at least one piece of content available from the source. For example, the Netflix back-end control plane manages status information received from the origin servers hosting Netflix content within its content delivery network. This status information

includes health metrics available from the source (*e.g.*, Open Connect Appliances (or “OCAs”)) (*see e.g.*, purple references below) and a listing of content streams of different encoded formats available from the source (*see e.g.*, yellow reference below).[48]


Client Devices
6 OCA serves files to Client Device
OCAs
5 Client Device requests files from OCA
1 Reports health status, learned routes, and available files
2 "Play" request
3 Determines required files
4 Picks OCAs, sends URLs to Client Device
Playback Apps
Steering Service (CODA)
Cache Control Service (CCS)
Netflix in AWS

Netflix Open Connect Overview



Interaction with Client Devices and Netflix AWS Services

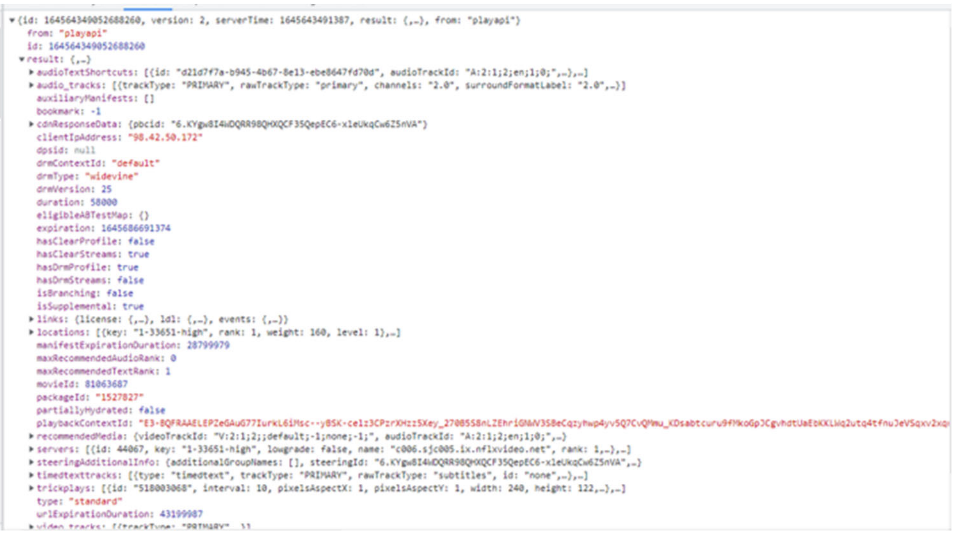
OCAs do not store client data (for example - viewing history, DRM info, or member data). Essentially, OCA servers only do the following two things:

- Report their status to the Open Connect control plane services in Amazon Web Services (AWS). For example, they report health metrics, BGP routes they have learned from the BGP peer (router or switch) they have a configured BGP session with, and what files they have stored on disk.
- Serve content via HTTP/HTTPS when it is requested by a client device.

Netflix Open Connect Overview

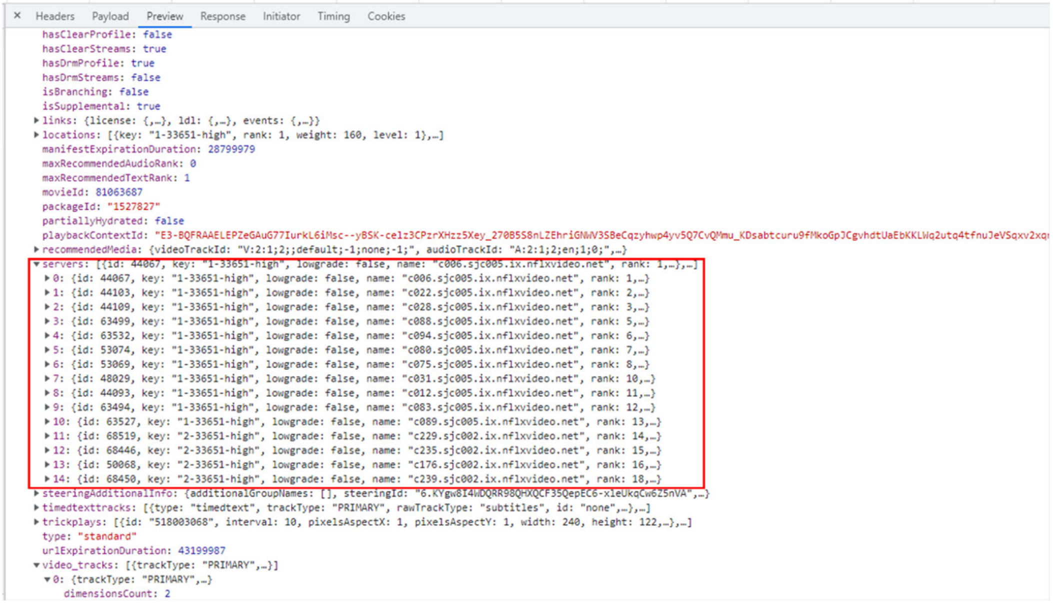
116. The Netflix back-end control plane processor is further configured to send, to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information and the server status information. For example, when the Netflix subscriber requests to stream the content program from the Netflix streaming and content delivery services, the Netflix back-end control plane sends to the subscriber’s terminal a response to the content status. The response is sent, for example, as payload information in a JSON container to the subscriber’s terminal:

[48] *See e.g.*, Ex. 49, “Open Connect Overview”, NETFLIX, *https://openconnect.netflix.com/Open-Connect-Overview.pdf*.

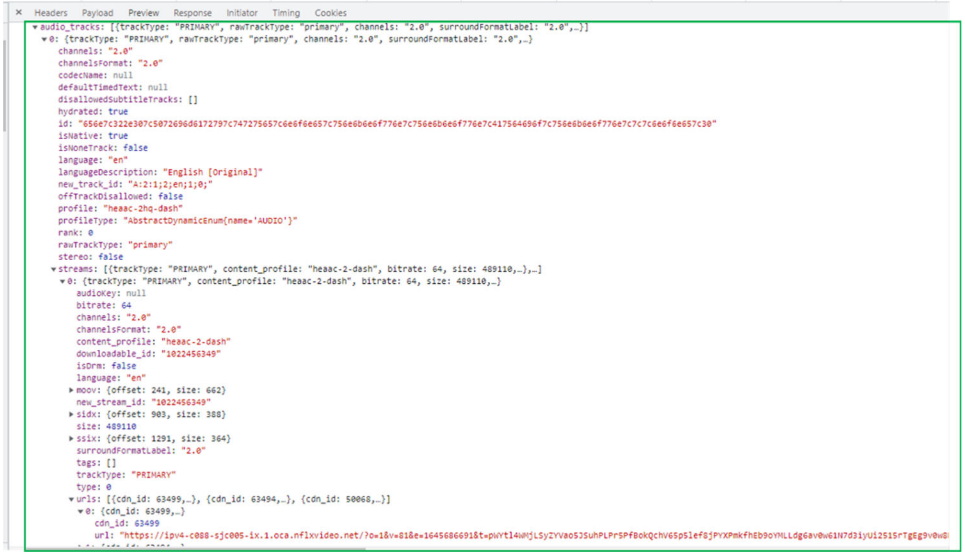


```
▾{id: 16456434905268826O, version: 2, serverTime: 1645643491387, result: {,…}, from: "playapi"}
   from: "playapi"
   id: 16456434905268826O
 ▾result: {,…}
  ▸audioTextShortcuts: [{id: "d21d7f7a-b945-4b67-8e13-ebe8647fd70d", audioTrackId: "A:2:1;2;en;1;0;",…},…]
  ▸audio_tracks: [{trackType: "PRIMARY", rawTrackType: "primary", channels: "2.0", surroundFormatLabel: "2.0",…}]
   auxiliaryManifests: []
   bookmark: -1
  ▸cdnResponseData: {pbcid: "6.KYgw8I4WDQRR98QHXQCF35QepEC6-xleUkqCw6Z5nVA"}
   clientIpAddress: "98.42.50.172"
   dpsid: null
   drmContextId: "default"
   drmType: "widevine"
   drmVersion: 25
   duration: 58000
   eligibleABTestMap: {}
   expiration: 1645686691374
   hasClearProfile: false
   hasClearStreams: true
   hasDrmProfile: true
   hasDrmStreams: false
   isBranching: false
   isSupplemental: true
  ▸links: {license: {,…}, ldl: {,…}, events: {,…}}
  ▸locations: [{key: "1-33651-high", rank: 1, weight: 160, level: 1},…]
   manifestExpirationDuration: 28799979
   maxRecommendedAudioRank: 0
   maxRecommendedTextRank: 1
   movieId: 81063687
   packageId: "1527827"
   partiallyHydrated: false
   playbackContextId: "E3-BQFRAAELEPZeGAuG77IurkL6iMsc--yBSK-celz3CPzrXHzz5Xey_27085S8nLZEhriGNWV3S8eCqzyhwp4yv5Q7CvQMmu_KDsabtcuru9fMkoGpJCgvhdtUaEbKKLWq2utq4tfnuJeVSqxv2xq
  ▸recommendedMedia: {videoTrackId: "V:2:1;2;;default;-1;none;-1;", audioTrackId: "A:2:1;2;en;1;0;",…}
  ▸servers: [{id: 44067, key: "1-33651-high", lowgrade: false, name: "c006.sjc005.ix.nflxvideo.net", rank: 1,…},…]
  ▸steeringAdditionalInfo: {additionalGroupNames: [], steeringId: "6.KYgw8I4WDQRR98QHXQCF35QepEC6-xleUkqCw6Z5nVA",…}
  ▸timedtexttracks: [{type: "timedtext", trackType: "PRIMARY", rawTrackType: "subtitles", id: "none",…},…]
  ▸trickplays: [{id: "518003068", interval: 10, pixelsAspectX: 1, pixelsAspectY: 1, width: 240, height: 122,…},…]
   type: "standard"
   urlExpirationDuration: 43199987
  ▸video_tracks: [{trackType: "PRIMARY",…}]
```

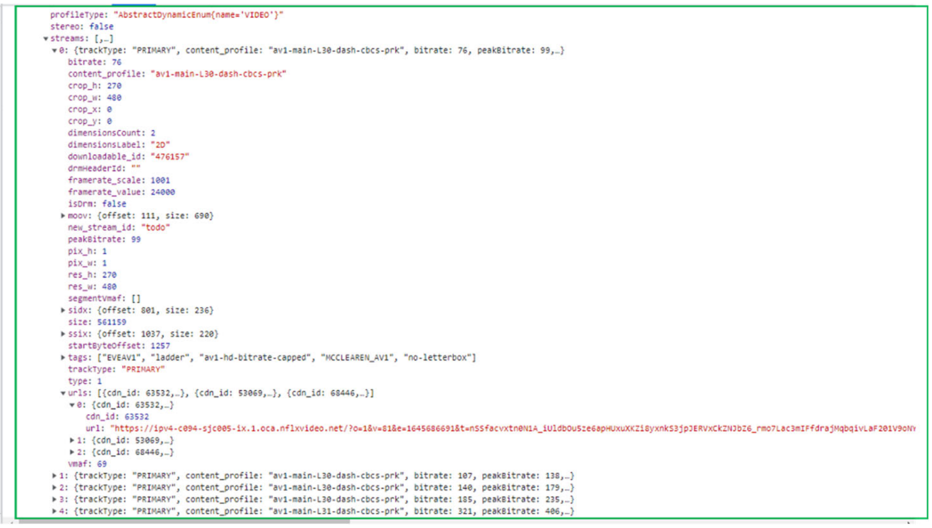
117. The response includes information to instruct the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information and the server status information. For example, the response includes a federated listing of the different origin servers from where the client can obtain the program stream (red highlighting below), as well as the segment index references and base URLs for respective audio and video tracks (green highlighting below) for the content program to enable the Netflix client to obtain the individual segments of the content program stream from the ideal source(s) depending on bandwidth constraints, etc.



```
    hasClearProfile: false
    hasClearStreams: true
    hasDrmProfile: true
    hasDrmStreams: false
    isBranching: false
    isSupplemental: true
  ▶ links: {license: {,…}, ldl: {,…}, events: {,…}}
  ▶ locations: [{key: "1-33651-high", rank: 1, weight: 160, level: 1},…]
    manifestExpirationDuration: 28799979
    maxRecommendedAudioRank: 0
    maxRecommendedTextRank: 1
    movieId: 81063687
    packageId: "1527827"
    partiallyHydrated: false
    playbackContextId: "E3-BQFRAAELEPZeGAuG77IurkL6iMsc--yBSK-celz3CPzrXHzz5Xey_270B5S8nLZEhriGNWV3SBeCqzyhwp4yv5Q7CvQMmu_KDsabtcuru9fMkoGpJCgvhdtUaEbKKLWq2utq4tfnuJeVSqxv2xq
  ▶ recommendedMedia: {videoTrackId: "V:2:1;2;;default;-1;none;-1;", audioTrackId: "A:2:1;2;en;1;0;",…}
  ▼ servers: [{id: 44067, key: "1-33651-high", lowgrade: false, name: "c006.sjc005.ix.nflxvideo.net", rank: 1,…},…]
    ▶ 0: {id: 44067, key: "1-33651-high", lowgrade: false, name: "c006.sjc005.ix.nflxvideo.net", rank: 1,…}
    ▶ 1: {id: 44103, key: "1-33651-high", lowgrade: false, name: "c022.sjc005.ix.nflxvideo.net", rank: 2,…}
    ▶ 2: {id: 44109, key: "1-33651-high", lowgrade: false, name: "c028.sjc005.ix.nflxvideo.net", rank: 3,…}
    ▶ 3: {id: 63499, key: "1-33651-high", lowgrade: false, name: "c088.sjc005.ix.nflxvideo.net", rank: 5,…}
    ▶ 4: {id: 63532, key: "1-33651-high", lowgrade: false, name: "c094.sjc005.ix.nflxvideo.net", rank: 6,…}
    ▶ 5: {id: 53074, key: "1-33651-high", lowgrade: false, name: "c080.sjc005.ix.nflxvideo.net", rank: 7,…}
    ▶ 6: {id: 53069, key: "1-33651-high", lowgrade: false, name: "c075.sjc005.ix.nflxvideo.net", rank: 8,…}
    ▶ 7: {id: 48029, key: "1-33651-high", lowgrade: false, name: "c031.sjc005.ix.nflxvideo.net", rank: 10,…}
    ▶ 8: {id: 44093, key: "1-33651-high", lowgrade: false, name: "c012.sjc005.ix.nflxvideo.net", rank: 11,…}
    ▶ 9: {id: 63494, key: "1-33651-high", lowgrade: false, name: "c083.sjc005.ix.nflxvideo.net", rank: 12,…}
    ▶ 10: {id: 63527, key: "1-33651-high", lowgrade: false, name: "c089.sjc005.ix.nflxvideo.net", rank: 13,…}
    ▶ 11: {id: 68519, key: "2-33651-high", lowgrade: false, name: "c229.sjc002.ix.nflxvideo.net", rank: 14,…}
    ▶ 12: {id: 68446, key: "2-33651-high", lowgrade: false, name: "c235.sjc002.ix.nflxvideo.net", rank: 15,…}
    ▶ 13: {id: 50068, key: "2-33651-high", lowgrade: false, name: "c176.sjc002.ix.nflxvideo.net", rank: 16,…}
    ▶ 14: {id: 68450, key: "2-33651-high", lowgrade: false, name: "c239.sjc002.ix.nflxvideo.net", rank: 18,…}
  ▶ steeringAdditionalInfo: {additionalGroupNames: [], steeringId: "6.KYgw8I4WDQRR98QHXQCF35QepEC6-xleUkqCw6Z5nVA",…}
  ▶ timedtexttracks: [{type: "timedtext", trackType: "PRIMARY", rawTrackType: "subtitles", id: "none",…},…]
  ▶ trickplays: [{id: "518003068", interval: 10, pixelsAspectX: 1, pixelsAspectY: 1, width: 240, height: 122,…},…]
    type: "standard"
    urlExpirationDuration: 43199987
  ▼ video_tracks: [{trackType: "PRIMARY",…}]
    ▼ 0: {trackType: "PRIMARY",…}
        dimensionsCount: 2
```

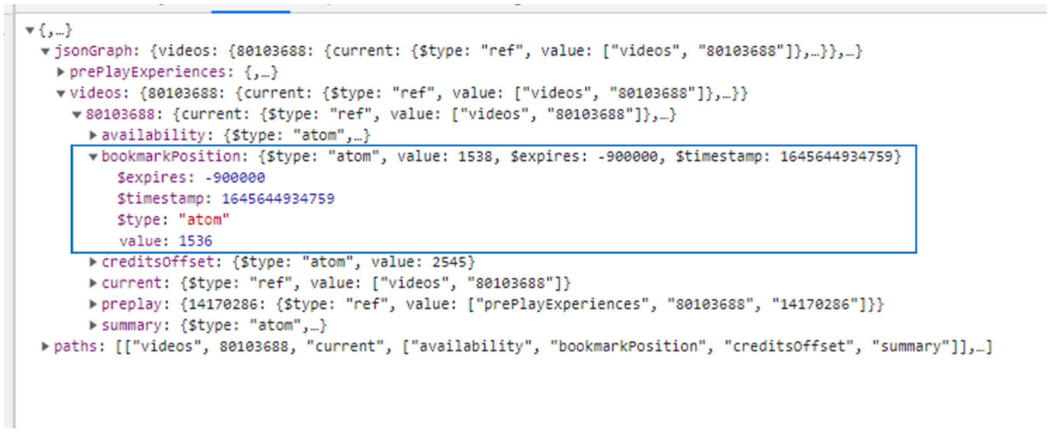


```
▼ audio_tracks: [{trackType: "PRIMARY", rawTrackType: "primary", channels: "2.0", surroundFormatLabel: "2.0",…}]
  ▼ 0: {trackType: "PRIMARY", rawTrackType: "primary", channels: "2.0", surroundFormatLabel: "2.0",…}
      channels: "2.0"
      channelsFormat: "2.0"
      codecName: null
      defaultTimedText: null
      disallowedSubtitleTracks: []
      hydrated: true
      id: "656e7c322e307c5072696d6172797c747275657c6e6f6e657c756e6b6e6f776e7c756e6b6e6f776e7c417564696f7c756e6b6e6f776e7c7c7c6e6f6e657c30"
      isNative: true
      isNoneTrack: false
      language: "en"
      languageDescription: "English [Original]"
      new_track_id: "A:2:1;2;en;1;0;"
      offTrackDisallowed: false
      profile: "heaac-2hq-dash"
      profileType: "AbstractDynamicEnum{name='AUDIO'}"
      rank: 0
      rawTrackType: "primary"
      stereo: false
    ▼ streams: [{trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 489110,…},…]
      ▼ 0: {trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 489110,…}
          audioKey: null
          bitrate: 64
          channels: "2.0"
          channelsFormat: "2.0"
          content_profile: "heaac-2-dash"
          downloadable_id: "1022456349"
          isDrm: false
          language: "en"
        ▶ moov: {offset: 241, size: 662}
          new_stream_id: "1022456349"
        ▶ sidx: {offset: 903, size: 388}
          size: 489110
        ▶ ssix: {offset: 1291, size: 364}
          surroundFormatLabel: "2.0"
          tags: []
          trackType: "PRIMARY"
          type: 0
        ▼ urls: [{cdn_id: 63499,…}, {cdn_id: 63494,…}, {cdn_id: 50068,…}]
          ▼ 0: {cdn_id: 63499,…}
              cdn_id: 63499
              url: "https://ipv4-c088-sjc005-ix.1.oca.nflxvideo.net/?o=1&v=81&e=1645686691&t=pWYtl4WMjLSyZYVao5JSuhPLPr5PfBokQchV65p5lef8jPYXPmkfhEb9oYMLLdg6av0w61N7d3iyUi2515rTgEg9v0w8
```



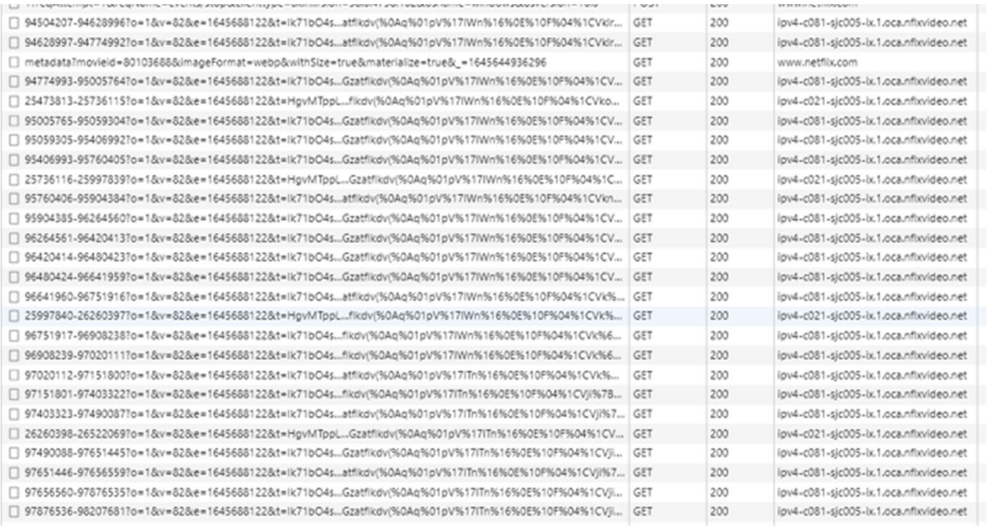
```
      profileType: "AbstractDynamicEnum{name='VIDEO'}"
      stereo: false
    ▼ streams: [,…]
      ▼ 0: {trackType: "PRIMARY", content_profile: "av1-main-L30-dash-cbcs-prk", bitrate: 76, peakBitrate: 99,…}
          bitrate: 76
          content_profile: "av1-main-L30-dash-cbcs-prk"
          crop_h: 270
          crop_w: 480
          crop_x: 0
          crop_y: 0
          dimensionsCount: 2
          dimensionsLabel: "2D"
          downloadable_id: "476157"
          drmHeaderId: ""
          framerate_scale: 1001
          framerate_value: 24000
          isDrm: false
        ▶ moov: {offset: 111, size: 690}
          new_stream_id: "todo"
          peakBitrate: 99
          pix_h: 1
          pix_w: 1
          res_h: 270
          res_w: 480
          segmentVmaf: []
        ▶ sidx: {offset: 801, size: 236}
          size: 561159
        ▶ ssix: {offset: 1037, size: 220}
          startByteOffset: 1257
        ▶ tags: ["EVEAV1", "ladder", "av1-hd-bitrate-capped", "MCCLEAREN_AV1", "no-letterbox"]
          trackType: "PRIMARY"
          type: 1
        ▼ urls: [{cdn_id: 63532,…}, {cdn_id: 53069,…}, {cdn_id: 68446,…}]
          ▼ 0: {cdn_id: 63532,…}
              cdn_id: 63532
              url: "https://ipv4-c094-sjc005-ix.1.oca.nflxvideo.net/?o=1&v=81&e=1645686691&t=nSSfacvxtn0N1A_iUldbOu5ze6apHUxuXKZi8yxnkS3jpJERVxCkZNJbZ6_rmo7Lac3mIFfdrajMqbqivLaF201V9oNY
          ▶ 1: {cdn_id: 53069,…}
          ▶ 2: {cdn_id: 68446,…}
          vmaf: 69
      ▶ 1: {trackType: "PRIMARY", content_profile: "av1-main-L30-dash-cbcs-prk", bitrate: 107, peakBitrate: 138,…}
      ▶ 2: {trackType: "PRIMARY", content_profile: "av1-main-L30-dash-cbcs-prk", bitrate: 140, peakBitrate: 179,…}
      ▶ 3: {trackType: "PRIMARY", content_profile: "av1-main-L30-dash-cbcs-prk", bitrate: 185, peakBitrate: 235,…}
      ▶ 4: {trackType: "PRIMARY", content_profile: "av1-main-L31-dash-cbcs-prk", bitrate: 321, peakBitrate: 406,…}
```

118. The response also includes the bookmark position within the content program at which the Netflix client player should initiate content streaming (blue highlighting below):



```
▼ {,…}
  ▼ jsonGraph: {videos: {80103688: {current: {$type: "ref", value: ["videos", "80103688"]},…}},…}
    ▶ prePlayExperiences: {,…}
    ▼ videos: {80103688: {current: {$type: "ref", value: ["videos", "80103688"]},…}}
      ▼ 80103688: {current: {$type: "ref", value: ["videos", "80103688"]},…}
        ▶ availability: {$type: "atom",…}
        ▼ bookmarkPosition: {$type: "atom", value: 1538, $expires: -900000, $timestamp: 1645644934759}
            $expires: -900000
            $timestamp: 1645644934759
            $type: "atom"
            value: 1536
        ▶ creditsOffset: {$type: "atom", value: 2545}
        ▶ current: {$type: "ref", value: ["videos", "80103688"]}
        ▶ preplay: {14170286: {$type: "ref", value: ["prePlayExperiences", "80103688", "14170286"]}}
        ▶ summary: {$type: "atom",…}
  ▶ paths: [["videos", 80103688, "current", ["availability", "bookmarkPosition", "creditsOffset", "summary"]],…]
```

119. The Netflix application and/or the Netflix browser, under the programming control by Netflix, carries out the instructions by issuing a series of consecutive HTTP GET requests starting at the designated media offset point to the designated content source at specific time intervals thereby controlling the flow of multimedia content to the terminal.



```
□ 94504207-94628996?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVklr...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 94628997-94774992?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVklr...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ metadata?movieid=80103688&imageFormat=webp&withSize=true&materialize=true&_=1645644936296          GET  200  www.netflix.com
□ 94774993-95005764?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 25473813-25736115?o=1&v=82&e=1645688122&t=HgvMTppL...fikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVko...    GET  200  ipv4-c021-sjc005-ix.1.oca.nflxvideo.net
□ 95005765-95059304?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 95059305-95406992?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 95406993-95760405?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 25736116-25997839?o=1&v=82&e=1645688122&t=HgvMTppL...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1C...   GET  200  ipv4-c021-sjc005-ix.1.oca.nflxvideo.net
□ 95760406-95904384?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVkn...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 95904385-96264560?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 96264561-96420413?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 96420414-96480423?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 96480424-96641959?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CV...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 96641960-96751916?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVk%...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 25997840-26260397?o=1&v=82&e=1645688122&t=HgvMTppL...fikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVk%...    GET  200  ipv4-c021-sjc005-ix.1.oca.nflxvideo.net
□ 96751917-96908238?o=1&v=82&e=1645688122&t=Ik71bO4s...fikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVk%6...    GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 96908239-97020111?o=1&v=82&e=1645688122&t=Ik71bO4s...fikdv(%0Aq%01pV%17IWn%16%0E%10F%04%1CVk%6...    GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 97020112-97151800?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVk%...   GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 97151801-97403322?o=1&v=82&e=1645688122&t=Ik71bO4s...fikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVJI%7B...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 97403323-97490087?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVJI%7...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 26260398-26522069?o=1&v=82&e=1645688122&t=HgvMTppL...Gzatfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CV...  GET  200  ipv4-c021-sjc005-ix.1.oca.nflxvideo.net
□ 97490088-97651445?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVJi...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 97651446-97656559?o=1&v=82&e=1645688122&t=Ik71bO4s...atfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVJI%7...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 97656560-97876535?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVJi...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
□ 97876536-98207681?o=1&v=82&e=1645688122&t=Ik71bO4s...Gzatfikdv(%0Aq%01pV%17ITn%16%0E%10F%04%1CVJi...  GET  200  ipv4-c081-sjc005-ix.1.oca.nflxvideo.net
```

120. At least one piece of content available from the source, and the content for which the processor is configured to control the flow, comprise multimedia content. As noted, all the content available for streaming from the Netflix streaming and content delivery services is multimedia content.

121. On information and belief, the "Skip Intro," "Skip Recap," and "Autoplay"/"Next Episode" features operate substantially the same as described above, and thus infringe for substantially the same reasons.

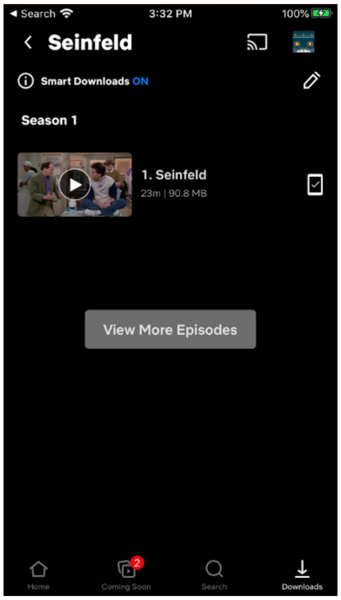
122. The Netflix '559 Accused Products also directly infringe at least claim 13 of the '559 Patent, for example, by receiving, at a network entity, from a remote terminal accessing the Netflix streaming and content delivery services, content status information, including content download information, and in response to the content status information, sending instructions to the terminal to perform actions, such as delete and download control instructions, based on the terminal status information, to control the flow of multimedia content to the terminal.

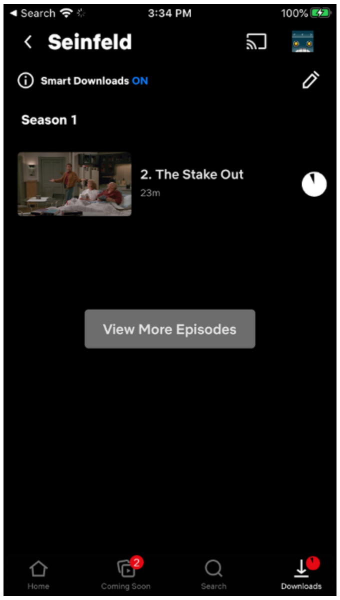
123. The Netflix '559 Accused Products meet every limitation of at least claim 13 of the '559 Patent, which recites:

> 13. A method for controlling a flow of content, the method comprising:
>
> receiving, at a network entity from a terminal located remote therefrom, a content status including terminal status information comprising a listing of at least one piece of content stored in a memory of the terminal; and sending, from the network entity to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information , and
>
> wherein the at least one piece of content stored in the memory of the terminal, and the content for which the flow is controlled, comprise multimedia content.

124. The architecture of the Netflix streaming and content delivery services generally constitutes (among other things) a back-end control plane, which controls and manages the myriad of microservices employed by Netflix to manage and deliver streaming content to Netflix subscribers, a terminal application, accessible directly from a subscriber's browser, or via the Netflix app installed on one or more of a subscriber's connected devices, and a content distribution network in which implementations of Netflix's content library are stored, distributed and managed, and subsequently delivered, to Netflix subscribers on demand.

125. One important feature that Netflix offers its subscribers is content downloads. For example, Netflix subscribers can download Netflix programming content, such as episodes of Seinfeld, to their connected mobile device(s) and watch the programming off-line. Netflix remotely manages the remote client by controlling the deletion and addition of Netflix content programming on the client device.



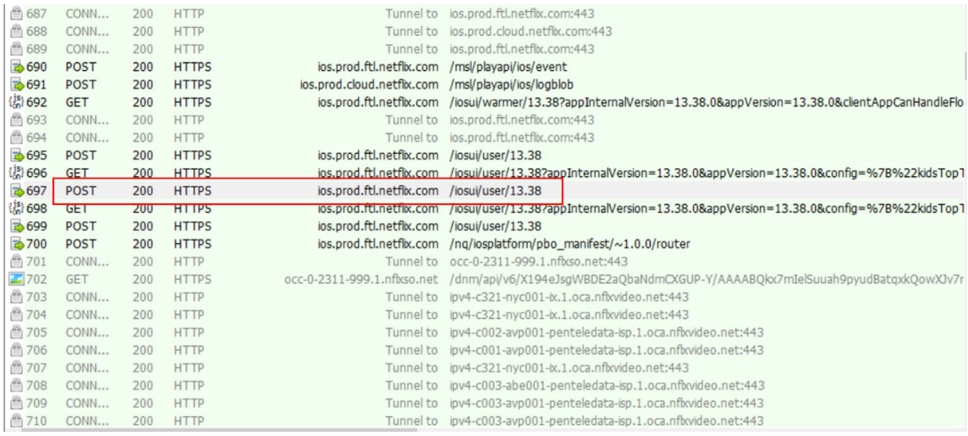


126. Netflix refers to this important feature as "smart downloads".[49] With smart downloads, when a subscriber is finished watching a downloaded episode, Smart Downloads will delete it, and then automatically download the next episode.

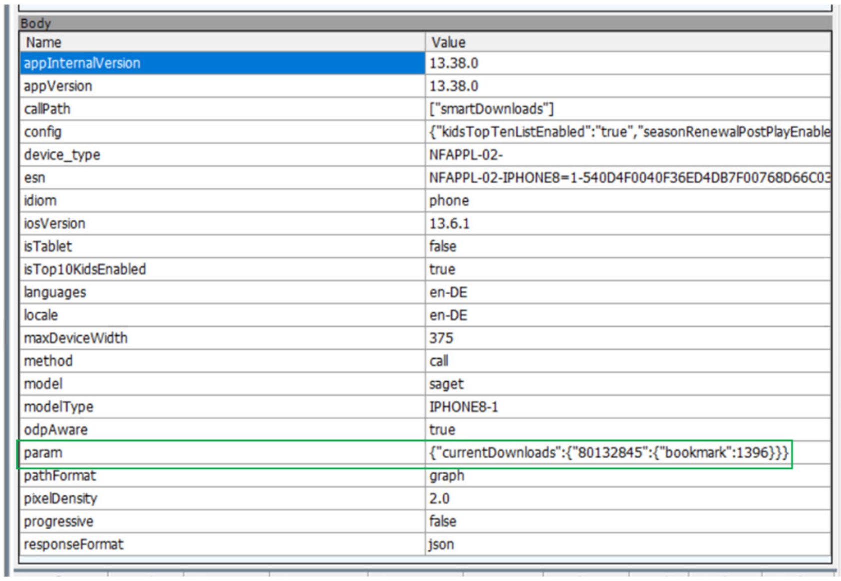
127. To accomplish this, Netflix servers receive, from a remote terminal, a content status including terminal status information. During a streaming session with a remote terminal accessing the Netflix streaming and content delivery services using, for example, the Netflix application installed on the subscriber's connected device, the remote terminal routinely provides content status messages to Netflix servers in the back-end control plane. One such content status message provided in this regard is a content status message ("/iosui/user/13.38") provided by the remote terminal to the Netflix domain ios.prod.ftl.netflix.com, via an API message [red highlighting below]. This content status message is provided to the Netflix servers in the back-end control plane via a HTTPS POST method and includes, in its message payload, terminal status information about downloaded content the Netflix subscriber is watching or has recently finished

[49] *See* Ex. 31, https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1.

watching (for example, the first episode of season 1 of Seinfeld, which correlates to the program reference "80132845") [green highlighting below].



687	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
688	CONN...	200	HTTP	Tunnel to	ios.prod.cloud.netflix.com:443
689	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
690	POST	200	HTTPS	ios.prod.ftl.netflix.com	/msl/playapi/ios/event
691	POST	200	HTTPS	ios.prod.cloud.netflix.com	/msl/playapi/ios/logblob
692	GET	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/warmer/13.38?appInternalVersion=13.38.0&appVersion=13.38.0&clientAppCanHandleFlo
693	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
694	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
695	POST	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38
696	GET	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38?appInternalVersion=13.38.0&appVersion=13.38.0&config=%7B%22kidsTop1
697	POST	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38
698	GET	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38?appInternalVersion=13.38.0&appVersion=13.38.0&config=%7B%22kidsTop1
699	POST	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38
700	POST	200	HTTPS	ios.prod.ftl.netflix.com	/nq/iosplatform/pbo_manifest/~1.0.0/router
701	CONN...	200	HTTP	Tunnel to	occ-0-2311-999.1.nflxso.net:443
702	GET	200	HTTPS	occ-0-2311-999.1.nflxso.net	/dnm/api/v6/X194eJsgWBDE2aQbaNdmCXGUP-Y/AAAABQkx7mIelSuuah9pyudBatqxkQowXJv7r
703	CONN...	200	HTTP	Tunnel to	ipv4-c321-nyc001-ix.1.oca.nflxvideo.net:443
704	CONN...	200	HTTP	Tunnel to	ipv4-c321-nyc001-ix.1.oca.nflxvideo.net:443
705	CONN...	200	HTTP	Tunnel to	ipv4-c002-avp001-penteledata-isp.1.oca.nflxvideo.net:443
706	CONN...	200	HTTP	Tunnel to	ipv4-c001-avp001-penteledata-isp.1.oca.nflxvideo.net:443
707	CONN...	200	HTTP	Tunnel to	ipv4-c321-nyc001-ix.1.oca.nflxvideo.net:443
708	CONN...	200	HTTP	Tunnel to	ipv4-c003-abe001-penteledata-isp.1.oca.nflxvideo.net:443
709	CONN...	200	HTTP	Tunnel to	ipv4-c003-avp001-penteledata-isp.1.oca.nflxvideo.net:443
710	CONN...	200	HTTP	Tunnel to	ipv4-c003-avp001-penteledata-isp.1.oca.nflxvideo.net:443

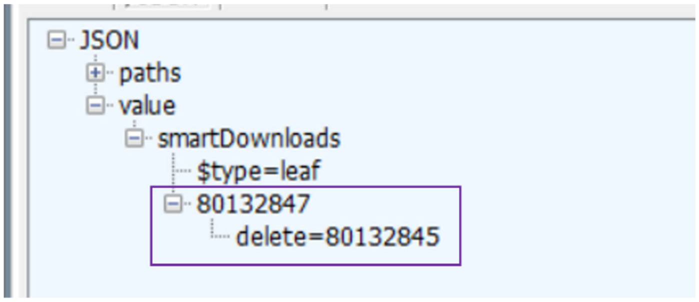


Body

Name	Value
appInternalVersion	13.38.0
appVersion	13.38.0
callPath	["smartDownloads"]
config	{"kidsTopTenListEnabled":"true","seasonRenewalPostPlayEnable
device_type	NFAPPL-02-
esn	NFAPPL-02-IPHONE8=1-540D4F0040F36ED4DB7F00768D66C03
idiom	phone
iosVersion	13.6.1
isTablet	false
isTop10KidsEnabled	true
languages	en-DE
locale	en-DE
maxDeviceWidth	375
method	call
model	saget
modelType	IPHONE8-1
odpAware	true
param	{"currentDownloads":{"80132845":{"bookmark":1396}}}
pathFormat	graph
pixelDensity	2.0
progressive	false
responseFormat	json

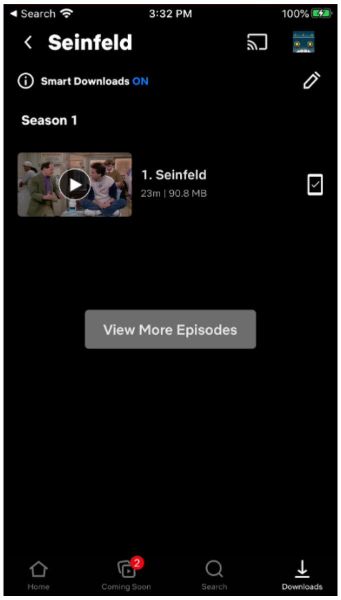
128. The content status message received by the Netflix servers in the back-end control plane from the remote terminal specifies, among other information, the view progress within the content the Netflix subscriber has recently watched locally on their connected device. For example, the terminal status information relating to the view progress in the above example is specified by an API schema defined by Netflix and includes the parameter *bookmark*. Netflix maintains this terminal status information in its back-end control plane and uses this information at least to manage the smart download feature in its streaming and content delivery services.

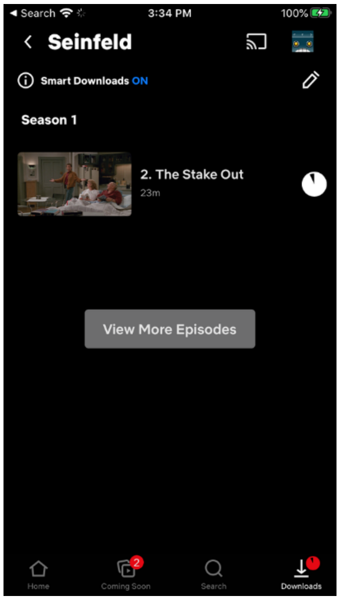
129. The Netflix back-end control plane sends, to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information. For example, when the content status indicates that the Netflix subscriber has completed watching the locally stored content program, the Netflix back-end control plane sends to the subscriber's terminal a response to the content status [purple highlighting below]. The response is sent, for example, as payload information in a JSON container to the subscriber's terminal that includes information to instruct the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information.



```
⊟ JSON
    ⊞ paths
    ⊟ value
        ⊟ smartDownloads
            $type=leaf
            ⊟ 80132847
                delete=80132845
```

130. The Netflix application, under the programming control by Netflix, carries out the instructions by flagging the episode ("80132845") for deletion from the local memory of the client and initiate downloading of the next episode ("80132847") in the series to the client. The Netflix client automatically executes the instructions accordingly to update the Smart Downloads on the client device.





131. At least one piece of content stored in the memory of the terminal, and the content for which the flow is controlled, comprise multimedia content. As noted, all the content available for streaming from the Netflix streaming and content delivery services is multimedia content.

132. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.[50] Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.[51] Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory call on March 19, 2021.[52] Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.[53] After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a "standstill" for an indefinite period of time.[54] Given that nearly two years had already

[50] *See* Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

[51] *See* Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

[52] *See* Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

[53] *See id.* at 8-10.

[54] *See id.* at 2-7.

passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs' communications. On February 22, 2022, VideoLabs sent an email to Netflix apprising it of the '559 Patent, identifying the Netflix products that infringe the '559 Patent, and requesting that the parties discuss the terms for Netflix to take a license to the '559 Patent.[55]

133. Netflix of course knows how its products operate, and on information and belief, upon receiving notice of the '559 Patent, has begun investigating the '559 Patent and its infringement of the Netflix '559 Accused Products. Netflix has been given further notice of the '559 Patent and its infringement of the '559 Patent through the filing of this Complaint. On information and belief, Netflix is either knowingly infringing the '559 Patent or is willfully blind to its infringement — including by ignoring VideoLabs' communications and/or using a disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs and learning more about VideoLabs' patents — and continues to act in wanton disregard of VideoLabs' patent rights.

134. Despite becoming aware of or willfully blinding itself to its infringement of the '559 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '559 Accused Products. On information and belief, Netflix has made no attempts to design around the '559 Patent or otherwise stop its infringing behavior.

135. Netflix's infringement of the '559 Patent therefore has been and remains willful.

136. Netflix also indirectly infringes the '559 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '559 Accused

[55] *See id.* at 1-2.

Products in this District and throughout the United States. As described above, on information and belief, Netflix has known about the ’559 Patent since at least February 22, 2022.

137. On information and belief, Netflix has actively induced the infringement of the ’559 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix ’559 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the ’559 Patent in violation of 35 U.S.C. § 271(a).

138. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the ’559 Patent by, for example, marketing the Netflix ’559 Accused Products and infringing functionalities to consumers; working with consumers to implement, install and/or operate the Netflix ’559 Accused Products and infringing functionalities; fully supporting and managing consumers’ continuing use of the Netflix ’559 Accused Products and infringing functionalities; and providing technical assistance to consumers during their continued use of the ‘559 Accused Products and infringing functionalities.[56]

139. For example, Netflix induces third parties to infringe the ’559 Patent by encouraging them to install and operate the Netflix streaming service, which alone and/or in combination with the third parties’ devices constitutes infringement of the ’559 Patent. Netflix advertises and promotes its Netflix streaming service on its website and in various app stores such

[56] *See e.g.,* Ex. 54, *Netflix Help Center*, https://help.netflix.com/en/ (last accessed Feb. 15, 2022); *See* Ex. 35, *How to Autoplay the Next Episode*, NETFLIX, https://help.netflix.com/en/node/121518; Ex. 31, *Downloading on Netflix Just Got Smarter*, NETFLIX, https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1 (last visited Feb. 17, 2022); *See* Ex. 32, *How to Skip Intros on TV Shows*, NETFLIX, https://help.netflix.com/en/node/63402 (last visited Feb. 17, 2022); Ex. 55, *How to Use “Download Next Episode”,* NETFLIX, https://help.netflix.com/en/node/101262 (last visited Feb. 21, 2022).

as Apple's app store and Android's app store in connection with the Netflix mobile application that can be installed on consumers' respective connected iOS and Android devices (as well as others), and encourages consumers to configure and operate their mobile and computer devices in an infringing manner.[57] In response, consumers acquire, configure, and operate the Netflix streaming service in an infringing manner.

140. On information and belief, Netflix contributorily infringes the '559 Patent under 35 U.S.C. § 217(c) by importing, selling, and/or offering to sell within the United States the Netflix '559 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the Netflix streaming application, and the code for sending, receiving, and/or processing status information, are material, has no insubstantial non-infringing uses, and are known by Netflix to be especially made or adapted for use in a manner that infringes the '559 Patent.

COUNT III
INFRINGEMENT OF U.S. PATENT NO. 7,233,790

141. VideoLabs incorporates by reference the foregoing paragraphs of this Complaint as if fully set forth herein.

142. VL is the assignee and lawful owner of all right, title, and interest in and to the '790 Patent. The '790 Patent is valid and enforceable.

143. On information and belief, Netflix has infringed and continues to infringe the '790 Patent in violation of 35 U.S.C. § 271(a), either literally or through the doctrine of equivalents, by using methods and/or taking steps that practice at least claim 2 of the '790 Patent, including with respect to the Netflix '790 Accused Products.

[57] *See e.g.,* Ex. 54, *Netflix Help Center*, https://help.netflix.com/en/ (last accessed Feb. 15, 2022).

144. On information and belief, Netflix uses the ’790 Accused Products for its own business purposes. In addition, Netflix regularly conducts testing and troubleshooting of the Netflix ’790 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix’s subsidiaries) use the ’790 Accused Products.

145. On information and belief, Netflix’s infringement through certain instrumentalities of the Netflix streaming service, described below, are exemplary of all of Netflix’s infringement with respect to all the Netflix ’790 Accused Products.

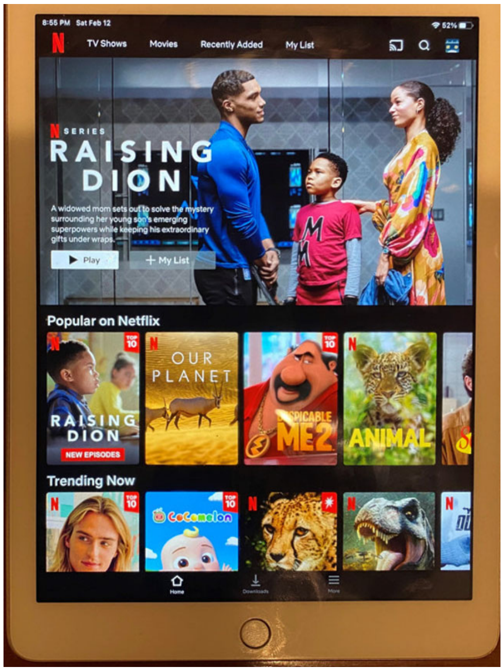
146. The Netflix ’790 Accused Products directly infringe at least claim 2 of the ’790 Patent, for example, by providing subscribers access to content on their respective connected devices through the receipt and storage of different implementations of Netflix content programming, each corresponding to a different set of device capabilities, in its content management system, and the selection of a portion of the Netflix content programming catalog which includes references to each content program implementation based on capabilities of the subscriber’s device.

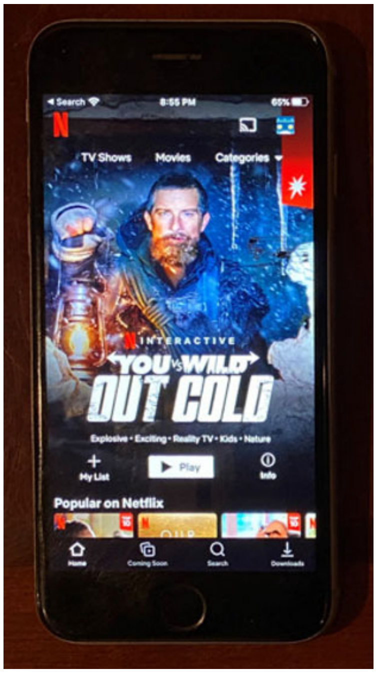
147. The Netflix ’790 Accused Products meet every limitation of at least claim 2 of the ’790 Patent, which recites:

> 2. A method of providing access to digital content for use on wireless communication devices, the method comprising:
>
> receiving and storing in a server system a plurality of items of digital content to be made available for use in wireless communication devices used by a plurality of wireless services subscribers, including receiving and storing a plurality of different implementations of at least one of the items of digital content, where each implementation of any given item of digital content corresponds to a different set of device capabilities;
>
> operating the server system to maintain a product catalog containing a description of the items of digital content, wherein the product catalog includes, in association with each item of digital content, a reference to each implementation of said item of digital content;

> receiving a request from a wireless device used by one of the subscribers;
>
> in response to the request, selecting a portion of the product catalog to be presented to the subscriber, based on device capabilities of the wireless device used by the subscriber; and
>
> presenting the selected portion of the product catalog to the subscriber via a wireless network, such that the selected portion, as presented to the subscriber, provides only a single description of each item of digital content in said portion, regardless of the number of implementations of each said item.

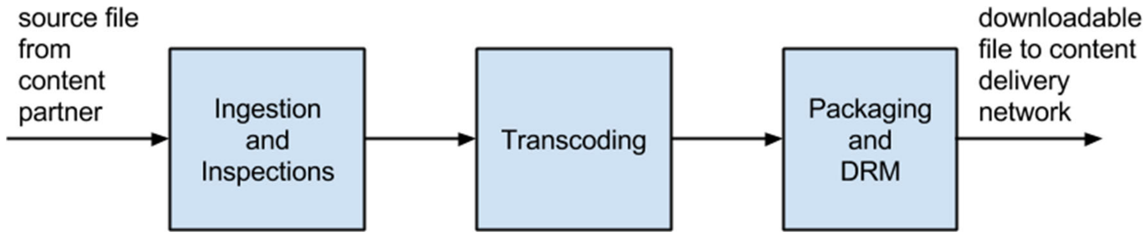
148. The Netflix streaming service, when accessed by a Netflix subscriber, provides access to content for use on subscribers' wireless communication devices. Netflix subscribers access the Netflix streaming service, for example, via the Netflix app on their connected device or by accessing the website from their browser at www.netflix.com:





149. Netflix receives and stores a plurality of items of content to be made available for use in wireless communication devices used by a plurality of wireless services subscribers. Specifically, Netflix controls the ingestion and management of a comprehensive catalog of

programming content that is made available to mobile device users via the Netflix app or by accessing the Netflix OTT streaming service at www.netlix.com. Original format content programming is received by Netflix in its content processing pipeline.[58]


source file from content partner
Ingestion and Inspections
Transcoding
Packaging and DRM
downloadable file to content delivery network

150. For every Interoperability Master Format Package (IMP) delivered to Netflix for a specific title, Netflix performs transfer/delivery validations on the contents of the IMP before updating the content catalog in its asset management system as the content is ingested into the Netflix content management system.[59] Netflix receives IMF package contents via its Backlot portal.[60] Netflix also receives production related materials from Netflix production content creators via its Content Hub asset storage, management, collaboration, and delivery platform.[61]

151. As part of its content ingestion workflow Netflix transcodes the raw mezzanine content into multiple different implementation formats, each corresponding to a different set of device capabilities. This video encoding pipeline runs on AWS EC2 spot instances to seamlessly scale up additional encoders when more titles need to be processed into the system. Specifically, Netflix employs a parallel encoding process to ingest high quality video sources and generate video

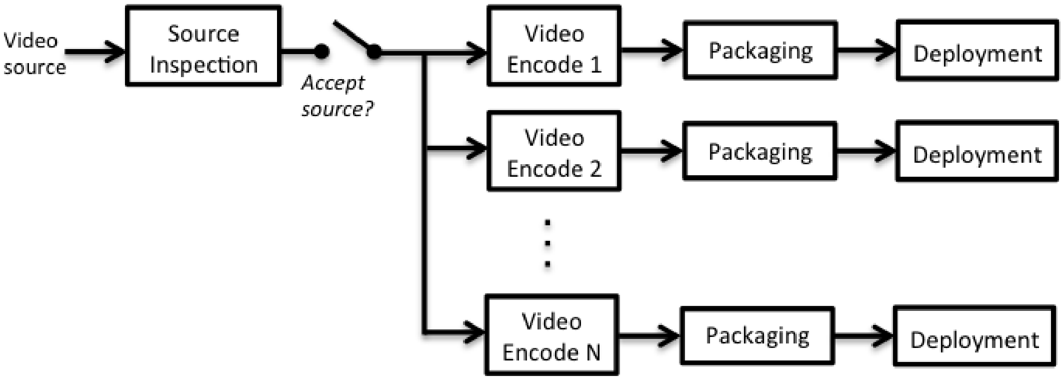
[58] *See* Ex. 56, "The Netflix IMF Workflow", NETFLIX, *https://netflixtechblog.com/the-netflix-imf-workflow-f45dd72ed700,* (last retrieved, Feb. 21, 2022).

[59] *Id.*

[60] *See e.g.,* Ex. 57, *https://partnerhelp.netflixstudios.com/hc/en-us/articles/115004926427-Backlot-Overview-for-Content-Partners* (last retrieved, Feb. 21, 2022).

[61] *See e.g.,* Ex. 58, *https://partnerhelp.netflixstudios.com/hc/en-us/articles/360000509487-Content-Hub-Overview-Introduction* (last retrieved, Feb. 21, 2022).

encodes of various codec profiles, at multiple quality representations per profile.[62] Exemplary codec profiles supported by Netflix transcodes include VP9, AV1, H.264/AVC Main, H.264/AVC Mobile-Hi, and HEVC.[63]



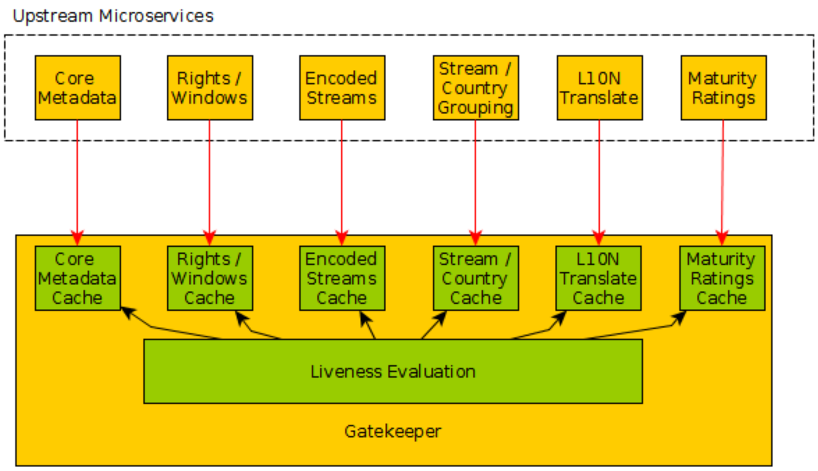
152. Netflix operates its content management services to maintain a product catalog containing a description of the items of digital content. Netflix maintains an enormous, ever-changing content library. Recently, its US content library alone exceeded 6,000 titles.[64] Netflix leverages its Gatekeeper service to evaluate the "liveness" of the titles in its catalog – a title is not available to a subscriber until Gatekeeper approves it.[65]

[62] *See e.g.,* Ex. 6, "High Quality Video Encoding at Scale", NETFLIX, https://netflixtechblog.com/high-quality-video-encoding-at-scale-d159db052746 (last retrieved, Feb. 21, 2022)

[63] *Id.*; *see also* Ex. 59, "Netflix Now Streaming AV1 on Android", NETFLIX, https://netflixtechblog.com/netflix-now-streaming-av1-on-android-d5264a515202 (last retrieved, Feb. 21, 2022); *See* Ex. 26, "More Efficient Mobile Encodes for Netflix Downloads", NETFLIX, https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909 (last retrieved Feb. 21, 2022).

[64] *See e.g.,* Ex. 60, *https://www.whats-on-netflix.com/news/netflix-library-by-the-numbers-2021/* (last retrieved, Feb. 21, 2022).

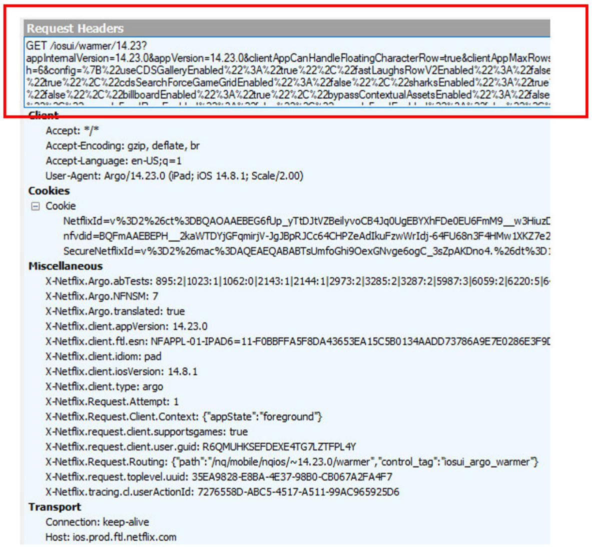
[65] *See* Ex. 61, "Re-Architecting the Video Gatekeeper", NETFLIX, https://netflixtechblog.com/re-architecting-the-video-gatekeeper-f7b0ac2f6b00 (last retrieved, Feb. 21, 2022).



153. In association with each title in the content catalog, Netflix maintains a reference association to each individual program encode so that only certain encodes are chosen for a particular user device. For example, Netflix employs a video metadata service that includes links to content program images, trailers, encoded video files, subtitles, individual episodes and seasons (if applicable), title, genre, synopsis, cast, maturity ratings, etc.[66]

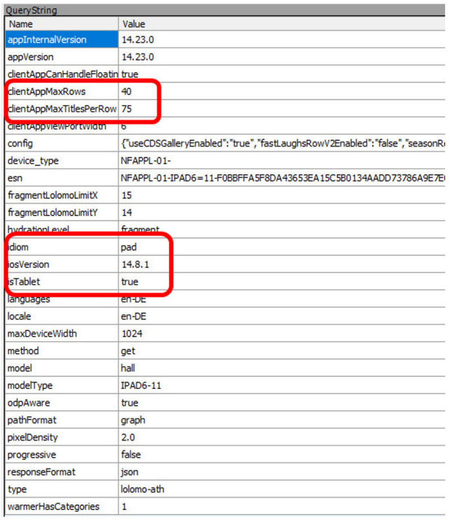
154. Netflix receives a request from a wireless device used by one of the subscribers, such as when the Netflix subscriber launches the Netflix app on their connected mobile device to access the Netflix service. When Netflix subscribers access the streaming service from their respective connected device, an HTTP GET request is sent from the subscriber's device to Netflix:

[66] *See e.g.,* Ex. 62, https://netflixtechblog.com/object-cache-for-scaling-video-metadata-management-c3c17830983e.



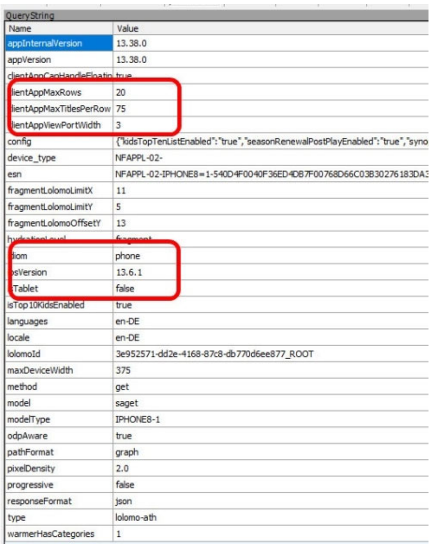
```
Request Headers
GET /iosui/warmer/14.23?
appInternalVersion=14.23.0&appVersion=14.23.0&clientAppCanHandleFloatingCharacterRow=true&clientAppMaxRows
h=6&config=%7B%22useCDSGalleryEnabled%22%3A%22true%22%2C%22fastLaughsRowV2Enabled%22%3A%22false
%22true%22%2C%22cdsSearchForceGameGridEnabled%22%3A%22false%22%2C%22sharksEnabled%22%3A%22true
%22false%22%2C%22billboardEnabled%22%3A%22true%22%2C%22bypassContextualAssetsEnabled%22%3A%22false
Client
    Accept: */*
    Accept-Encoding: gzip, deflate, br
    Accept-Language: en-US;q=1
    User-Agent: Argo/14.23.0 (iPad; iOS 14.8.1; Scale/2.00)
Cookies
⊟ Cookie
        NetflixId=v%3D2%26ct%3DBQAOAAEBEG6fUp_yTtDJtVZBeilyvoCB4Jq0UgEBYXhFDe0EU6FmM9__w3HiuzD
        nfvdid=BQFmAAEBEPH__2kaWTDYjGFqmirjV-JgJBpRJCc64CHPZeAdIkuFzwWrIdj-64FU68n3F4HMw1XKZ7e2
        SecureNetflixId=v%3D2%26mac%3DAQEAEQABABTsUmfoGhi9OexGNvge6ogC_3sZpAKDno4.%26dt%3D
Miscellaneous
    X-Netflix.Argo.abTests: 895:2|1023:1|1062:0|2143:1|2144:1|2973:2|3285:2|3287:2|5987:3|6059:2|6220:5|6
    X-Netflix.Argo.NFNSM: 7
    X-Netflix.Argo.translated: true
    X-Netflix.client.appVersion: 14.23.0
    X-Netflix.client.ftl.esn: NFAPPL-01-IPAD6=11-F0BBFFA5F8DA43653EA15C5B0134AADD73786A9E7E0286E3F9D
    X-Netflix.client.idiom: pad
    X-Netflix.client.iosVersion: 14.8.1
    X-Netflix.client.type: argo
    X-Netflix.Request.Attempt: 1
    X-Netflix.Request.Client.Context: {"appState":"foreground"}
    X-Netflix.request.client.supportsgames: true
    X-Netflix.request.client.user.guid: R6QMUHKSEFDEXE4TG7LZTFPL4Y
    X-Netflix.Request.Routing: {"path":"/nq/mobile/nqios/~14.23.0/warmer","control_tag":"iosui_argo_warmer"}
    X-Netflix.request.toplevel.uuid: 35EA9828-E8BA-4E37-98B0-CB067A2FA4F7
    X-Netflix.tracing.cl.userActionId: 7276558D-ABC5-4517-A511-99AC965925D6
Transport
    Connection: keep-alive
    Host: ios.prod.ftl.netflix.com
```

155. Associated with the request are various capabilities of the subscriber's device, including, for example, the maximum number of rows that can be displayed on the device ("clientAppMaxRows"); the maximum number of content items that can be shown in each row ("clientAppMaxTitlesPerRow"); the type of device the subscriber is using ("idiom"); the version of operating system the device is using ("iosVersion"); and a flag indicating whether the device is a tablet or mobile device ("isTablet"). Representative images from an iPad (left) and an iPhone (right) accessing the Netflix service are shown below:



QueryString

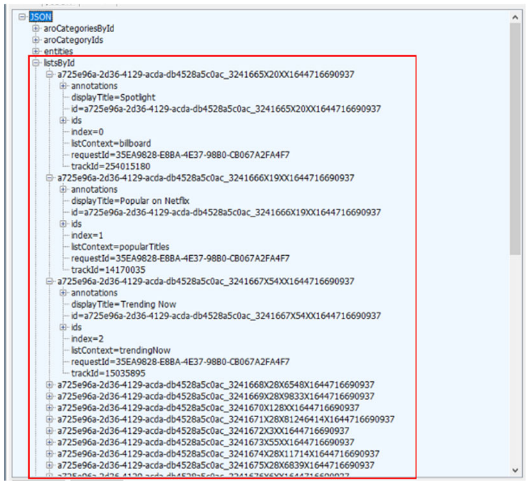
Name	Value
appInternalVersion	14.23.0
appVersion	14.23.0
clientAppCanHandleFloatin	true
clientAppMaxRows	40
clientAppMaxTitlesPerRow	75
clientAppViewPortWidth	6
config	{"useCDSGalleryEnabled":"true","fastLaughsRowV2Enabled":"false","seasonR
device_type	NFAPPL-01-
esn	NFAPPL-01-IPAD6=11-F0BBFFA5F8DA43653EA15C5B0134AADD73786A9E7E
fragmentLolomoLimitX	15
fragmentLolomoLimitY	14
hydrationLevel	fragment
idiom	pad
iosVersion	14.8.1
isTablet	true
languages	en-DE
locale	en-DE
maxDeviceWidth	1024
method	get
model	hall
modelType	IPAD6-11
odpAware	true
pathFormat	graph
pixelDensity	2.0
progressive	false
responseFormat	json
type	lolomo-ath
warmerHasCategories	1



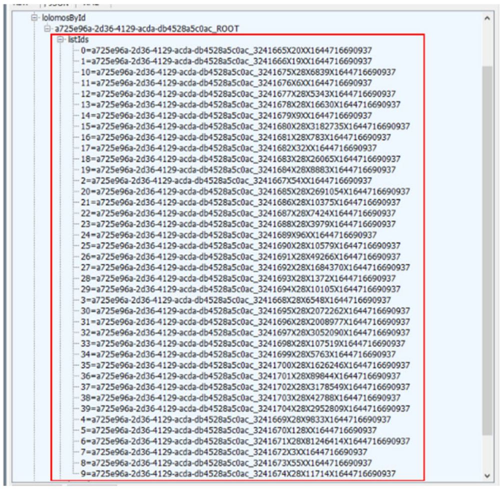
QueryString

Name	Value
appInternalVersion	13.38.0
appVersion	13.38.0
clientAppCanHandleFloatin	true
clientAppMaxRows	20
clientAppMaxTitlesPerRow	75
clientAppViewPortWidth	3
config	{"kidsTopTenListEnabled":"true","seasonRenewalPostPlayEnabled":"true","syno
device_type	NFAPPL-02-
esn	NFAPPL-02-IPHONE8=1-540D4F0040F36ED4DB7F00768D66C03B30276183DA3
fragmentLolomoLimitX	11
fragmentLolomoLimitY	5
fragmentLolomoOffsetY	13
hydrationLevel	fragment
idiom	phone
iosVersion	13.6.1
isTablet	false
isTop10KidsEnabled	true
languages	en-DE
locale	en-DE
lolomoId	3e952571-dd2e-4168-87c8-db770d6ee877_ROOT
maxDeviceWidth	375
method	get
model	saget
modelType	IPHONE8-1
odpAware	true
pathFormat	graph
pixelDensity	2.0
progressive	false
responseFormat	json
type	lolomo-ath
warmerHasCategories	1

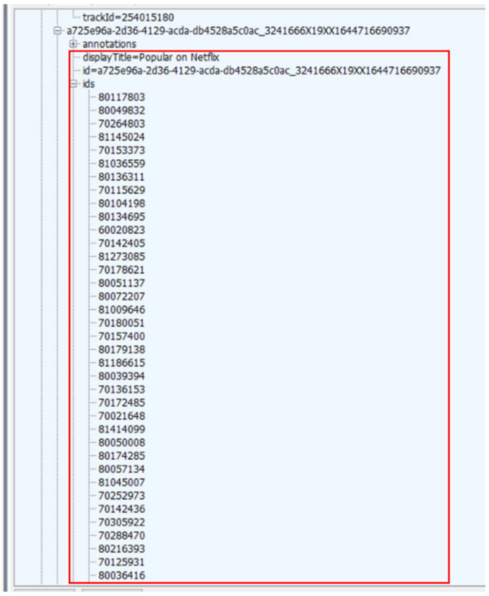
156. In response to the request, Netflix selects a portion of its product catalog to present to the subscriber, based upon the device capabilities. Specifically, Netflix returns a JSON payload to the requesting client device that includes category and content item listings for a subset of its product catalog that match the reported device capabilities (i.e., 40 rows of content categories and a maximum of 75 content items selected for each category row for an iPad tablet):



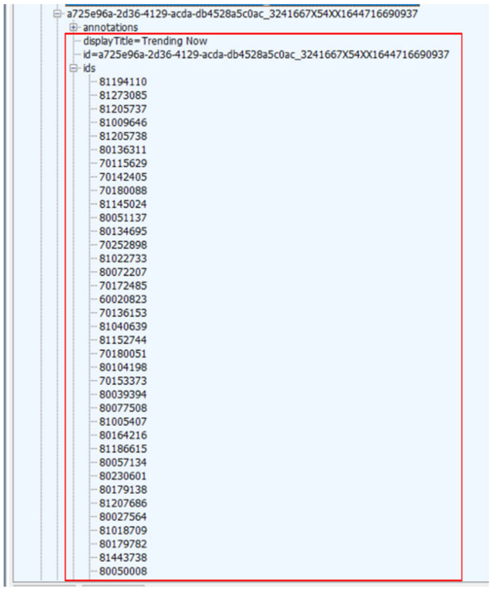
```
JSON
  aroCategoriesById
  aroCategoryIds
  entities
  listsById
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241665X20XX1644716690937
      annotations
      displayTitle=Spotlight
      id=a725e96a-2d36-4129-acda-db4528a5c0ac_3241665X20XX1644716690937
      ids
      index=0
      listContext=billboard
      requestId=35EA9828-E8BA-4E37-98B0-CB067A2FA4F7
      trackId=254015180
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241666X19XX1644716690937
      annotations
      displayTitle=Popular on Netflix
      id=a725e96a-2d36-4129-acda-db4528a5c0ac_3241666X19XX1644716690937
      ids
      index=1
      listContext=popularTitles
      requestId=35EA9828-E8BA-4E37-98B0-CB067A2FA4F7
      trackId=14170035
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241667X54XX1644716690937
      annotations
      displayTitle=Trending Now
      id=a725e96a-2d36-4129-acda-db4528a5c0ac_3241667X54XX1644716690937
      ids
      index=2
      listContext=trendingNow
      requestId=35EA9828-E8BA-4E37-98B0-CB067A2FA4F7
      trackId=15035895
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241668X28X6548X1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241669X28X9833X1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241670X128XX1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241671X28X81246414X1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241672X3XX1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241673X55XX1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241674X28X11714X1644716690937
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241675X28X6839X1644716690937
```



```
lolomosById
  a725e96a-2d36-4129-acda-db4528a5c0ac_ROOT
    listIds
      0=a725e96a-2d36-4129-acda-db4528a5c0ac_3241665X20XX1644716690937
      1=a725e96a-2d36-4129-acda-db4528a5c0ac_3241666X19XX1644716690937
      10=a725e96a-2d36-4129-acda-db4528a5c0ac_3241675X28X6839X1644716690937
      11=a725e96a-2d36-4129-acda-db4528a5c0ac_3241676X6XX1644716690937
      12=a725e96a-2d36-4129-acda-db4528a5c0ac_3241677X28X5343X1644716690937
      13=a725e96a-2d36-4129-acda-db4528a5c0ac_3241678X28X16630X1644716690937
      14=a725e96a-2d36-4129-acda-db4528a5c0ac_3241679X9XX1644716690937
      15=a725e96a-2d36-4129-acda-db4528a5c0ac_3241680X28X3182735X1644716690937
      16=a725e96a-2d36-4129-acda-db4528a5c0ac_3241681X28X783X1644716690937
      17=a725e96a-2d36-4129-acda-db4528a5c0ac_3241682X32XX1644716690937
      18=a725e96a-2d36-4129-acda-db4528a5c0ac_3241683X28X26065X1644716690937
      19=a725e96a-2d36-4129-acda-db4528a5c0ac_3241684X28X8883X1644716690937
      2=a725e96a-2d36-4129-acda-db4528a5c0ac_3241667X54XX1644716690937
      20=a725e96a-2d36-4129-acda-db4528a5c0ac_3241685X28X2691054X1644716690937
      21=a725e96a-2d36-4129-acda-db4528a5c0ac_3241686X28X10375X1644716690937
      22=a725e96a-2d36-4129-acda-db4528a5c0ac_3241687X28X7424X1644716690937
      23=a725e96a-2d36-4129-acda-db4528a5c0ac_3241688X28X3979X1644716690937
      24=a725e96a-2d36-4129-acda-db4528a5c0ac_3241689X96XX1644716690937
      25=a725e96a-2d36-4129-acda-db4528a5c0ac_3241690X28X10579X1644716690937
      26=a725e96a-2d36-4129-acda-db4528a5c0ac_3241691X28X49266X1644716690937
      27=a725e96a-2d36-4129-acda-db4528a5c0ac_3241692X28X1684370X1644716690937
      28=a725e96a-2d36-4129-acda-db4528a5c0ac_3241693X28X1372X1644716690937
      29=a725e96a-2d36-4129-acda-db4528a5c0ac_3241694X28X10105X1644716690937
      3=a725e96a-2d36-4129-acda-db4528a5c0ac_3241668X28X6548X1644716690937
      30=a725e96a-2d36-4129-acda-db4528a5c0ac_3241695X28X2072262X1644716690937
      31=a725e96a-2d36-4129-acda-db4528a5c0ac_3241696X28X2008977X1644716690937
      32=a725e96a-2d36-4129-acda-db4528a5c0ac_3241697X28X3052090X1644716690937
      33=a725e96a-2d36-4129-acda-db4528a5c0ac_3241698X28X107519X1644716690937
      34=a725e96a-2d36-4129-acda-db4528a5c0ac_3241699X28X5763X1644716690937
      35=a725e96a-2d36-4129-acda-db4528a5c0ac_3241700X28X1626246X1644716690937
      36=a725e96a-2d36-4129-acda-db4528a5c0ac_3241701X28X89844X1644716690937
      37=a725e96a-2d36-4129-acda-db4528a5c0ac_3241702X28X3178549X1644716690937
      38=a725e96a-2d36-4129-acda-db4528a5c0ac_3241703X28X42788X1644716690937
      39=a725e96a-2d36-4129-acda-db4528a5c0ac_3241704X28X2952809X1644716690937
      4=a725e96a-2d36-4129-acda-db4528a5c0ac_3241669X28X9833X1644716690937
      5=a725e96a-2d36-4129-acda-db4528a5c0ac_3241670X128XX1644716690937
      6=a725e96a-2d36-4129-acda-db4528a5c0ac_3241671X28X81246414X1644716690937
      7=a725e96a-2d36-4129-acda-db4528a5c0ac_3241672X3XX1644716690937
      8=a725e96a-2d36-4129-acda-db4528a5c0ac_3241673X55XX1644716690937
      9=a725e96a-2d36-4129-acda-db4528a5c0ac_3241674X28X11714X1644716690937
```



```
      trackId=254015180
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241666X19XX1644716690937
      annotations
      displayTitle=Popular on Netflix
      id=a725e96a-2d36-4129-acda-db4528a5c0ac_3241666X19XX1644716690937
      ids
        80117803
        80049832
        70264803
        81145024
        70153373
        81036559
        80136311
        70115629
        80104198
        80134695
        60020823
        70142405
        81273085
        70178621
        80051137
        80072207
        81009646
        70180051
        70157400
        80179138
        81186615
        80039394
        70136153
        70172485
        70021648
        81414099
        80050008
        80174285
        80057134
        81045007
        70252973
        70142436
        70305922
        70288470
        80216393
        70125931
        80036416
```



```
    a725e96a-2d36-4129-acda-db4528a5c0ac_3241667X54XX1644716690937
      annotations
      displayTitle=Trending Now
      id=a725e96a-2d36-4129-acda-db4528a5c0ac_3241667X54XX1644716690937
      ids
        81194110
        81273085
        81205737
        81009646
        81205738
        80136311
        70115629
        70142405
        70180088
        81145024
        80051137
        80134695
        70252898
        81022733
        80072207
        70172485
        60020823
        70136153
        81040639
        81152744
        70180051
        80104198
        70153373
        80039394
        80077508
        81005407
        80164216
        81186615
        80057134
        80230601
        80179138
        81207686
        80027564
        81018709
        80179782
        81443738
        80050008
```

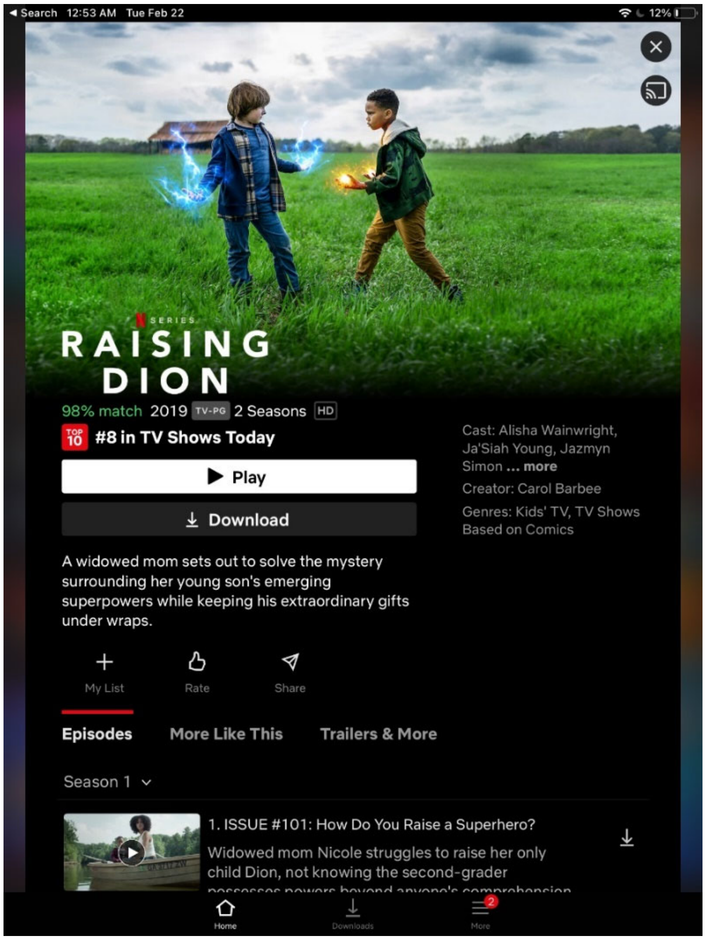
157. Netflix presents the selected portion of the product catalog to the subscriber during

a streaming session as a function of its content personalization microservice.[67] In particular, according to Netflix, each device has different hardware capabilities that can limit the number of videos or rows displayed at any one time and how big the whole page can be. As such, the page generation process must be aware of the constraints of the device for which it is creating the page, including the number of rows, the minimum and maximum length of a row, the size of the visible portion of the page, and whether or not certain rows are required or are not applicable to a certain device.[68]

158. The portion of the product catalog as presented provides only a single description of each item of digital content, regardless of the number of implementations of each item. For example, although each digital program in the selected portion of the product catalog has multiple associated different bitrate and format encodes, as discussed previously, in presenting the selected portion of the product catalog to the Netflix subscriber, Netflix provides only a single description of each content program to the subscriber.

[67] *See* Ex. 63, "Learning a Personalized Homepage", NETFLIX, https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a (last retrieved, Feb. 21, 2022).

[68] *Id.*



Cast: Alisha Wainwright, Ja'Siah Young, Jazmyn Simon ... more

Creator: Carol Barbee

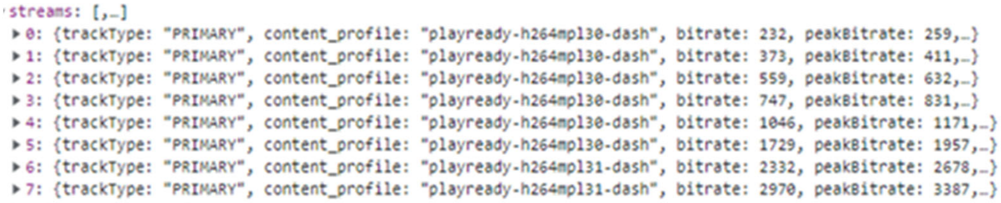
Genres: Kids' TV, TV Shows Based on Comics

A widowed mom sets out to solve the mystery surrounding her young son's emerging superpowers while keeping his extraordinary gifts under wraps.

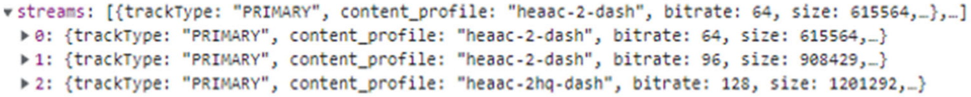
1. ISSUE #101: How Do You Raise a Superhero?

Widowed mom Nicole struggles to raise her only child Dion, not knowing the second-grader possesses powers beyond anyone's comprehension

In contrast, the manifest listing for the program Raising Dion, for example, indicates at least eight different video encoding implementations (first image below) and three different audio encoding implementations (second image below) for the content program:



```
streams: [,…]
 ▸0: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 232, peakBitrate: 259,…}
 ▸1: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 373, peakBitrate: 411,…}
 ▸2: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 559, peakBitrate: 632,…}
 ▸3: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 747, peakBitrate: 831,…}
 ▸4: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 1046, peakBitrate: 1171,…}
 ▸5: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 1729, peakBitrate: 1957,…}
 ▸6: {trackType: "PRIMARY", content_profile: "playready-h264mpl31-dash", bitrate: 2332, peakBitrate: 2678,…}
 ▸7: {trackType: "PRIMARY", content_profile: "playready-h264mpl31-dash", bitrate: 2970, peakBitrate: 3387,…}
```



```
▾streams: [{trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 615564,…},…]
  ▸0: {trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 615564,…}
  ▸1: {trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 96, size: 908429,…}
  ▸2: {trackType: "PRIMARY", content_profile: "heaac-2hq-dash", bitrate: 128, size: 1201292,…}
```

159. On information and belief, to the extent applicable, VideoLabs has complied with 35 U.S.C. § 287(a) with respect to the ’790 Patent.

160. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.[69] Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.[70] Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory call on March 19, 2021.[71] Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.[72] After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a "standstill" for an indefinite period of time.[73] Given that nearly two years had already passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs' communications. On February 22, 2022, VideoLabs sent an email to Netflix apprising it of the '790 Patent, identifying the Netflix products that infringe the '790 Patent, and requesting that the parties discuss the terms for Netflix to take a license to the '790 Patent.[74]

161. Netflix of course knows how its products operate, and on information and belief, upon receiving notice of the '790 Patent, has begun investigating the '790 Patent and its infringement of the Netflix '790 Accused Products. Netflix has been given further notice of the '790 Patent and its infringement of the '790 Patent through the filing of this Complaint. On information and belief, Netflix is either knowingly infringing the '790 Patent or is willfully blind to its infringement — including by ignoring VideoLabs' communications and/or using a

[69] *See* Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

[70] *See* Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

[71] *See* Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

[72] *See id.* at 8-10.

[73] *See id.* at 2-7.

[74] *See id.* at 1-2.

disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs and learning more about VideoLabs' patents — and continues to act in wanton disregard of VideoLabs' patent rights.

162. Despite becoming aware of or willfully blinding itself to its infringement of the '790 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '790 Accused Products. On information and belief, Netflix has made no attempts to design around the '790 Patent or otherwise stop its infringing behavior.

163. Netflix's infringement of the '790 Patent therefore has been and remains willful.

164. Netflix also indirectly infringes the '790 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '790 Accused Products in this District and throughout the United States. As described above, on information and belief, Netflix has known about the '790 Patent since at least February 22, 2022.

165. On information and belief, Netflix has actively induced the infringement of the '790 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix '790 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the '790 Patent in violation of 35 U.S.C. § 271(a).

166. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the '790 Patent by, for example, marketing the Netflix '790 Accused Products and infringing functionalities to consumers; working with consumers to implement, install and/or operate the Netflix '790 Accused Products and infringing functionalities; fully supporting and managing consumers' continuing use of the Netflix '790 Accused Products and infringing

functionalities; and providing technical assistance to consumers during their continued use of the '790 Accused Products and infringing functionalities.[75]

167. For example, Netflix induces third parties to infringe the '790 Patent by encouraging them to install and then operate the Netflix streaming service, which constitutes infringement of the '790 Patent. Netflix advertises and promotes its Netflix streaming service on its website and in various app stores such as Apple's app store and Android's app store in connection with the Netflix mobile application that can be installed on consumers' respective connected iOS and Android devices (as well as others), and encourages consumers to configure and then operate their mobile and computer devices in an infringing manner.[76] In response, consumers acquire, configure, and then operate the Netflix streaming service in an infringing manner.

168. Netflix further induces third parties to infringe the '790 Patent by encouraging and instructing Netflix content partners to provide their original content programming to be ingested into the Netflix service through the Netflix Backlot portal and/or the Netflix Content Hub.[77] In response, Netflix content partners follow Netflix guidelines and instructions to submit their raw mezzanine content to Netflix for ingestion into the Netflix service.

169. On information and belief, Netflix contributorily infringes the '790 Patent under 35 U.S.C. § 217(c) by importing, selling, and/or offering to sell within the United States the Netflix

[75] *See e.g.,* Ex. 54, *Netflix Help Center*, https://help.netflix.com/en/ (last accessed Feb. 15, 2022).

[76] *See e.g.,* Ex. 54, *Netflix Help Center*, https://help.netflix.com/en/ (last accessed Feb. 15, 2022).

[77] *See e.g.,* Ex. 57, *https://partnerhelp.netflixstudios.com/hc/en-us/articles/115004926427-Backlot-Overview-for-Content-Partners* (last retrieved, Feb. 21, 2022).

[77] *See e.g.,* Ex. 58, *https://partnerhelp.netflixstudios.com/hc/en-us/articles/360000509487-Content-Hub-Overview-Introduction* (last retrieved, Feb. 21, 2022).

’790 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the content management and delivery techniques used by Netflix to provide subscribers access to Netflix content on their respective connected devices is material, has no insubstantial non-infringing uses, and is known by Netflix to be especially made or adapted for use that practices at least claim 2 of the ’790 Patent with respect to the ’790 Accused Products.

Prayer For Relief

WHEREFORE, VideoLabs prays for judgment as follows:

a) That Netflix directly and/or indirectly infringes the ’878, ’559, and ’790 Patents;

b) That such infringement is willful;

c) That Netflix and its respective officers, directors, agents, partners, servants, employees, attorneys, licensees, successors, and assigns, and those in active concert or participation with any of them, be permanently enjoined from engaging in infringing activities with respect to the ’878, ’559, and ’790 Patents;

d) In the alternative, in the event injunctive relief is not granted as requested by VideoLabs, an award of a mandatory future royalty payable on each future product sold by Netflix that is found to infringe one or more claims of the ’878, ’559, and ’790 Patents, and on all future products which are not colorably different from products found to infringe;

e) That Netflix be required to pay VideoLabs damages in an amount adequate to compensate VideoLabs for Netflix’s infringement, but in no event less than a reasonable royalty under 35 U.S.C. § 284, including supplemental damages for any continuing post-verdict infringement up until entry of judgment and beyond, with accounting, as needed;

f) That VideoLabs be awarded all statutory and actual damages to which it is entitled, including the profits reaped by Netflix through its illegal conduct, and prejudgment and post-judgment interest;

g) That VideoLabs be awarded enhanced damages, up to and including trebling of the damages awarded to VideoLabs;

h) That VideoLabs be awarded recovery of the costs of this suit, including reasonable attorneys' fees; and

i) That VideoLabs be awarded such other and further relief as this Court deems just and proper.

Demand For Jury Trial

170. Pursuant to Rule 38 of the Federal Rules of Civil Procedure and D. Del. LR 38.1, VideoLabs demands a jury trial on its claims for patent infringement and any and all issues triable of right before a jury.

Dated: February 23, 2022

Christine E. Lehman (*pro hac vice*
forthcoming)
clehman@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
1710 Rhode Island Ave NW, 12th Floor
Washington, DC 20036
Telephone: (202) 894-7310
Fax: (650) 623-1449

Wesley L. White *(pro hac vice* forthcoming*)*
wwhite@reichmanjorgensen.com
Jaime F. Cardenas-Navia *(pro hac vice*
forthcoming*)*
jcardenas-navia@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
750 Third Avenue, Suite 2400
New York, NY 10017
Telephone: (212) 381-1965
Facsimile: (650) 623-1449

Respectfully submitted,

FARNAN LLP

/s/ Michael J. Farnan
Brian E. Farnan (Bar No. 4089)
Michael J. Farnan (Bar No. 5165)
919 N. Market St., 12th Floor
Wilmington, DE 19801
Telephone: (302) 777-0300
Fax: (302) 777-0301
bfarnan@farnanlaw.com
mfarnan@farnanlaw.com

Courtland L. Reichman (*pro hac vice*
forthcoming)
creichman@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
100 Marine Parkway, Suite 300
Redwood Shores, California 94065
Telephone: (650) 623-1401
Fax: (650) 623-1449

Taylor N. Mauze (*pro hac vice* forthcoming)
tmauze@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
7500 Rialto Blvd., Ste. 1-250
Austin, Texas 78735
Telephone: (650) 623-1401
Facsimile: (650) 623-1449

Attorneys for Plaintiffs VideoLabs, Inc. and VL Collective IP LLC