Roman M. Silberfeld, SBN 62783
RSilberfeld@RobinsKaplan.com
Daniel L. Allender, SBN 264651
DAllender@RobinsKaplan.com
ROBINS KAPLAN LLP
2049 Century Park East, Suite 3400
Los Angeles, CA 90067
Telephone: (310) 552-0130
Facsimile: (310) 229-5800

Christopher A. Seidl (*pro hac vice* to be filed)
CSeidl@RobinsKaplan.com
Aaron R. Fahrenkrog (*pro hac vice* to be filed)
AFahrenkrog@RobinsKaplan.com
Bryan J. Mechell (*pro hac vice* to be filed)
BMechell@RobinsKaplan.com
ROBINS KAPLAN LLP
800 LaSalle Avenue, Suite 2800
Minneapolis, MN 55402
Telephone: (612) 349-8500
Facsimile: (612) 339-4181

Attorneys for Plaintiff
DivX, LLC

ROBINS KAPLAN LLP
ATTORNEYS AT LAW
LOS ANGELES

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

DIVX, LLC, a Delaware limited liability company,

Plaintiff,

v.

NETFLIX, INC., a Delaware corporation,

Defendant.

Case No. 2:19-cv-1602

COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

Plaintiff DivX, LLC ("Plaintiff" or "DivX"), by its attorneys, for its complaint against Defendant Netflix, Inc. ("Defendant" or "Netflix") for patent infringement alleges as follows:

INTRODUCTION

1. Since 2000, DivX has been setting the standard for high-quality digital video. DivX® technology helps people around the world enjoy digital media on their own terms.

2. DivX is one of the first companies to enable successful delivery of high-quality digital video over the internet. For nearly 20 years, DivX has been developing innovative technology to enable quality digital entertainment experiences for consumers—making internet video high-quality, secure, easy, and enjoyable for consumers to watch on any device.

3. Today, consumers expect to play high-quality video from the internet on any device at the touch of a button. However, when DivX's engineers accomplished this feat almost 20 years ago, they had to overcome significant technical obstacles to do so. Through those efforts, DivX engineers invented foundational technologies that made high-quality internet video possible long before platforms like Netflix or Hulu existed.

4. DivX began by improving video compression technology that made it possible to transmit large video files over the internet. It created technology allowing those video files to be played on a wide variety of consumer electronics devices, and it licensed that technology to consumer electronics manufacturers. It next developed Digital Rights Management (DRM) technology, including encryption for video files, to protect valuable video content so that content producers would be comfortable making their original works available on the internet. Finally, building on all of these technologies, DivX launched Stage6, one of the first platforms for streaming high-quality, user-created and professional video

ROBINS KAPLAN LLP
ATTORNEYS AT LAW
LOS ANGELES

over the internet. All of this work paved the way (and provided a roadmap) for today's proliferation of internet video streaming on consumer devices.

5. As a result of the many DivX innovations relating to internet video and streaming media, consumer electronics (CE) companies have licensed DivX's technologies and integrated them into more than one billion devices worldwide.

6. DivX's investments in research and development for internet video led to technical innovations that shaped internet video as the world knows it today. DivX patented these inventions and today has a portfolio of over 300 issued patents and patent applications.

7. Today, Netflix is the world's most successful video streaming business, delivering streaming video over the internet to more than 150 million subscribers in countries around the world. Netflix's video streaming success depends upon the technical innovations developed and patented by DivX, including DivX technologies enabling:

- a streaming ecosystem of many consumer devices;
- efficient compression for high-quality video delivery and playback;
- efficient and effective DRM to protect video content from unauthorized use and copying; and
- video playback features that make internet video easier and more enjoyable for consumers to access.

Without these DivX innovations, Netflix would not enjoy the success that it does today.

8. DivX brings this lawsuit to seek fair compensation from Netflix for its unauthorized and unlicensed use of DivX's patented technology.

NATURE OF THE ACTION

9. This complaint alleges patent infringement. DivX alleges that Netflix has infringed and continues to infringe, directly and/or indirectly, eight DivX patents: U.S. Patent Nos. 7,295,673 (the "'673 patent"), 8,139,651 (the "'651 patent"), 8,472,792 (the "'792 patent"), 9,184,920 (the "'920 patent"), 9,270,720 (the "'720 patent"), 9,998,515 (the "'515 patent"), 10,212,486 (the "'486 patent"), and 10,225,588 (the "'588 patent"), copies of which are attached as Exhibits 1-8 (collectively, the "DivX Patents").

10. The DivX Patents cover foundational internet video streaming technologies for delivering secure digital video content to a variety of consumer electronic devices and enabling content viewing on those devices. These technologies are necessary for Netflix to deliver the viewing experience that its users expect and to obtain and distribute content for its streaming service. The DivX Patents disclose technologies that enable many benefits, including:

- receipt and playback of streaming digital video on a wide variety of consumer electronic devices;
- high-quality video playback, including 4K high-resolution video, without stalls;
- robust and efficient DRM; and
- features that improve user experience, including trick play and fast start.

11. Netflix directly infringes the DivX Patents by making, using, offering to sell, selling, and/or importing into the United States internet video streaming technology, software, and services that practice the inventions claimed in the DivX Patents.

12. Netflix indirectly infringes at least seven of the DivX Patents by inducing its consumer end-users to directly infringe those DivX Patents. Netflix induces infringement by providing software (the Netflix application) that, when

used by consumers or other content viewers to stream digital video to televisions, personal computers, phones, tablets, and other devices, as directed and intended by Netflix, causes those users to make, use, and practice the inventions claimed in the DivX Patents.

13. DivX seeks damages and other relief for Netflix's infringement of the DivX Patents.

THE PARTIES

14. DivX is a Delaware limited liability company. Its principal place of business is 4350 La Jolla Village Drive, Suite 950, San Diego, California, 92122. DivX owns patents covering foundational internet video streaming technologies, including those asserted here.

15. Netflix is a Delaware corporation.[1] Its principal place of business and global headquarters is at 100 Winchester Circle, Los Gatos, California, 95032.[2]

16. Upon information and belief, Netflix maintains an office in Los Angeles, California, that employs about 800 employees. According to Netflix's website, the Los Angeles office "is the entertainment hub for Netflix with teams such as Content, Legal, Marketing & Publicity and is located on the Sunset Bronson Studio Lot where a variety of Netflix content is created."[3]

17. Upon information and belief, Netflix is the global leader in streaming digital video content, which includes films, television series, and other video content. Upon information and belief, Netflix designs, operates, tests, manufactures, uses, offers for sale, sells, and/or imports into the United States—including into the

[1] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

[2] *Id.*

[3] https://jobs.netflix.com/locations/los-angeles-california.

Central District of California—internet video streaming software and services that generate billions of dollars of revenue for Netflix each year.

JURISDICTION AND VENUE

18. This is an action for patent infringement under the Patent Laws of the United States, 35 U.S.C. § 1 et seq., over which this Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

19. This Court has both general and specific jurisdiction over Netflix because Netflix has committed acts within the Central District of California giving rise to this action and has established minimum contacts with this forum such that the exercise of jurisdiction over Netflix would not offend traditional notions of fair play and substantial justice. Defendant Netflix, directly and through subsidiaries and intermediaries (including distributors, retailers, franchisees and others), has committed and continues to commit acts of patent infringement in this District, by, among other things, making, using, testing, selling, licensing, importing and/or offering for sale/license products and services that infringe the DivX Patents.

20. Venue is proper in this district and division under 28 U.S.C. §§ 1391(b)-(d) and 1400(b) because Netflix has committed acts of infringement in the Central District of California and has a regular and established physical place of business and employees in the Central District of California, in Los Angeles. At its Los Angeles facility, Netflix employs technical engineers in many disciplines, including cloud and platform engineering, information security, data engineering and infrastructure, product engineering, and data science and analytics.[4]

FACTUAL BACKGROUND

I. DivX

21. Established in 2000, DivX pioneered the delivery of high-quality digital video content to consumers over the internet. Among other things, DivX has

[4] *Id.*

invented technologies for video compression, transmission, playback, and security that enable distribution of high-quality video over the internet for playback on a wide variety of consumer devices.

22. DivX distributes consumer software implementing its technologies, and licenses its software to CE manufacturers. DivX has licensed and integrated its software into more than one billion consumer electronic devices. Consumers have downloaded DivX's software more than one billion times. DivX continues to invest in research and development for internet video streaming innovations today.

A. DivX's Origin

23. In 1999, Jérôme Rota, a compositing infographist and video engineer, wanted to compress digital video files in order to be able to share them over the internet.

24. Frustrated with the restrictions and limitations of existing digital video technologies, Mr. Rota created code enabling the MPEG-4 (Moving Picture Experts Group Phase 4 Standard) video codec to be used in a more open way. Mr. Rota modified the MPEG-4 codec for use outside of Windows Media Player (.asf-restricted implementation), enabling it in .avi (audio video interleaved) formatted files.

25. Mr. Rota distributed this code for free online, using the moniker "DivX ;-)"—a play on the now-defunct Circuit City's Digital Video Express DVD service. The DivX ;-) code proved popular and soon became synonymous with how to compress digital video content.

26. Around that same time, Jordan Greenhall, a former Mp3.com executive, learned of the DivX ;-) code. He wanted to create a company around this disruptive technology and be the first to market technology that enabled the efficient transfer and distribution of high-quality digital video content over the internet.

27. Mr. Greenhall contacted Mr. Rota in March 2000 and the two began to build a team of software engineers. Around September 2000, Mr. Greenhall and others co-founded DivXNetworks, Inc., the predecessor business of plaintiff DivX, LLC.

28. DivX's initial goal was to build an internet video solution—or, perhaps more accurately, an internet video revolution. It identified two ways to achieve its early goal: (1) distribute software, including a video codec, to consumers to make it easier to use and share video with each other over the internet; and (2) create a system for video delivery over the internet from a server to multiple users, later called the DivX Open Video System (OVS). Thus, DivX set off to create a mechanism for encoding digital video content for easy distribution via the internet.

B. The DivX Software

29. DivX recognized that consumers wanted *accessible*, *high-quality* digital video content. To satisfy this demand, DivX created a new implementation of the MPEG-4 video standard. In 2001, after starting from scratch, DivX released the DivX Codec 4.0 to replace the earlier DivX ;-) code. A "codec" is a computer program for encoding—that is, compressing—and decoding digital video files. Over the next decade, DivX released numerous new versions of the DivX Codec (collectively, the "DivX Software").

30. The DivX Software functioned like a master translator for digital video files, allowing for variations in codecs, containers, and playback across different file types on different devices. It allowed consumers to compress, decode, and play back digital video using a standard program.

31. DivX offered its DivX Software for free. At the same time, access to and use of digital video became more widespread as computing power increased. These factors led to widespread adoption of the DivX Software and a large base of DivX users.

32. The DivX Software, in its latest form, combines the DivX Codec, video player, and video converter into what is known as the DivX "Consumer Bundle." DivX offers the Consumer Bundle for free to allow consumers to continue to enjoy high-quality video playback (via the DivX Player), to convert video (via the DivX Converter), and to cast media from a computer to a TV (via the DivX Media Server). DivX also sells a "DivX Pro" version of the DivX Software, which includes additional advanced features.

C. The DivX OVS

33. Around November 2000, Mr. Greenhall hired Eric Grab to lead a team of engineers focused on building an online video consumer service and application called the Open Video System ("OVS") that would allow protected digital video content to travel over the internet.

34. Mr. Grab is a named inventor on the ’673, ’920, and ’588 patents.

35. In 2001, DivX launched the DivX OVS, which could ingest, store, protect, transmit, and authenticate secure digital video content. Consumers could access content using DivX OVS, through the DivX Player.

36. The DivX OVS was one of the world’s first MPEG-4 full-screen internet video playback systems with state-of-the-art compression capabilities. It allowed the first DVD-like quality digital video content to securely travel over the internet.

37. The DivX OVS enabled companies possessing video content, such as studios—the content holders—to allow consumers to download and play back videos using the DivX OVS. DivX allowed content holders and distributors to build internet video websites using DivX Software to support the backend system and video playback.

D. Meeting Competing Needs: The DivX Internet Video Ecosystem and the DivX DRM

38. As the DivX Software and the DivX OVS gained popularity in the market, DivX's continued growth depended on its ability to balance competing needs among (1) content holders, (2) CE manufacturers, and (3) consumers. Content holders demanded better security, CE manufacturers demanded better performance, and consumers demanded greater accessibility and improved user experience—in particular, the ability to watch video on devices other than personal computers, such as televisions (and later, smartphones and tablets).

39. Content holders (including studios) demanded additional content protection before agreeing to license the DivX OVS. To put the studios at ease, DivX invested substantial resources in developing state-of-the-art content protection technology. From 2000 to 2005, DivX met with many studios about content distribution, including Disney, Warner Bros., Sony, Universal, and Paramount Pictures.

40. DivX created a system, with input from the studios and CE manufacturers, that met the studios' needs for security and solved the problems associated with internet delivery of secure studio content to CE devices and personal computers (PCs). The DivX DRM technology evolved to solve these problems. The DivX DRM established an elegant system that allowed each content holder to authorize playback of its content on multiple manufacturers' devices.

41. DivX's role in operating the DRM allowed DivX to focus on quality, standardization, and optimization.

42. Leading content distributors responded to DivX's technology. Throughout the mid-2000s, DivX was approached by several companies to discuss using DivX's technology to power online video content delivery platforms. Those companies included Blockbuster, Netflix, Amazon, and others.

E. DivX's Stage6 Platform

43. In 2006, DivX launched Stage6, an internet streaming platform and HTTP-based website for high-resolution user-generated and professional video. The platform incorporated DivX's proprietary technologies. This type of platform went on to become the core of adaptive bitrate streaming (ABS) systems.

44. Stage6 provided internet video users with a higher-resolution alternative to platforms like YouTube. Upon information and belief, at that time Stage6 was the only platform supporting high-resolution video. It allowed users to upload, share, and view high-resolution videos with DivX's Software. Stage6 allowed for uploading of much larger video files than platforms like YouTube; therefore, users could upload and share much larger video files. DivX made significant investments in bandwidth to facilitate this user experience.

45. Even in 2007, Stage6 supported streaming of 720p and 1080p high-definition video. The quality of the high-resolution video playback on Stage6 surprised reviewers, with one commenting "DivX has clearly got something right with web playback of higher-resolution video!"[5]

46. Stage6 enjoyed rapid user traffic growth, and by January 2008, it had over 10,000,000 monthly views. However, increased traffic resulted in increased bandwidth costs to DivX; DivX shut down Stage6 in February 2008 to focus its human resources and capital on the core DivX businesses.

F. DivX's CE Software & Certification Program

47. Beginning around 2002, CE manufacturers began receiving requests from consumers to implement functionality to enable playback of DivX video files.

48. CE manufacturers reached out to DivX to discuss OVS technology implementation in CE devices.

[5] *DivX Stage6 (beta)—the high-def rival to YouTube*, Hexus.net, May 1, 2007.

49. To meet CE manufacturers' needs—driven by consumer demand—DivX created a CE software development kit ("SDK") that would allow DVD players and other media players to play DivX files (on CD, DVD, USB, or network) while incorporating a secure DRM protocol.

50. DivX began testing CE devices to determine whether they could successfully use the DivX SDK to play DivX files. This testing matured into the DivX Certification Program. DivX developed Certification Test Kits ("CTKs") for CE manufacturers to certify their licensed devices.

51. DivX Certification was valuable to CE manufacturers, who could use the certification to demonstrate to consumers that their devices could play DivX files as well as a broad range of other video files. DivX also ensured that its video files would play on a wide range of devices by requiring its CE SDK licensees, also known as DivX Partners, to certify their devices using the CTKs.

52. DivX licensed its technology in the DivX SDKs through various DivX Profiles, including DivX Home Theater, DivX HD, DivX Plus HD, DivX HEVC Ultra HD, DivX Plus Streaming, DivX Mobile, and DivX Mobile Theater.

53. DivX has integrated its technology into more than one billion consumer electronic devices via the DivX SDKs.

54. To this day, DivX has numerous CE licensees for its SDKs and CTKs, including leading digital television, smartphone, in-car video device, DVD / Blu-ray disc, integrated circuit (IC), and original equipment manufacturers. DivX continues to invest in research and development to innovate in the area of video technology.

55. The DivX innovations relating to compression, playback, trick play, fast start, security, high quality, and easy access made video delivery to consumer electronics devices over the internet possible and is the foundation of streaming technology today.

G. Industry Interest in DivX's Technologies

56. DivX's internet video technologies attracted the interest of many companies, including Netflix, interested in launching video streaming services. In fact, Netflix expressed early interest in DivX's technologies. Before Netflix launched its video streaming business, DivXNetworks (the DivX predecessor business) and Netflix engaged in discussions relating to DivX's technologies. DivX and Netflix discussed whether Netflix would license or purchase technology from DivX, but ultimately did not reach any agreement through their discussions.

II. Netflix

57. Today, Netflix claims that it is "the world's leading internet television network with over 117 million streaming memberships in over 190 countries enjoying more than 140 million hours of TV shows and movies per day, including original series, documentaries and feature films."[6] It claims to be "a pioneer in the internet delivery of TV shows and movies."[7] Since the launch of its streaming service, Netflix has "developed an ecosystem for internet-connected screens and [has] added increasing amounts of content that enable consumers to enjoy TV shows and movies directly on their internet-connected screens."[8]

58. Netflix began in 1997 as a DVD-by-mail service.[9] In 2007, Netflix launched its streaming video platform.[10]

59. In 2007, Reed Hastings, Netflix CEO, stated, "We named our

[6] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

[7] *Id.*

[8] *Id.*

[9] *Id.*

[10] *Id.*

company Netflix in 1998 because we believed internet-based movie rental represented the future, first as a means of improving service and selection, and then as a means of movie delivery." "While mainstream consumer adoption of online movie watching will take a number of years due to content and technology hurdles, the time is right for Netflix to take the first step."[11]

60. Netflix strives to deliver an ecosystem that is easy to use and supports many devices. For example, Netflix touts that it enables members to "watch anywhere, anytime, on thousands of devices."[12] Further, "Netflix streaming software allows you to instantly watch content from Netflix through any internet-connected device that offers the Netflix app, including smart TVs, game consoles, streaming media players, set-top boxes, smartphones, and tablets."[13]

61. The Netflix streaming ecosystem includes numerous playback devices and operating systems. Netflix operates this ecosystem by hosting video content on servers, and distributing that content to many diverse devices through its distribution network. Users can access and play back video content on their devices by using the Netflix application. [14]

[11] https://www.zdnet.com/article/netflix-watch-movies-on-your-pc/.

[12] https://help.netflix.com/en/node/412; Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

[13] https://help.netflix.com/en/node/412.

[14] https://www.slideshare.net/yunongx/going-faaster-functions-as-a-service-at-netflix?qid=f0f8ab80-cc1a-4ef4-a884-b55dd8dc213e&v=&b=&from_search=10; https://help.netflix.com/en/node/101653.

62. Netflix actively encourages the installation and use of its application and service on consumer devices. Netflix has successfully pursued agreements with cable, satellite, and telecommunications operators to make Netflix's service available through television set-top boxes.[15] Netflix also has entered agreements with other consumer electronics device manufacturers to make Netflix's service available on those consumer devices.[16] Those products include streaming media players, smart TVs, game consoles, Blu-ray players, smartphones and tablets, and personal computers.[17] Netflix further recommends, directly to consumers, certain consumer electronics devices preloaded with Netflix.[18]

63. Netflix employs storage, transcoding, and distribution techniques to

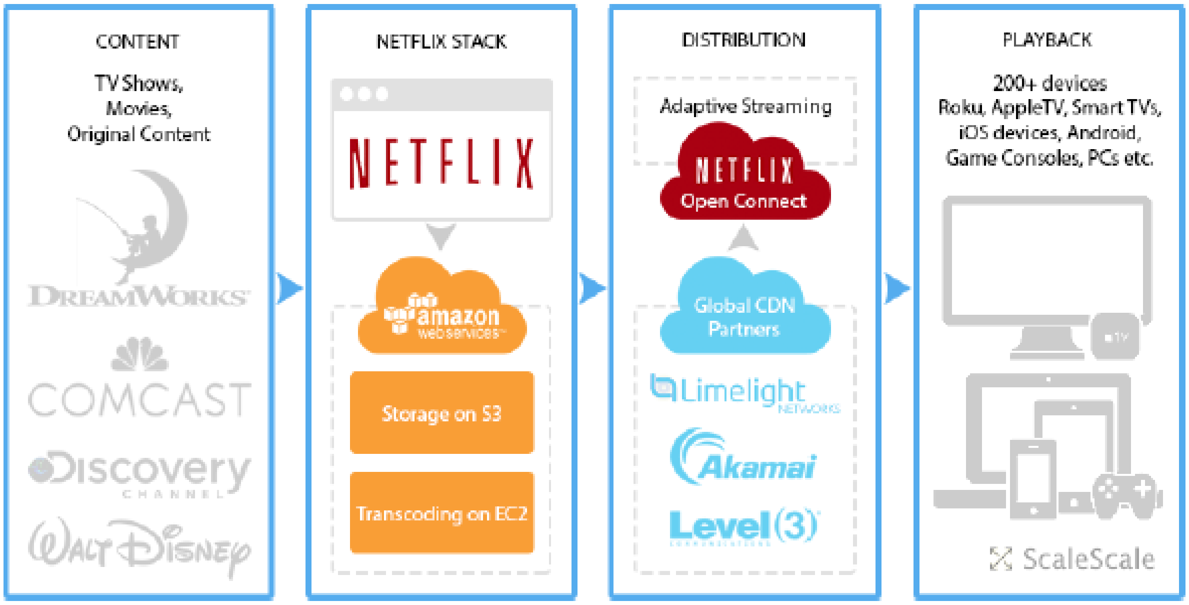
[15] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

[16] https://devices.netflix.com/en/.

[17] *Id.*

[18] https://devices.netflix.com/en/recommendedtv/2018/.

optimize delivery of content at maximum quality and speed.[19]



ScaleScale/scalescale.com

64. Netflix claims that it provides efficient compression for high-quality video and continuous streaming. Netflix aims "to serve your favorite shows and movies at the best possible quality."[20] It claims to do this by using "the video encoding technology" "to transform our video content into compressed bitstreams."[21] According to Netflix, it is "regularly evaluating the performance of existing and upcoming video codecs and encoders. [It] select[s] the freshest and best encoding technologies so that you can savor our content."[22]

65. Netflix has expanded its services to many countries beyond the United States, including to countries with slower wired and wireless networks. The

[19] https://medium.com/refraction-tech-everything/how-netflix-works-the-hugely-simplified-complex-stuff-that-happens-every-time-you-hit-play-3a40c9be254b.

[20] https://medium.com/netflix-techblog/performance-comparison-of-video-coding-standards-an-adaptive-streaming-perspective-d45d0183ca95.

[21] *Id.*

[22] *Id.*

bandwidth restrictions of these networks require Netflix to provide efficient video compression to deliver its service without "buffering." CEO Reed Hastings explained how Netflix wants to address this issue: "[s]ome of you are old enough to remember dial-up internet . . . now that seems like such a relic. Well, that's what we want to make buffering . . . . We're investing very heavily at many levels, on the network servers, on the interconnects with different [internet service providers] around the world, on the [video encoding] side so that the experience on mobile, on laptop, on the TV is just instant, there's no delay and then that really changes your relationship with the service."[23]

66. Additionally, concerned with data caps (restrictions imposed by internet service providers on the transfer of data over their networks), Netflix mitigates the potential trouble from data caps with encoding technology: "What we've done is invest in the codex [sic], the video encoders, so that at a half a megabit, you get incredible picture quality on a four and five-inch screen. Now, we're down in some cases to 300 kilobits and we're hoping someday to be able to get to 200 kilobits for an amazing picture. So we're getting more and more efficient at using operators' bandwidth."[24]

67. Netflix operates encoding servers and a content delivery network in the United States.[25]

68. Netflix touts that an advantage of its technology is adaptive bitrate streaming, which allows dynamic switching among video streams of different

[23] https://www.fool.com/investing/2017/03/18/how-netflix-addresses-its-toughest-challenges.aspx.

[24] *Id.*

[25] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

qualities if bandwidth or performance capabilities change during playback.[26]

69. Netflix relies upon DRM software for authorizing the playback of copyrighted material.[27] Indeed, Netflix has said that it depends upon DRM technology to satisfy the requirements of both Netflix's content suppliers and its device partners.[28]

70. Netflix depends on the ability to obtain rights to and produce video content that users want to watch.[29] It explains that "[w]e are continuously improving our members' experience by expanding our streaming content with a focus on a programming mix of content that delights our members."[30] Netflix competes for this content against both other video providers and other content providers.[31]

71. Netflix's success depends on differentiating its service from other entertainment sources by offering superior technology and superior content.[32]

[26] https://medium.com/netflix-techblog/performance-comparison-of-video-coding-standards-an-adaptive-streaming-perspective-d45d0183ca95; https://medium.com/netflix-techblog/optimized-shot-based-encodes-now-streaming-4b9464204830; https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f; https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming.

[27] https://help.netflix.com/en/node/395.

[28] https://news.microsoft.com/2010/05/25/netflix-taps-microsoft-playready-as-its-primary-drm-technology-for-netflix-ready-devices-and-applications/.

[29] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

[30] *Id.*

[31] *Id.*

[32] *Id.*

THE DIVX PATENTS

72. DivX solely owns all rights, titles, and interests in and to the DivX Patents, each described below.

I. The ’673 Patent

73. The ’673 patent, entitled “Method and System for Securing Compressed Digital Video,” was duly and legally issued on November 13, 2007, from a patent application filed July 8, 2003, with Eric W. Grab and Adam H. Li as the named inventors. The ’673 patent claims priority to U.S. Provisional Application No. 60/420,500, filed on October 23, 2002.

74. The inventions recited in the ’673 patent enable Netflix to stream video to a diverse array of consumer devices while protecting the video content with secure encryption and decryption, allowing Netflix to both offer its service on a diverse device ecosystem and provide high-quality video content. Specifically, the ’673 patent is directed to a partial frame encryption architecture that enables improved, more efficient streaming of encrypted video to any device, providing secure decryption without decoding.

75. The ’673 patent addresses a technical problem. Digital video files can be very large and therefore difficult to transmit over networks. Compressing those files “reduce[s] the bandwidth required to transmit digital video.” *See, e.g.*, ’673 patent, 1:46-49. But there is a tradeoff—modern compression and decompression techniques require a significant amount of processing power. *See, e.g.*, *id.* at 1:63-2:9. Video files must also be secure to protect the content, which requires encrypting and decrypting the files—further increasing the processing power needed to play back video and increasing the cost and complexity of the playback device. The ’673 patent explains that a need existed for providing adequate security while also limiting the resources consumed during video decryption. *See, e.g.*, *id.* at 3:39-50.

76. The ’673 patent claims specific ways to solve this problem with secure compressed digital video content that requires less processing power to decrypt—specifically, selective partial frame encryption. The ’673 patent describes creating a set of encrypted frames “by encrypting selected parts of selected frames of the sequence of frames in accordance with a frame encryption function” and generating “frame decryption information necessary to decrypt the set of encrypted frames.” *See, e.g., id.* at 3:59-63; *see also* 3:66-4:2; 5:25-32.

77. In particular, claim 1 of the ’673 patent recites generating, storing, and using a frame encryption key and generating frame decryption information to produce a protected stream of compressed video content. ’673 patent, claim 1. Claim 1 recites a novel solution of synchronizing decryption information with encrypted frames for frame-based encryption to provide secure digital video while reducing processing resources consumed during decryption in a manner that was not well-understood, routine, or conventional at the time of the ’673 patent. *Id.*

II. The ’651 Patent

78. The ’651 patent, entitled “Video Deblocking Filter,” was duly and legally issued on March 20, 2012, from a patent application filed May 26, 2010, with Cheng Huang as the named inventor. The ’651 patent claims priority to U.S. Provisional Application No. 60/611,513, filed on September 20, 2005.

79. The inventions recited in the ’651 patent allow Netflix’s users to stream high-resolution 4K content with smooth playback and without flaws in the video. Specifically, the ’651 patent is directed to a multidimensional adaptive deblocking filter that allows for more efficient compression, creating a higher-quality 4K streaming video experience.

80. The ’651 patent addresses a technical problem. Compressing digital video to make it smaller comes with the downside of potentially losing visual information and degrading the quality of playback. *See, e.g.*, ’651 patent, 1:27-34. To overcome this problem, a “deblocking filter” can be used by the computing

system when reconstructing compressed digital video to produce better image quality. *See, e.g.*, *id.* at 1:29-34. However, multiple types of deblocking filters existed, and if the wrong one was applied by the computing system, the process could make the image quality worse, not better. *See, e.g.*, *id.* at 1:48-49; 1:60-63; 1:67-2:3. Selecting the appropriate filter to apply to a given video frame, therefore, is critical. Accordingly, a need existed for an improved method of accurately and efficiently selecting the appropriate deblocking filter to apply based on the digital video data itself.

81. The ’651 patent claims specific ways of solving this problem with improved methods of inspecting a boundary of a block of video data in a frame using a computing system and selecting the appropriate deblocking filter to apply to produce the best visual result based on that data. *See, e.g.*, *id.* at 7:65-8:5; 8:38-43.

82. In particular, claim 1 of the ’651 patent recites identifying a block boundary, analyzing the boundary by inspecting pixels from multiple rows and multiple columns “that encompass pixels immediately adjacent to at least two sides of the block boundary and includes at least one pixel that is not immediately adjacent to the block boundary,” and selecting the appropriate filter. *Id.* at claim 1. Claim 1 recites a novel solution for more efficiently processing digital video data to improve the visual quality of the video in a manner that was not well-understood, routine, or conventional at the time of the ’651 patent.

III. The ’792 Patent

83. The ’792 patent, entitled “Multimedia Distribution System,” was duly and legally issued on June 25, 2013, from a patent application filed October 24, 2005, with Abou Ul Aala Ahsan, Steve R. Bramwell, and Brian T. Fudge as the named inventors.

84. The inventions recited in the ’792 patent enable playback features that video streaming users expect, enjoy, and use to navigate digital video easily, and they improve the user experience by reducing delays in loading and playing a video

when it is selected by the user. Specifically, the ’792 patent is directed to providing an abridged video index that improves the user playback experience by enabling chunk-based adaptive bitrate streaming, “trick play,” and “fast start” functionality.

85. The ’792 patent addresses a technical problem. Originally, multimedia, like video, was transmitted over the internet with a single index for all the content in the multimedia file. As internet multimedia became more sophisticated and complex, the size of this index and the computing resources needed to process it increased. The process of obtaining the index therefore was time- and resource-consuming and either delayed the start of video playback for the user or prevented the user from using desirable technical playback features like seeking, fast-forward, and rewind. Accordingly, a need existed for an improved multimedia file format and systems for generating, distributing, and decoding multimedia files with an improved index structure that can enable desirable playback features while reducing the computing resources, and associated delays, required to obtain and process the index.

86. The ’792 patent claims a solution to this problem with an improved, novel index structure that solves the technical problems and resource-intensive computing issues associated with complex video files. *See, e.g.*, ’792 patent, 15:10-16:36; 48:21-49:42. This technical solution enables desired video playback features like starting video immediately and the ability to fast forward, rewind, and skip scenes. *See, e.g.*, *id.* at 16:16-29; 48:21-37.

87. In particular, claim 9 of the ’792 patent recites an encoder for encoding a multimedia file that comprises a memory including “a full index” and a processor configured to generate an “abridged index,” enabling trick play functionality and improved playback within the video file. *Id.* at claim 9. Claim 9 recites a novel solution for more efficiently processing multimedia files to enable desirable playback features in a manner that was not well-understood, routine, or conventional at the time of the ’792 patent.

IV. The ’920 Patent

88. The ’920 patent, entitled “Federated Digital Rights Management Scheme Including Trusted Systems,” was duly and legally issued on November 10, 2015, from a patent application filed February 18, 2014, with Eric W. Grab, Chris Russell, Francis Yee-Dug Chan, and Michael George Kiefer as the named inventors. The ’920 patent claims priority to U.S. Provisional Application No. 60/782,215, filed on March 14, 2006.

89. The inventions recited in the ’920 patent allow Netflix to deliver video content securely to many different devices, supporting a large and diverse streaming device ecosystem. Specifically, the content security provided by the ’920 inventions also allows Netflix to obtain and offer its users a library of high-quality video content. The ’920 patent is directed to a DRM architecture that enhances content security by binding active encryption keys to a user, allowing secure streaming.

90. The ’920 patent addresses a technical problem. Digital content must be protected to make sure that only those people who have paid for it can access it. *See, e.g.*, ’920 patent, 1:25-29. This can be done by issuing “keys” to authorized users to unlock the content. *See, e.g.*, *id.* at 1:29-31. Those keys can be incorporated in devices that play back video, but content providers want to share their keys with as few others as possible—including the device manufacturers. *See, e.g.*, *id.* at 1:34-44. Accordingly, content providers needed a way to control access to digital content without involving the manufacturers of playback devices.

91. The ’920 patent solves this problem with systems and methods for decoding and playing back secure content on a variety of playback devices using multiple levels of content encryption, including encryption keys assigned to a specific user account. *See, e.g.*, *id.* at 6:14-28; 10:44-11:27. Utilizing encryption keys assigned to users adds an additional level of security that improves the security of digital content compared to the prior art. *See, e.g.*, *id.* at 10:44-11:27.

92. In particular, claim 1 of the ’920 patent recites a method of decoding and decrypting encrypted content with multiple levels of encryption using a playback device on which an active user encryption key is stored. *Id.* at claim 1. The method includes decrypting the content using the active user encryption key, specific to the user. Claim 1 recites a novel solution for improving the security of digital content in a manner that was not well-understood, routine, or conventional at the time of the ’920 patent.

V. The ’720 Patent

93. The ’720 patent, entitled “Systems and Methods for Automatically Generating Top Level Index Files,” was duly and legally issued on February 23, 2016, from a patent application filed July 21, 2014, with Jason Braness, Evan Wallin, and Ederson Ferreira as the named inventors. The ’720 patent claims priority to U.S. Provisional Application No. 61/529,403, filed on August 31, 2011.

94. The inventions recited in the ’720 patent enable Netflix to offer adaptive bitrate streaming services that perform smoothly and without stalls when switching among video streams of different resolution during playback on a user’s device. Specifically, the ’720 patent is directed to a playback server system that automatically generates a top-level index file tailored to a particular playback device that the playback device uses to request a streaming file, improving adaptive bitrate streaming.

95. The ’720 patent addresses a technical problem related to “adaptive bitrate streaming,” a popular digital video streaming method. “Adaptive bit rate streaming or adaptive streaming involves detecting the present streaming conditions . . . in real time and adjusting the quality of the streamed media accordingly.” *See, e.g.*, ’720 patent, 1:30-45. Many different types of consumer devices can play back video delivered over the internet, including computers, mobile phones, Blu-ray players, television, and set-top boxes. *See, e.g.*, *id.* at 9:1-8. All of these devices have different characteristics and technical capabilities for video playback. *See,*

e.g., *id.* at 7:55-62; 11:46-66; 12:20-31. Adaptive bitrate streaming further increases the complexity of digital video delivery by enabling the playback device to switch among different quality streams based on changes in device conditions. *See, e.g.*, *id.* at 1:30-45; 12:20-31. Each playback device needs a separate index file for each piece of video content that a user will watch using adaptive bitrate streaming. *See, e.g.*, *id.* at 12:20-40. The computing resources needed to compile and maintain a separate index for each combination of content and device make such a system infeasible. Further, using the same index for devices with different characteristics would produce poor playback, including video stalls, on many devices. Accordingly, a need existed for an efficient system to automatically generate index files for different playback devices for adaptive bitrate streaming based on device characteristics in order to improve the performance of the computing devices playing back video.

96. The ’720 patent claims a solution to this problem with systems and methods for automatically generating a top-level index file for a particular playback device and particular video content for use in adaptive bitrate streaming. The ’720 patent describes filtering the streams associated with requested content using criteria specific to the playback device, to generate a top-level index file. *See, e.g.*, *id.* at 2:24-28. The “top level index is a file that describes the location and content of container files containing streams of media (for example, audio, video, metadata, and subtitles) that can be utilized by the playback device to stream and playback content.” *See, e.g.*, *id.* at 6:39-43. The playback server system filters the streams based on playback device capabilities, information associated with the user account, or other rules defined by the content owner. *See, e.g.*, *id.* at 6:50-55. The playback server system then sends the top-level index file to the playback device for use in adaptive bitrate streaming for improved performance.

97. In particular, claim 1 of the ’720 patent recites a specific way to generate a top-level index file for adaptive bitrate streaming tailored to a specific

playback device. *Id.* at claim 1. The playback server system generates the top-level index file based on capabilities of the device, and sends the index to the playback device, which can use the index "to determine which assets to request for playback on the device"—for more efficient adaptive bitrate streaming specific to the technical capabilities of a particular playback device. Claim 1 recites a novel solution for improving the performance of adaptive bitrate streaming in a manner that was not well-understood, routine, or conventional at the time of the '720 patent.

VI. The '515 Patent

98. The '515 patent, entitled "Systems and Methods for Automatically Generating Top Level Index Files," was duly and legally issued on June 12, 2018, from a patent application filed January 28, 2016, with Jason Braness, Evan Wallin, and Ederson Ferreira as the named inventors. The '515 patent claims priority to U.S. Provisional Application No. 61/529,403, filed on August 31, 2011.

99. The inventions recited in the '515 patent enable Netflix to offer adaptive bitrate streaming services that perform smoothly and without stalls when switching among video streams of different resolution during playback on a user's device. Specifically, the '515 patent is directed to a playback server system that automatically generates a top-level index file tailored to a particular playback device that the playback device uses to request a streaming file, improving adaptive bitrate streaming.

100. The '515 patent shares a specification with the '720 patent and thus addresses the corresponding technical problem related to adaptive bitrate streaming for a diverse device ecosystem with many different kinds of devices and corresponding technical capabilities. *See, e.g.*, '515 patent, 1:30-45; 8:2-9; 9:17-23; 11:65-12:16; 12:40-60; *see also supra* ¶ 95.

101. The '515 patent claims a solution to the problem with systems and methods for automatically generating a top-level index file for a particular playback

device for use in adaptive bitrate streaming, based on the device's specific attributes, including the type of device. *See, e.g.*, *id.* at 11:40-46; *see also supra* ¶ 96.

102. In particular, claim 1 of the '515 patent recites a specific way to generate a top-level index file for adaptive bitrate streaming tailored to a specific playback device. *Id.* at claim 1. The playback server system generates the top-level index file based on the type of the device and software version loaded on the device, and sends the index to the playback device—for more efficient adaptive bitrate streaming specific to the technical capabilities of a particular playback device. Claim 1 recites a novel solution for improving the performance of adaptive bitrate streaming in a manner that was not well-understood, routine, or conventional at the time of the '515 patent.

VII. The '486 Patent

103. The '486 patent, entitled "Elementary Bitstream Cryptographic Material Transport Systems and Methods," was duly and legally issued on February 19, 2019, from a patent application filed June 6, 2017, with Francis Yee-Dug Chan, Kourosh Soroushian, and Andrew Jeffrey Wood as the named inventors. The '486 patent claims priority to U.S. Provisional Application No. 61/266,982, filed on December 4, 2009.

104. The inventions recited in the '486 patent enable Netflix to improve the security of its video streaming system, allowing it to obtain high-quality content from content providers and to trust in the security of its own, home-grown content. Specifically, the '486 patent is directed to a content security architecture that deciphers frame keys within a secure video decoder, efficiently enhancing content security.

105. The '486 patent addresses a technical problem. It explains that content providers need to make sure that only authorized users can access and play back digital content. *See, e.g.*, '486 patent, 1:31-35. This is a particular problem when the

content is transmitted over connections that are not secure and can be intercepted. *See, e.g.*, *id.* at 1:53-59. Accordingly, a need existed to improve the distribution of digital content to enhance security of content that may be transmitted over an unsecured connection, while enabling efficient access to the content for the correct users.

106. The ’486 patent claims a solution to this problem with specific ways to transmit “encrypted multimedia content over an unsecured connection” to improve security and enable efficient distribution and playback of multimedia content. *See, e.g.*, *id.* at 1:28-29. The ’486 inventions “do not secure the transmission but rather secure the data being transmitted via the unsecured connection.” *See, e.g.*, *id.* at 5:29-40. The inventions accomplish this by ciphering decryption key information in the multimedia data, and not deciphering those keys to decrypt the multimedia until the data is at the decoder and no longer being transmitted. *See, e.g.*, *id.*; *see also* 6:53-7:5. As a result, “by allowing the decryption to occur on the decoder the bitstream is protected even if the connection is compromised and an unauthorized component or process intercepts the bitstream.” *See, e.g.*, *id.* at 5:37-40.

107. In particular, claim 1 of the ’486 patent recites “deciphering a frame key” for a partially encrypted video frame on the playback device, and “decrypting the encrypted portion of each partially encrypted frame based upon the frame key.” *Id.* at claim 1. That is, the keys necessary to decrypt the video are protected until they are deciphered on the device. The invention recited in claim 1 solves the problem of enhancing multimedia content security by deciphering frame keys within a secure video decoder in a manner that was not well-understood, routine, or conventional at the time of the ’486 patent.

VIII. The ’588 Patent

108. The ’588 patent, entitled “Playback Devices and Methods for Playing Back Alternative Streams of Content Protected Using a Common Set of Cryptographic Keys,” was duly and legally issued on March 5, 2019, from a patent

application filed September 19, 2018, with Michael George Kiefer, Eric William Grab, and Jason Braness as the named inventors. The ’588 patent claims priority to U.S. Provisional Application No. 61/530,305, filed on September 1, 2011.

109. The inventions recited in the ’588 patent enable Netflix to offer its users an improved experience for adaptive bitrate streaming while maintaining the content security that it and other content providers require to make video content available over the internet. Specifically, the ’588 patent is directed to a DRM architecture that uses common frame encryption keys to encode alternate video streams, reducing playback stalls and improving performance during adaptive bitrate streaming.

110. The ’588 patent addresses a technical problem. “In many instances, content is divided into multiple streams,” and “some streams can be encoded as alternative streams that are suitable for different network connection bandwidths.” *See, e.g.*, ’588 patent, 1:45-58. In adaptive bitrate streaming, “the source media is encoded at multiple bitrates and the playback device or client switches between streaming the different encodings depending on available resources.” *See, e.g.*, *id.* at 1:59-67. Prior to the ’588 invention, each stream used different cryptographic information for authorizing secure playback. *See, e.g.*, *id.* at 8:37-61; 9:65-10:31. Storing and processing cryptographic information for each stream requires more computing resources and increases the cost and complexity of the playback device, and it can also result in stalls and delays when switching among video streams with different bitrates. *See, e.g.*, *id.* Accordingly, a need existed for a more efficient and high-performance DRM implementation for adaptive bitrate streaming that would reduce the computer memory consumed by cryptographic information and reduce the time and computing resources consumed by playback devices when switching among video streams having different bitrates.

111. The ’588 patent claims a solution to this problem with specific ways to reduce the computer memory and other resources consumed by cryptographic

information during adaptive bitrate streaming. With the ’588 invention, “each of the alternative streams of protected content are encrypted using common cryptographic information.” *See, e.g., id.* at Abstract; *see also id.* at 2:66-3:30; 8:37-61; 9:65-10:31. The ’588 invention allows an adaptive bitrate streaming system to switch among video streams having different bitrates more efficiently, consuming fewer computing resources and avoiding interruptions in video playback, improving the performance of the computing system.

112. In particular, claim 1 of the ’588 patent recites a “top level index file identifying a plurality of alternative streams of protected video,” “wherein each of the alternative streams of protected video includes partially encrypted video frames that are encrypted using a set of common keys comprising at least one key.” *Id.* at claim 1. Claim 1 solves the problem of inefficient and low-performance video playback caused by the use of different cryptographic information for each video stream in an adaptive bitrate streaming service in a manner that was not well-understood, routine, or conventional at the time of the ’588 patent.

NETFLIX’S INTERNAL TESTING

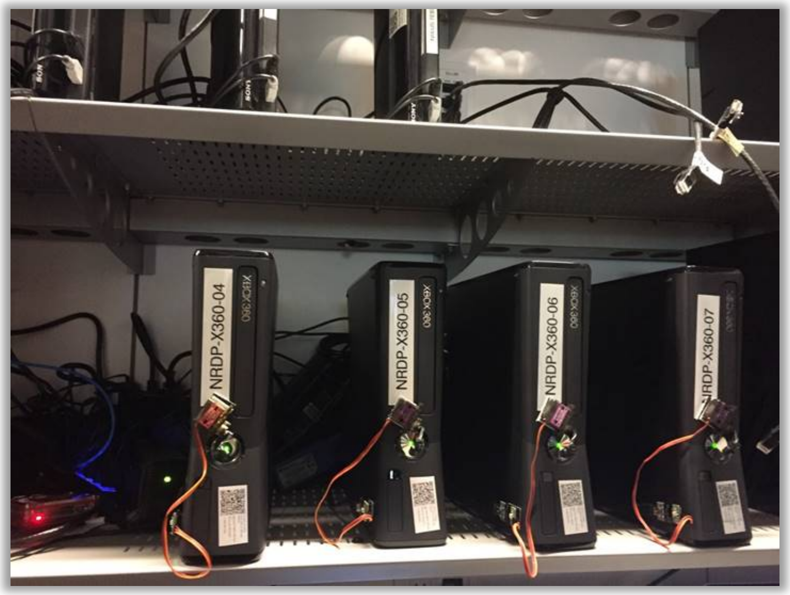
113. Upon information and belief, Netflix tests its software application and video streaming service on CE devices to confirm that the application and service work properly before releasing them to users.

114. Upon information and belief, device testing is important to Netflix’s success. Device testing allows Netflix to ensure that its application and service operate seamlessly on Netflix-compatible devices—a large ecosystem. Netflix’s testing further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible and operable with consumer devices.

115. Netflix has acknowledged the importance of device testing. “As part of the Netflix SDK team, our responsibility is to ensure the new release version of the Netflix application is thoroughly tested to its highest operational quality before deploying onto gaming consoles and distributing as an SDK (along with a reference

application) to Netflix device partners; eventually making its way to millions of smart TV's and set top boxes (STB's). Overall, our testing is responsible for the quality of Netflix running on millions of gaming consoles and internet connected TV's/STB's."[33]

116. Netflix has tested its application and service on, for example, Xbox 360, PlayStation 3, and PlayStation 4. For example, shown below are photographs provided by Netflix of Xbox 360 game consoles operating in an automated internal Netflix test environment: [34]



[33] https://medium.com/netflix-techblog/automated-testing-on-devices-fc5a39f47e24.

[34] *Id.*

117. As of August 10, 2016, Netflix employees estimated that the Netflix ecosystem ran approximately 20,000 test cases per day.[35]

118. Upon information and belief, Netflix directly infringes the DivX Patents during Netflix's internal testing of its application and video streaming service on consumer devices.

119. Upon information and belief, Netflix's internal testing enables Netflix to deliver its application and service in the United States and worldwide.

120. Upon information and belief, Netflix tests the DRM technologies that it employs to protect the security of the video content that it licenses from third parties, including studios, and that it produces itself.[36] Upon information and belief, Netflix's agreements with studios require that Netflix agree to provide secure DRM to protect content.[37]

[35] *Id.*

[36] Mark Watson, Netflix, *Adaptive HTTP streaming and HTML5*, W3C Web and TV Workshop (Feb. 8-9, 2011), *available at* https://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_62.pdf.

[37] https://www.webpronews.com/netflix-to-start-testing-html5-streaming-this-year/.

121. Upon information and belief, Netflix's internal testing of the DRM technologies it employs, therefore, enables Netflix to obtain video content from third parties and to invest in its own production of original content, which leads to increased adoption of Netflix's service by paying members in the United States and worldwide. Netflix contends that its ability to offer content differentiates its service from competitors and directly leads to attracting and retaining members.[38]

122. Indeed, Netflix identifies any compromise to its ability to obtain content as one a material risk to Netflix's business.[39] Upon information and belief, Netflix offsets this risk through its internal testing of the technologies it uses to secure and stream video over the internet, including DRM.

NETFLIX'S INDIRECT INFRINGEMENT

123. Netflix has indirectly infringed and continues to indirectly infringe at least the '673 patent, the '651 patent, the '792 patent, the '920 patent, the '515 patent, the '486 patent, and the '588 patent (collectively, the "Indirectly Infringed DivX Patents") by inducing third parties to directly infringe those patents.

124. Netflix has induced, and continues to induce, direct infringement of the Indirectly Infringed DivX Patents by customers, importers, sellers, resellers, and/or end users of infringing playback devices enabled with the Netflix application and service.

I. Netflix's Knowledge of the DivX Patents

125. At the very latest, Netflix had actual knowledge of the DivX Patents and of its infringement as of the date of this Complaint.

[38] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

[39] *Id.*

126. Netflix has known of DivX and its technology for more than a decade. At least as of 2004, Netflix had engaged with DivX in discussions regarding DivX's technology.

127. In prosecuting its own patents, Netflix has cited to at least one DivX patent application. Netflix's U.S. Patent Nos. 9,565,425 and 9,727,321 both cite U.S. Patent Application Number 2013/0007443 (to Grab, et al.), which issued as U.S. Patent No. 9,092,646, filed March 21, 2012, titled "Systems and methods for identifying consumer electronic products based on a product identifier." Netflix had knowledge that DivX owned patents relating to its digital video technologies at least as of the earliest date that it cited the DivX application, April 28, 2016.

II. Netflix's Knowledge of Third-Party Actions Infringing DivX's Patents

128. Netflix is a known market leader and one of the dominant players in internet digital video streaming.

129. Netflix knows that it provides and markets an application, through its website, the Apple App Store, and the Google Play Store, for use on playback devices that causes the playback devices and their users, importers, sellers, resellers, and customers to directly infringe Indirectly Infringed DivX Patents, when used as intended with Netflix's internet video streaming service. Indeed, Netflix touts that its "streaming software allows you to instantly watch content from Netflix through any internet-connected device that offers the Netflix app, including smart TVs, game consoles, streaming media players, set-top boxes, smartphones, and tablets."[40]

130. Netflix actively encourages the installation and use of its application and service on consumer devices. Netflix has successfully pursued agreements with cable, satellite, and telecommunications operators to make Netflix's service

[40] https://help.netflix.com/en/node/412.

available through television set-top boxes.[41] Netflix also has entered into agreements with other consumer electronics device manufacturers to make Netflix's service available on those consumer devices.[42] Those products include streaming media players, smart TVs, game consoles, Blu-ray players, smartphones and tablets, and personal computers.[43] Netflix recommends, directly to consumers, certain consumer electronics devices preloaded with Netflix.[44]

131. Netflix knows that its application is enabled in millions of infringing playback devices, claiming that its members are "streaming on more than half a billion devices spanning over 1,700 different types of devices from hundreds of brands."[45] Upon information and belief, Netflix knows which of its users install its software on their devices and stream video using Netflix's streaming service in the United States.

132. Netflix knows that third parties—including playback device manufacturers, importers, sellers, resellers, users, and customers—make, use, offer to sell, sell, and/or import into the United States playback devices and other products that incorporate and enable the Netflix application. Indeed, Netflix encourages use of its application on "thousands of internet-connected devices," and it advertises that many devices "have Netflix already on them—ready for you to watch":[46]

[41] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

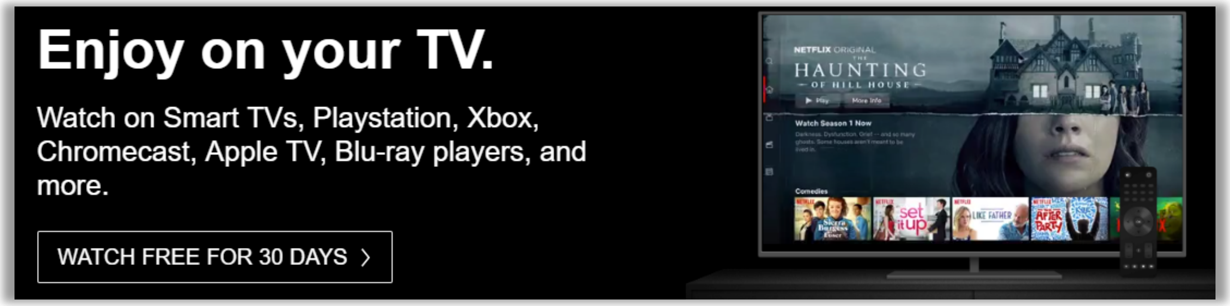
[42] https://devices.netflix.com/en/.

[43] *Id.*

[44] https://devices.netflix.com/en/recommendedtv/2018/.

[45] https://medium.com/netflix-techblog/detecting-performance-anomalies-in-external-firmware-deployments-ed41b1bfcf46.

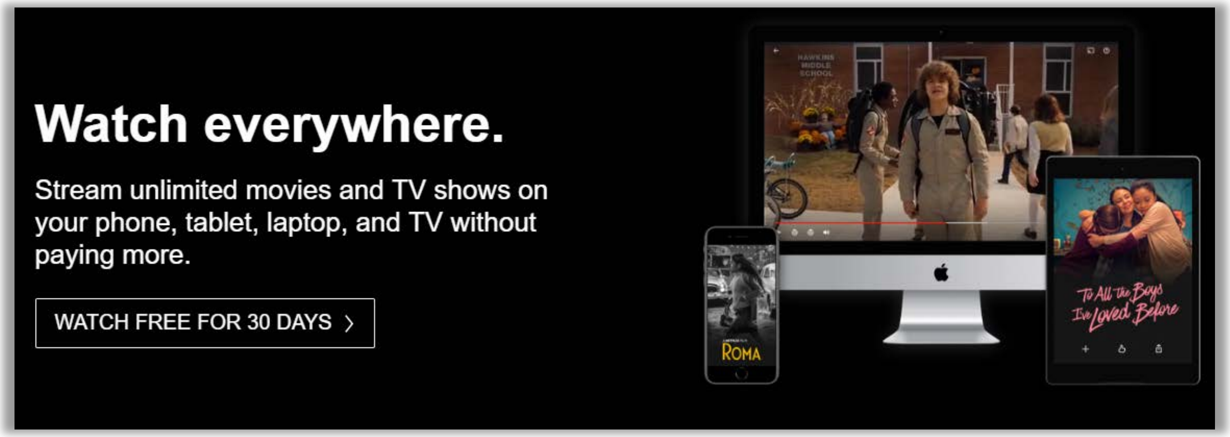
[46] https://www.netflix.com/; https://devices.netflix.com/en/.



Enjoy on your TV.

Watch on Smart TVs, Playstation, Xbox, Chromecast, Apple TV, Blu-ray players, and more.

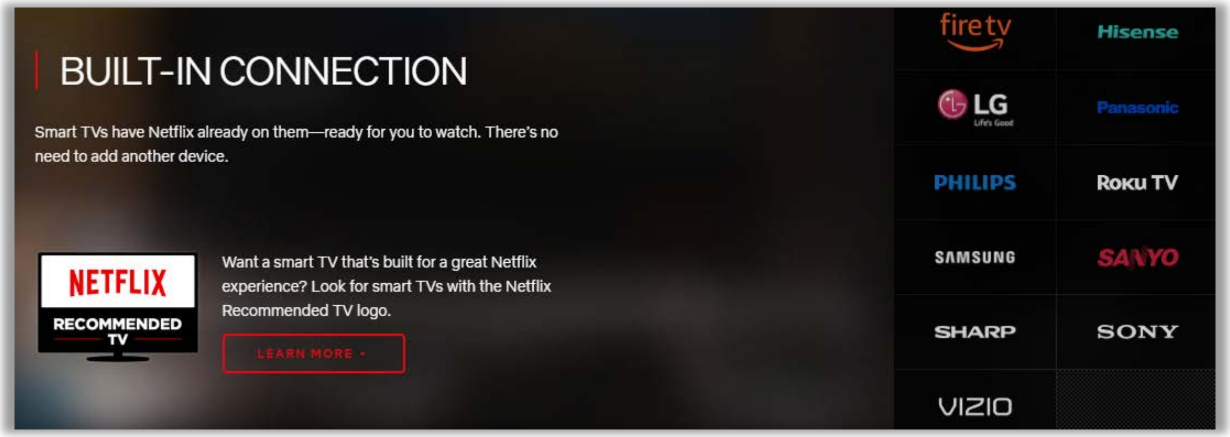
WATCH FREE FOR 30 DAYS ›



Watch everywhere.

Stream unlimited movies and TV shows on your phone, tablet, laptop, and TV without paying more.

WATCH FREE FOR 30 DAYS ›



BUILT-IN CONNECTION

Smart TVs have Netflix already on them—ready for you to watch. There's no need to add another device.

Want a smart TV that's built for a great Netflix experience? Look for smart TVs with the Netflix Recommended TV logo.

LEARN MORE ›

133. Upon information and belief, Netflix has designed its application such that, when third party CE playback devices incorporate and/or enable the Netflix application and such third party devices with the Netflix application are used as intended, the third-party products with the application directly infringe one or more claims of the Indirectly Infringed DivX Patents when made, used, offered for sale,

or sold in the United States, or when imported into the United States, as set forth in exemplary detail in the Counts herein.

134. At least as of the date of this Complaint, and based on its knowledge of the scope of the DivX Patents, its application, and products enabling that application, Netflix knows that third party sellers, resellers, importers, customer end-users, and other third parties have directly infringed and continue to directly infringe at least one claim of each of the Indirectly Infringed DivX Patents, as set forth in exemplary detail in the Counts herein.

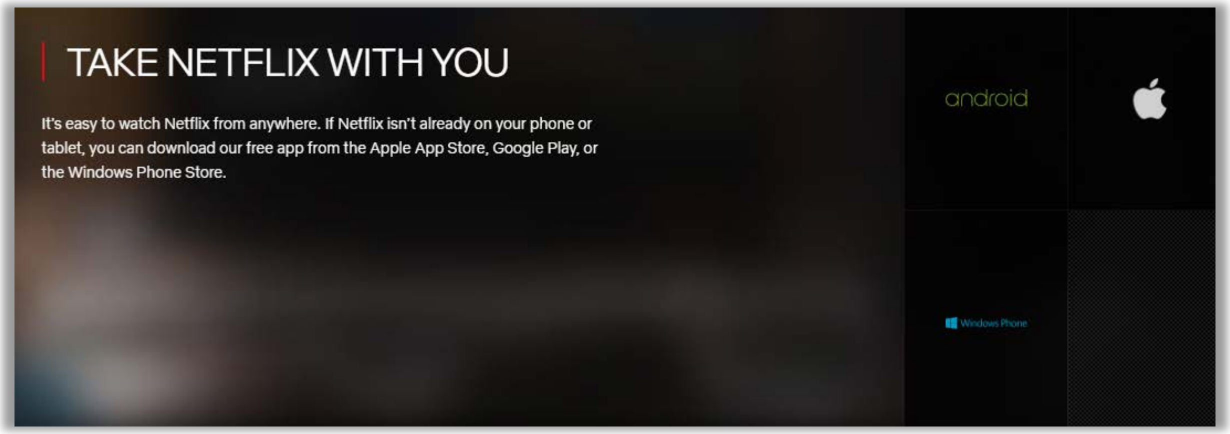
III. Netflix's Specific Intent to Cause Third-Party Actions Infringing DivX's Patents

135. Upon information and belief, Netflix has designed, marketed, and sold its application and service to third parties with knowledge and the specific intent to cause the third parties to make, use, offer to sell, or sell in the United States, and/or import into the United States products incorporating and enabling the Netflix application and service.

136. Upon information and belief, Netflix actively encourages its customers and end users to directly infringe the Indirectly Infringed DivX Patents by encouraging them to use the Netflix application as intended on various playback devices.

137. Netflix specifically encourages its customers to download its application onto smart phones and tablets through the Apple App Store for iOS devices or through the Google Play Store for Android devices:[47]

[47] https://devices.netflix.com/en/.



TAKE NETFLIX WITH YOU

It's easy to watch Netflix from anywhere. If Netflix isn't already on your phone or tablet, you can download our free app from the Apple App Store, Google Play, or the Windows Phone Store.

138. Netflix develops its application and service for third parties, promotes its application and service and the infringing third party products that incorporate the application and service to customers in the United States, and actively drives the adoption and use of its application and service through agreements with cable, satellite, and telecommunications operators, and consumer electronics manufacturers and sellers.[48]

139. Upon information and belief, at least as of the date of this Complaint, Netflix intends and continues to intend to induce patent infringement by these third parties, has actual knowledge that the inducing acts cause infringement, or is willfully blind to the possibility that its inducing acts cause infringement.

140. Upon information and belief, Netflix indirectly infringes one or more claims of the Indirectly Infringed DivX Patents by inducing numerous third-parties to make, have made, use, sell, offer to sell, and/or import into the United States playback devices with the Netflix application installed and/or enabled.

COUNT I: INFRINGEMENT OF U.S. PATENT NO. 7,295,673

141. The allegations of paragraphs 1-140 of this Complaint are incorporated by reference as though fully set forth herein.

[48] *Id.*; Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

142. Pursuant to 35 U.S.C. § 282, the ’673 patent is presumed valid.

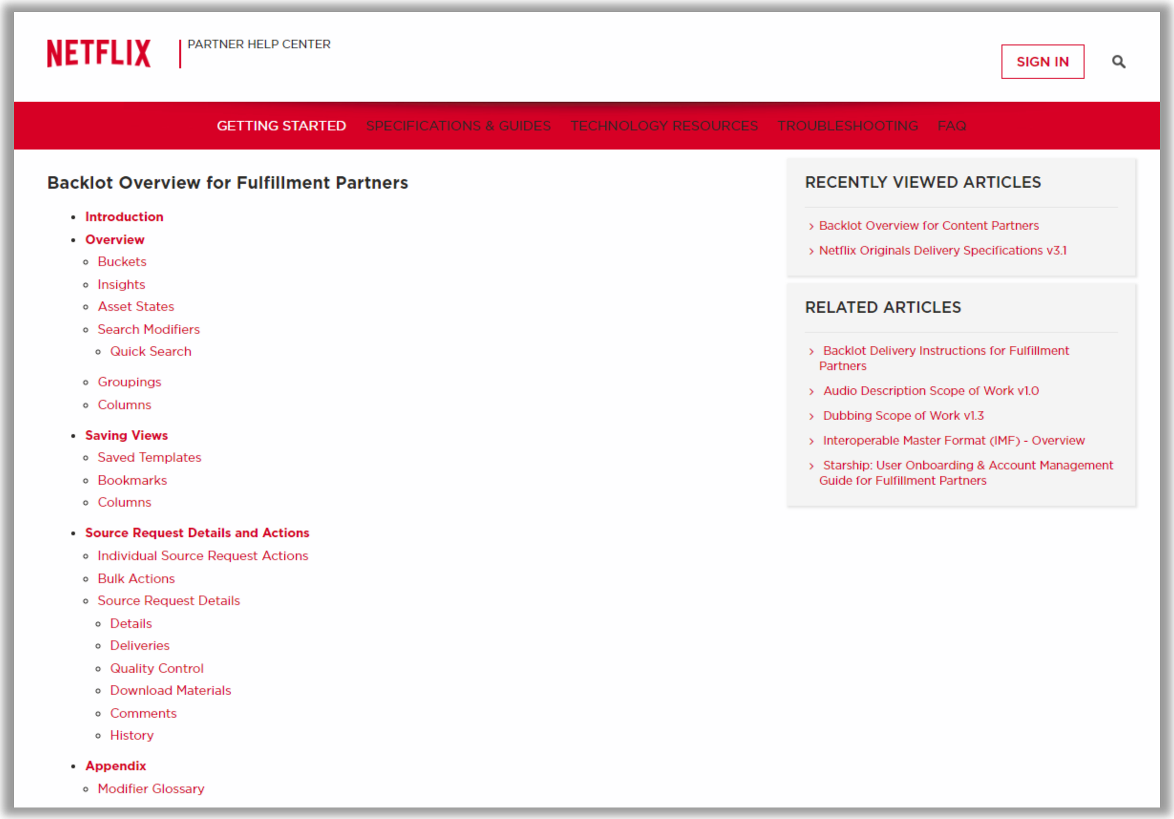
143. Upon information and belief, Netflix directly infringes the ’673 patent by making, using, offering to sell, selling, and/or importing into the United States its Netflix service, which provides a method and system for securing compressed digital video (collectively, the “Accused ’673 Infringing Products”).

144. Upon information and belief, the Accused ’673 Infringing Products directly infringe at least claim 1 of the ’673 patent at least in the exemplary manner described in paragraphs 145-152 below.

145. Netflix provides a “method for producing a protected stream of compressed video content,” namely, Netflix encoding, encrypting, and packaging videos in the H.265 (HEVC) format for streaming.

146. Netflix “receiv[es] an input stream of compressed video content containing a sequence of frames” by providing a “Backlot” for studios to upload content in JPEG2000 format, for example, which contains compressed video content containing a sequence of frames.[49]

[49] https://partnerhelp.netflixstudios.com/hc/en-us/articles/115004872247-Backlot-Overview-for-Fulfillment-Partners#Intro.



Backlot Overview for Fulfillment Partners

- Introduction
- Overview
  - Buckets
  - Insights
  - Asset States
  - Search Modifiers
    - Quick Search
  - Groupings
  - Columns
- Saving Views
  - Saved Templates
  - Bookmarks
  - Columns
- Source Request Details and Actions
  - Individual Source Request Actions
  - Bulk Actions
  - Source Request Details
    - Details
    - Deliveries
    - Quality Control
    - Download Materials
    - Comments
    - History
- Appendix
  - Modifier Glossary

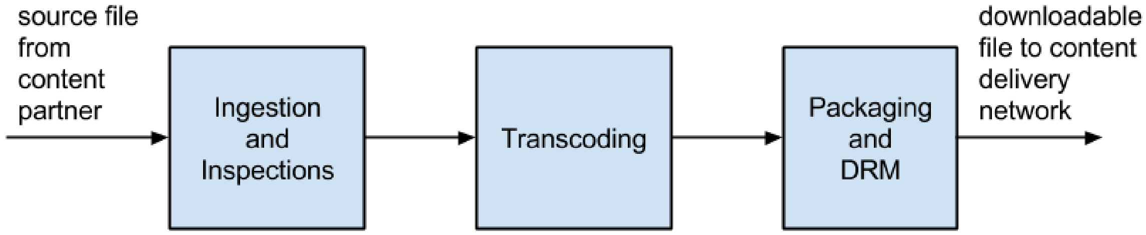
RECENTLY VIEWED ARTICLES

› Backlot Overview for Content Partners
› Netflix Originals Delivery Specifications v3.1

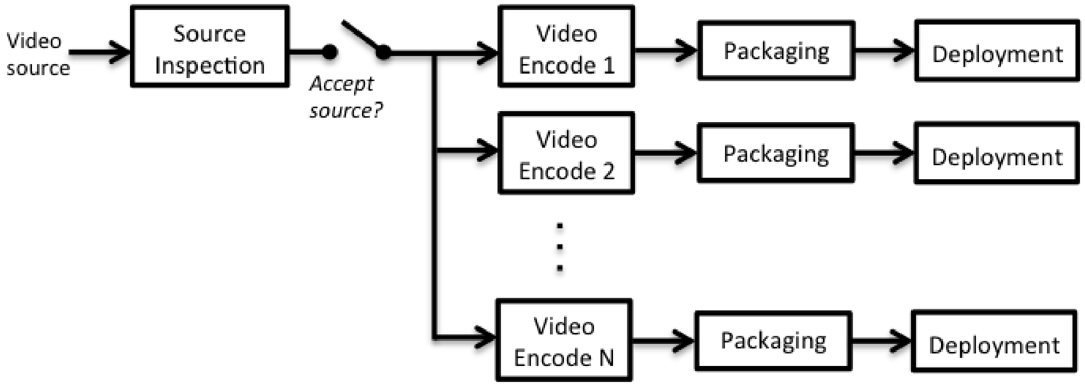
RELATED ARTICLES

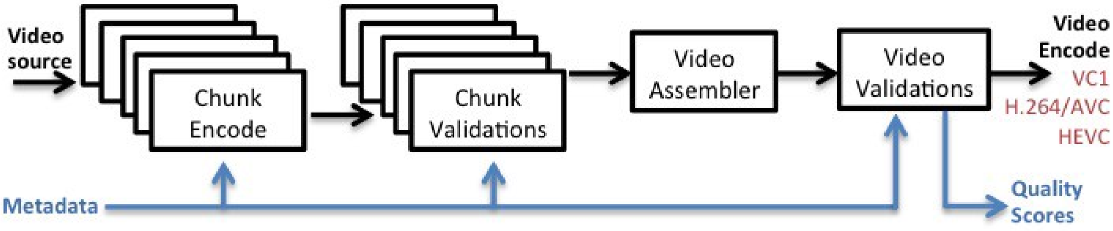
› Backlot Delivery Instructions for Fulfillment Partners
› Audio Description Scope of Work v1.0
› Dubbing Scope of Work v1.3
› Interoperable Master Format (IMF) - Overview
› Starship: User Onboarding & Account Management Guide for Fulfillment Partners

147. Netflix encodes videos via, for example, an H.265 codec, which becomes an input stream of compressed video content containing a sequence of frames.[50]


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

[50] https://medium.com/netflix-techblog/the-netflix-imf-workflow-f45dd72ed700; https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746.





148. Netflix "generat[es] a frame encryption key and stor[es] the encryption key in a key table" by creating a frame encryption key and storing it in a key table comprising multiple keys during sample encryption of the video in accordance with the ISO Common Encryption Standard and the Microsoft PIFF Specification.

149. Netflix "creat[es] a set of encrypted frames by encrypting at least selected portions of selected frames of said sequence of frames using the frame encryption keys in accordance with a frame encryption function" because Netflix's MP4 PIFF box specifies the use of sample encryption in accordance with the ISO Common Encryption Standard and the Microsoft PIFF Specification.

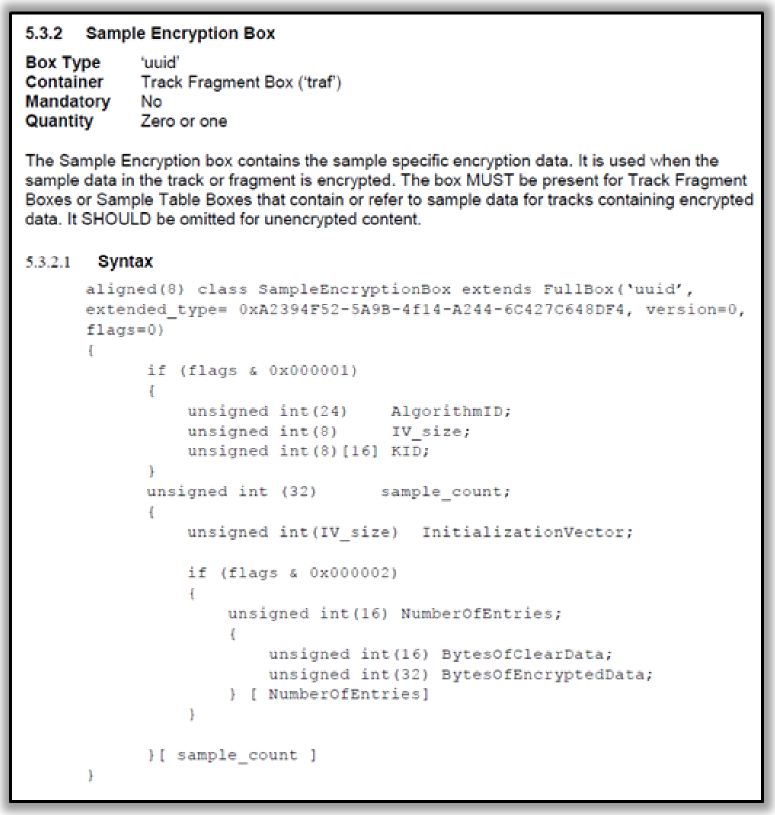
150. Netflix "generat[es] frame decryption information necessary to decrypt said set of encrypted frames including an encryption key pointer identifying a decryption key to be used in the decryption of each encrypted frame" by generating frame decryption information that includes (1) a pointer to a decryption key, and (2) information about the frames and portions of frames that are encrypted. Netflix, through its support of the ISO Common Encryption Standard and the Microsoft

PIFF Specification, which teach frame decryption information, includes an encryption key pointer in the files it encodes. The encryption key pointer identifies a decryption key to be used in the decryption of each encrypted frame.

151. Netflix "assembl[es] at least said set of encrypted frames, unencrypted frames of said sequence of frames, and said frame decryption information to produce the protected stream of compressed video content" by assembling the requisite information into MP4 files. The manifest delivered from Netflix and the files streamed using the manifest indicate that the Netflix encoding and packaging process creates the video file downloaded from Netflix's content delivery network (CDN), which is a protected stream of compressed video content.

152. Netflix's "frame decryption information is synchronized with said set of encrypted frames into a synchronized frame decryption stream" when Netflix synchronizes the frame decryption information by interleaving the PIFF Sample Encryption Boxes (uuid) and media data, or "mdat," boxes throughout the MP4 file. In addition, the PIFF Sample Encryption Box contains a separate entry for each frame in the corresponding mdat box. The Microsoft PIFF Specification and the ISO Common Encryption Standard disclose that the frame decryption information is synchronized with the set of encrypted frames into a synchronized frame decryption stream. For example, Microsoft PIFF-based schemes disclose the "Sample Encryption Box," which contains the sample-specific encryption data and are synchronized with the encrypted frames within the stream.[51]

[51] The Protected Interoperable File Format (PIFF) Microsoft, page 16.



5.3.2 Sample Encryption Box

Box Type 'uuid'
Container Track Fragment Box ('traf')
Mandatory No
Quantity Zero or one

The Sample Encryption box contains the sample specific encryption data. It is used when the sample data in the track or fragment is encrypted. The box MUST be present for Track Fragment Boxes or Sample Table Boxes that contain or refer to sample data for tracks containing encrypted data. It SHOULD be omitted for unencrypted content.

5.3.2.1 **Syntax**

```
aligned(8) class SampleEncryptionBox extends FullBox('uuid',
extended_type= 0xA2394F52-5A9B-4f14-A244-6C427C648DF4, version=0,
flags=0)
{
      if (flags & 0x000001)
      {
          unsigned int(24)    AlgorithmID;
          unsigned int(8)     IV_size;
          unsigned int(8)[16] KID;
      }
      unsigned int (32)      sample_count;
      {
          unsigned int(IV_size)  InitializationVector;

          if (flags & 0x000002)
          {
              unsigned int(16) NumberOfEntries;
              {
                  unsigned int(16) BytesOfClearData;
                  unsigned int(32) BytesOfEncryptedData;
              } [ NumberOfEntries]
          }

      }[ sample_count ]
}
```

153. Netflix directly infringes at least claim 1, at least as described, when it tests its service using various playback devices.

154. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

155. Netflix has infringed, and continues to infringe, at least claim 1 of the ’673 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’673 Infringing Products, in violation of 35 U.S.C. § 271(a).

156. Netflix has induced, and continues to induce, infringement of at least claim 14 of the ’673 patent, at least in the exemplary manner described in paragraphs 157-164, in violation of 35 U.S.C. § 271(b).

157. At least as of the date of this Complaint, Netflix knows that the ’673 patent enables it to stream video to a diverse array of consumer devices while protecting the video content with secure encryption and decryption, allowing Netflix to both offer its service on a diverse device ecosystem and provide high-quality video content. Specifically, at least as of the date of this Complaint, Netflix knows that the ’673 patent is directed to a partial frame encryption architecture that enables more efficient streaming of encrypted video to any device, providing secure decryption without decoding.

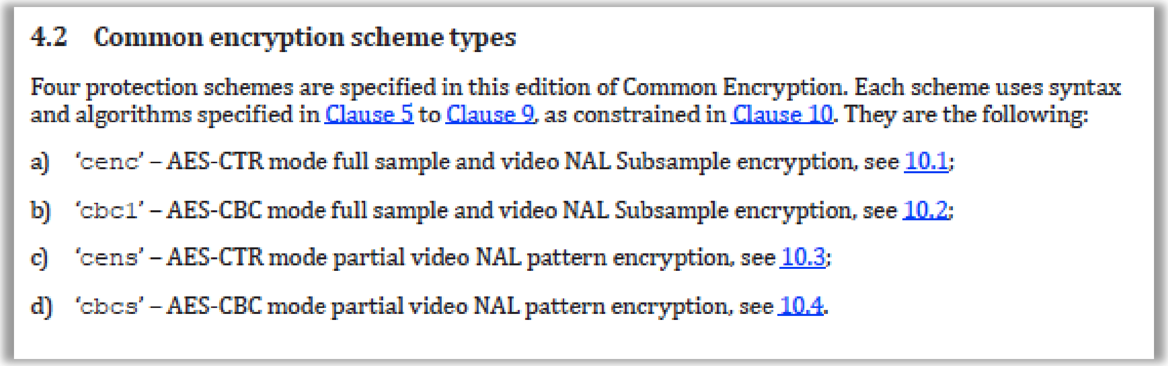
158. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, meets the limitations of claim 14.

159. At least as of the date of this Complaint, Netflix knows and specifically intends that its end users practice the method recited in claim 14 at least in the exemplary manner described below, when using its application and service as intended—namely, decrypting a protected stream of compressed video content.

160. Netflix induces “receiving an input stream of compressed video content containing encrypted frames and unencrypted frames” when its application enabled on a CE playback device receives an input stream of compressed video content containing encrypted frames and unencrypted frames. The ISO Common Encryption Standard and the Microsoft PIFF Specification have specified common encryption scheme types for ISO-based and PIFF-based media file format files.[52] For example, Netflix’s MP4 files include PIFF boxes that specify the use of sample encryption in accordance with the ISO Common Encryption Standard and the Microsoft PIFF Specification.

[52] *See* ISO/IEC 23001-7 at 3.

161. When encrypting the compressed video, Netflix uses sample encryption where a NAL unit may be fully encrypted, partially encrypted, or not encrypted.



4.2 Common encryption scheme types

Four protection schemes are specified in this edition of Common Encryption. Each scheme uses syntax and algorithms specified in Clause 5 to Clause 9, as constrained in Clause 10. They are the following:

a) 'cenc' – AES-CTR mode full sample and video NAL Subsample encryption, see 10.1;

b) 'cbc1' – AES-CBC mode full sample and video NAL Subsample encryption, see 10.2;

c) 'cens' – AES-CTR mode partial video NAL pattern encryption, see 10.3;

d) 'cbcs' – AES-CBC mode partial video NAL pattern encryption, see 10.4.

162. Netflix induces "receiving frame decryption information necessary to decrypt said encrypted frames, said frame decryption information is synchronized with said set of encrypted frames into a synchronized frame decryption stream and distinguishes said encrypted frames from said unencrypted frames" when its application enabled on a CE playback device receives frame decryption information containing key information and other information for proper decryption of each and every sample. Such frame decryption information can distinguish encrypted frames from unencrypted frames. The ISO Common Encryption Standard and the Microsoft PIFF Specification teach frame decryption information including an encryption key pointer identifying a decryption key to be used in the decryption of each encrypted frame. The ISO Common Encryption Standard and the Microsoft PIFF Specification further disclose that the frame decryption information is synchronized with the set of encrypted frames into a synchronized frame decryption stream. For example, Microsoft PIFF-based schemes disclose the "Sample Encryption Box," which contains the sample-specific encryption data and are synchronized with the encrypted frames within the stream.[53]

[53] The Protected Interoperable File Format (PIFF) Microsoft, page 16.

163. Netflix induces "obtaining an applicable frame decryption key from the received frame decryption information" when its application enabled on a CE playback device obtains frame decryption information, for example, as specified by the ISO Common Encryption Standard and the Microsoft PIFF Specification.

164. Netflix induces "decrypting selected portions of said encrypted frames using a frame decryption function in accordance with said frame decryption information, which identifies the specific portions of the frames to be decrypted and the applicable frame decryption key from the frame decryption information" when its application enabled on a CE playback device decrypts selected portions of said encrypted frames using a frame decryption function in accordance with said frame decryption information, which identifies the specific portions of the frames to be decrypted and the applicable frame decryption key from the frame decryption information. As described, since some encrypted frames are partially encrypted, the information contained within the decryption information will indicate which portion of said encrypted frames needs to be decrypted, and the applicable frame decryption key is used to decrypt the identified specific portions of the frames. The ISO Common Encryption Standard and the Microsoft PIFF Specification teach frame decryption information that includes an encryption key pointer identifying a decryption key to be used in the decryption of each encrypted frame.

165. Netflix's infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

COUNT II: INFRINGEMENT OF U.S. PATENT NO. 8,139,651

166. The allegations of paragraphs 1-165 of this Complaint are incorporated by reference as though fully set forth herein.

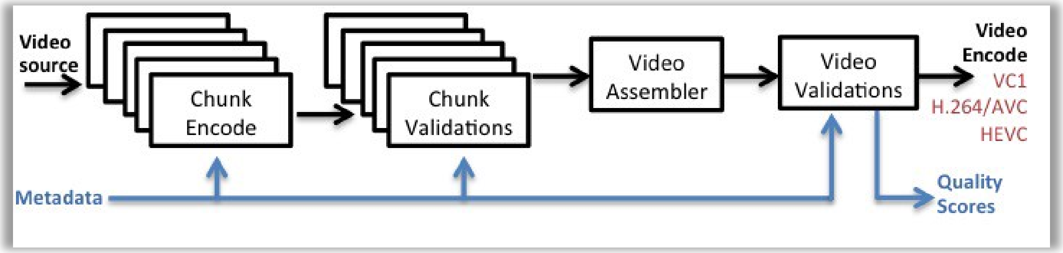
167. Pursuant to 35 U.S.C. § 282, the '651 patent is presumed valid.

168. Upon information and belief, Netflix directly infringes the '651 patent by making, using, offering to sell, selling, and/or importing into the United States

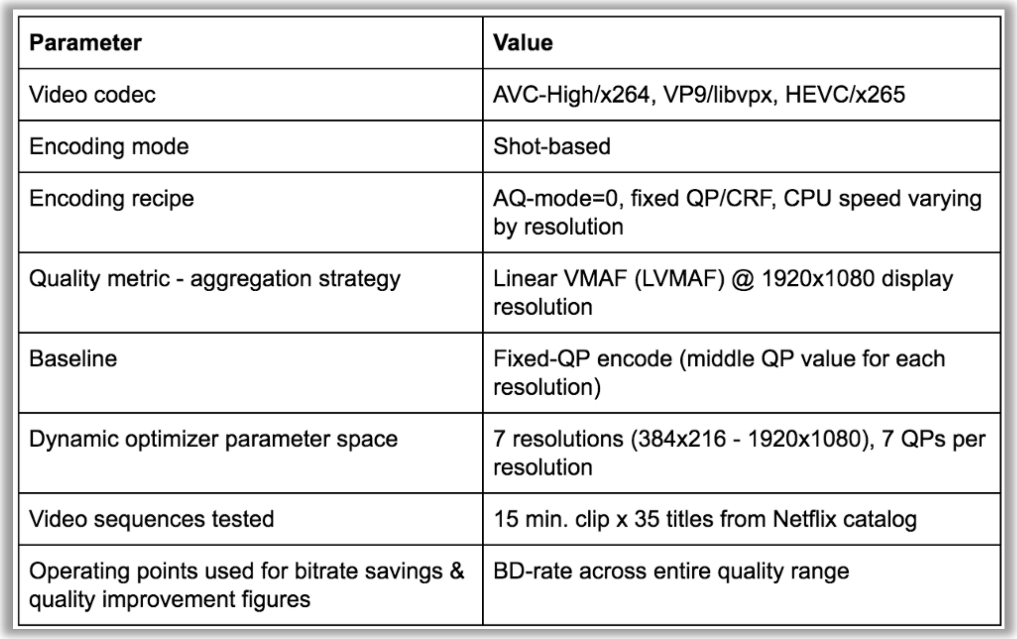
its Netflix service, which provides a video deblocking filter (collectively, the "Accused '651 Infringing Products").

169. Upon information and belief, the Accused '651 Infringing Products directly infringe at least claim 1 of the '651 patent at least in the exemplary manner described in paragraphs 170-173 below.

170. Netflix practices a "method of deblocking a reconstructed video frame." Netflix's encoding platform performs a method of deblocking a reconstructed video frame when encoding titles pursuant to the H.265 (HEVC) Standard. The method is an integral part of the video encoding process. Netflix encodes videos in H.265 format.[54]

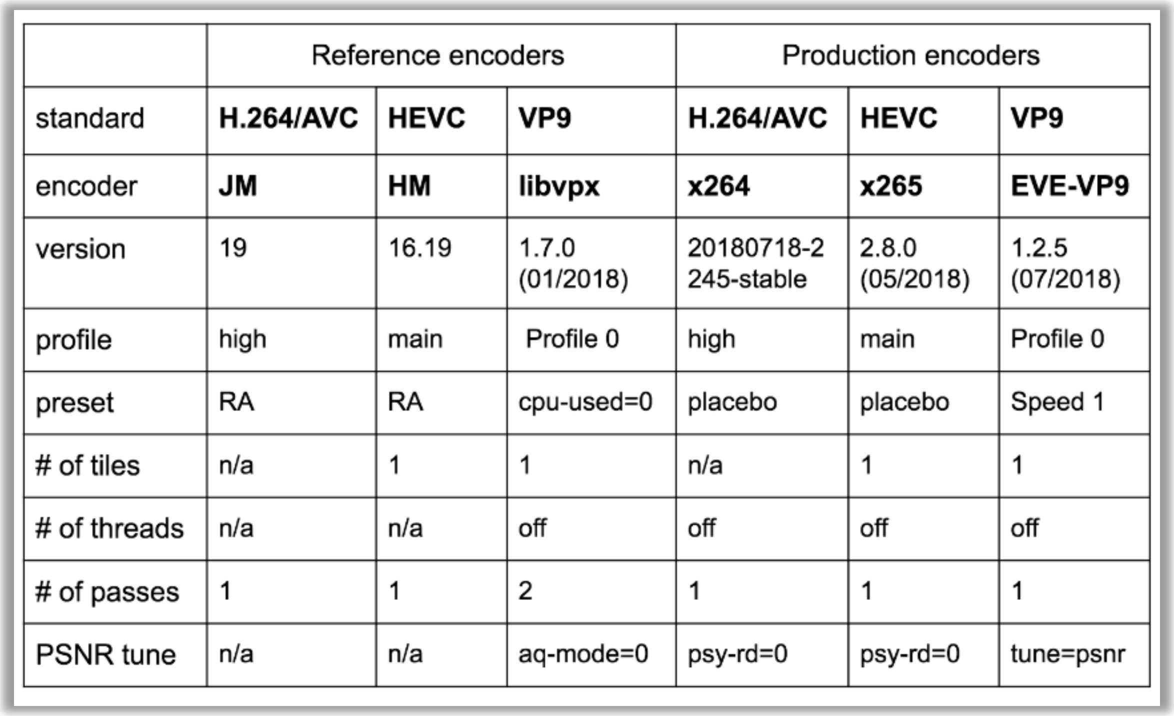


[54] https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746; https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f.



Parameter	Value
Video codec	AVC-High/x264, VP9/libvpx, HEVC/x265
Encoding mode	Shot-based
Encoding recipe	AQ-mode=0, fixed QP/CRF, CPU speed varying by resolution
Quality metric - aggregation strategy	Linear VMAF (LVMAF) @ 1920x1080 display resolution
Baseline	Fixed-QP encode (middle QP value for each resolution)
Dynamic optimizer parameter space	7 resolutions (384x216 - 1920x1080), 7 QPs per resolution
Video sequences tested	15 min. clip x 35 titles from Netflix catalog
Operating points used for bitrate savings & quality improvement figures	BD-rate across entire quality range

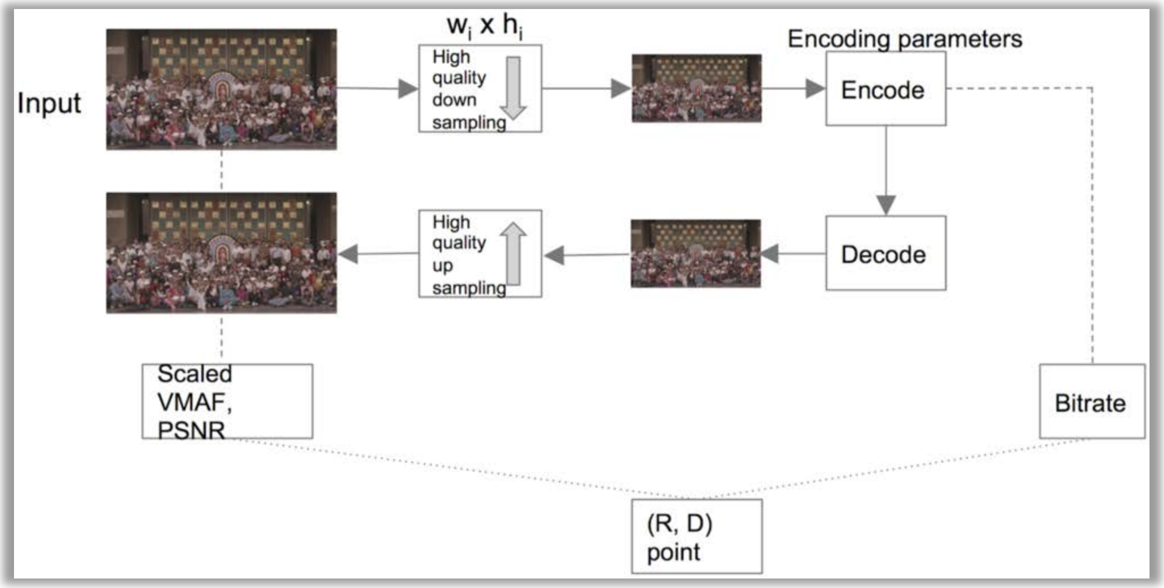
Netflix video content encoded for H.265 (HEVC) uses a "main" encoding profile.[55]



	Reference encoders			Production encoders		
standard	**H.264/AVC**	**HEVC**	**VP9**	**H.264/AVC**	**HEVC**	**VP9**
encoder	**JM**	**HM**	**libvpx**	**x264**	**x265**	**EVE-VP9**
version	19	16.19	1.7.0 (01/2018)	20180718-2 245-stable	2.8.0 (05/2018)	1.2.5 (07/2018)
profile	high	main	Profile 0	high	main	Profile 0
preset	RA	RA	cpu-used=0	placebo	placebo	Speed 1
# of tiles	n/a	1	1	n/a	1	1
# of threads	n/a	n/a	off	off	off	off
# of passes	1	1	2	1	1	1
PSNR tune	n/a	n/a	aq-mode=0	psy-rd=0	psy-rd=0	tune=psnr

[55] https://medium.com/netflix-techblog/performance-comparison-of-video-coding-standards-an-adaptive-streaming-perspective-d45d0183ca95.

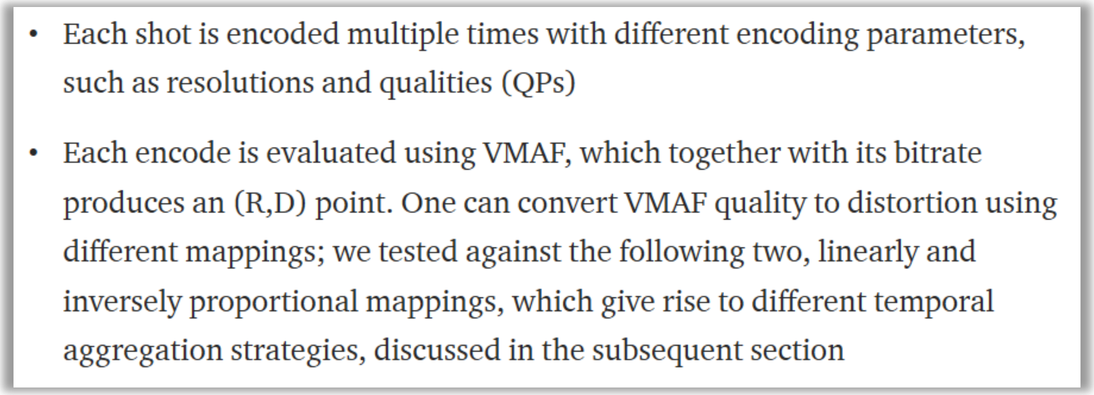
As part of the encoding process, Netflix performs per-title, per-chunk, or per-shot encoding.[56] Integral to this encoding process is an optimization process based on a quality measure. The quality measure is derived via Video Multimethod Assessment Fusion (VMAF) and/or peak signal-to-noise ratio (PSNR).[57]



ROBINS KAPLAN LLP
ATTORNEYS AT LAW
LOS ANGELES

[56] https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746 ("Similar to inspection, encoding is performed on chunks of the source file, which allows for efficient parallelization.").

[57] https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f ("We've implemented the dynamic optimizer framework in our encoding pipeline, leveraging our scalable cloud infrastructure . . . ."); https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2 ("We pre-encode streams at various bitrates applying optimized encoding recipes.").



- Each shot is encoded multiple times with different encoding parameters, such as resolutions and qualities (QPs)
- Each encode is evaluated using VMAF, which together with its bitrate produces an (R,D) point. One can convert VMAF quality to distortion using different mappings; we tested against the following two, linearly and inversely proportional mappings, which give rise to different temporal aggregation strategies, discussed in the subsequent section

Computing the quality measure(s) via VMAF and/or PSNR requires the decoding of encoded video (see flowchart above). Netflix encodes in H.265 format using encoding profiles that require a deblocking filter.[58] The deblocking filter is used during the encode and decode process within the H.265 (HEVC) Standard. Below, the gray boxes represent components that would be reused in a decoder:[59]

[58] "High efficiency video coding Recommendation ITU-T H.265 (02/2018)" at 185 ("H.265 (HEVC) Standard").

[59] Sullivan, *et al.*, *Overview of the High Efficiency Video Coding (HEVC) Standard*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, at 1651 (December 2012), *available at* http://iphome.hhi.de/wiegand/assets/pdfs/2012_12_IEEE-HEVC-Overview.pdf ("H.265 (HEVC) Overview").

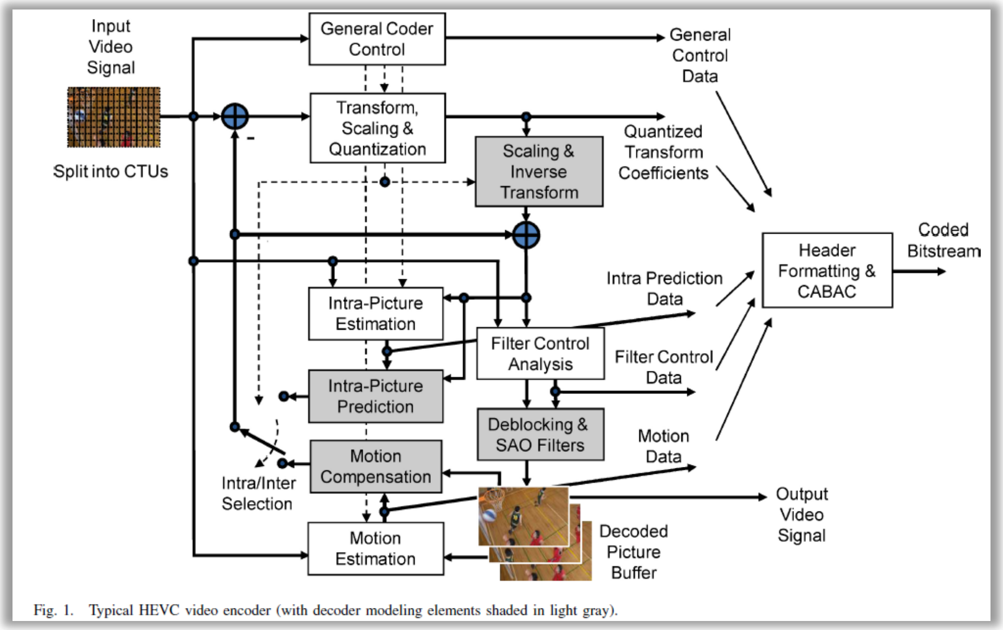


Fig. 1. Typical HEVC video encoder (with decoder modeling elements shaded in light gray).

More specifically, the encoding profile "main" within the H.265 (HEVC) Standard requires a deblocking filter.[60] The encoding of an H.265 (HEVC) video in general and the decoding of H.265 (HEVC) videos within Netflix's optimization loop practices the method of deblocking a reconstructed video frame.

171. Netflix "identif[ies] a boundary between two blocks of the reconstructed video frame." The H.265 (HEVC) Standard, used by Netflix to encode video in the H.265 (HEVC) format as just discussed, includes a deblocking filter as part of the encoder and decoder.[61]

[60] https://medium.com/netflix-techblog/performance-comparison-of-video-coding-standards-an-adaptive-streaming-perspective-d45d0183ca95.

[61] H.265 (HEVC) Overview at 1651.

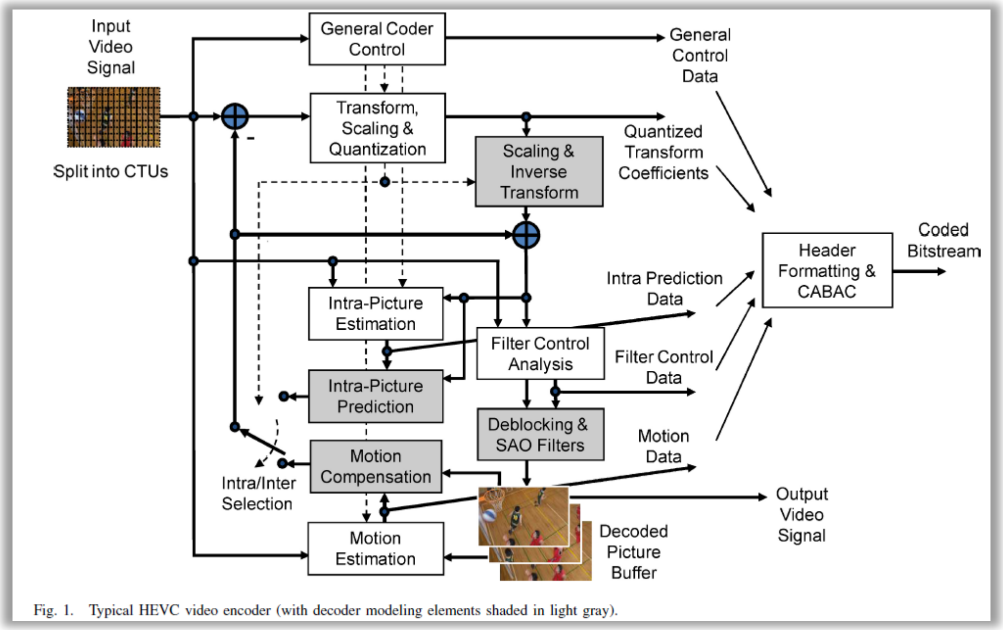


Fig. 1. Typical HEVC video encoder (with decoder modeling elements shaded in light gray).

The deblocking filter modifies a reconstructed video frame according to the deblocking filter process, including filtering the boundaries of the video frame.[62] The boundaries between blocks are determined as outlined in steps 4-5 of the deblocking filtering algorithm as specified in the H.265 (HEVC) Standard.[63]

[62] H.265 (HEVC) Standard at 185.

[63] *Id.* at 185-86; *see also* Norkin, *et al.*, *HEVC Deblocking Filter*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, at 1746-54 (December 2012), *available at* https://ieeexplore.ieee.org/abstract/document/6324414 ("H.265 (HEVC) Deblocking").

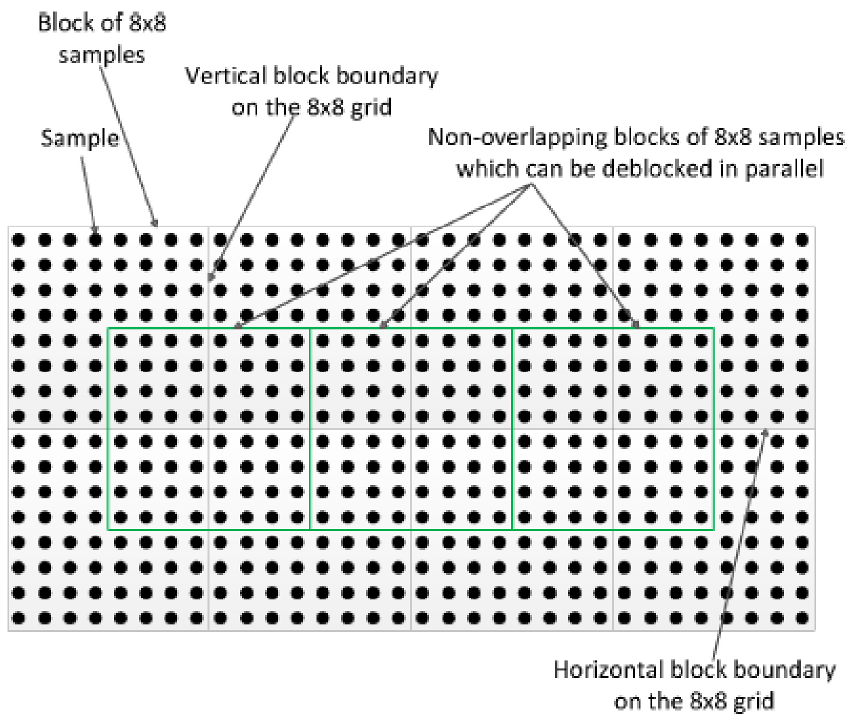
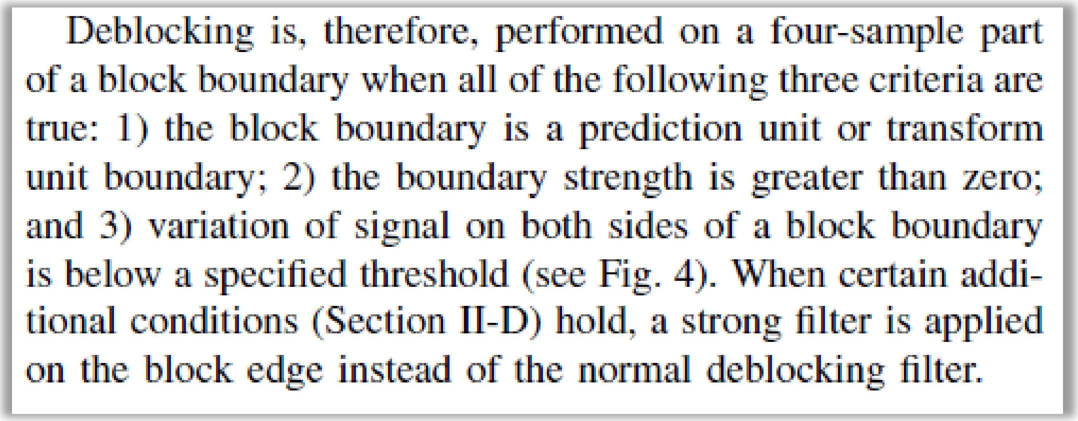

Block of 8x8 samples
Vertical block boundary on the 8x8 grid
Sample
Non-overlapping blocks of 8x8 samples, which can be deblocked in parallel
Horizontal block boundary on the 8x8 grid

Fig. 2. Illustration of picture samples and horizontal and vertical block boundaries on the 8 × 8 grid, and the nonoverlapping blocks of the 8 × 8 samples, which can be deblocked in parallel.


Deblocking is, therefore, performed on a four-sample part of a block boundary when all of the following three criteria are true: 1) the block boundary is a prediction unit or transform unit boundary; 2) the boundary strength is greater than zero; and 3) variation of signal on both sides of a block boundary is below a specified threshold (see Fig. 4). When certain additional conditions (Section II-D) hold, a strong filter is applied on the block edge instead of the normal deblocking filter.

Specifically, the H.265 (HEVC) encoder's and decoder's deblocking filter process identifies the boundary between two blocks of the reconstructed video frame.[64]

[64] H.265 (HEVC) Deblocking at 1747; *see also* H.265 (HEVC) Standard at 185 (Section 8.7.2.1).

172. Netflix "determin[es] the level of detail of the reconstructed video frame across a region in which the block boundary is located, wherein the region includes pixels from multiple rows and multiple columns of the reconstructed video frame that encompass pixels immediately adjacent to at least two sides of the block boundary and includes at least one pixel that is not immediately adjacent to the block boundary." The H.265 (HEVC) Standard requires a deblocking filter determining the level of detail by considering a region that includes pixels from multiple rows and multiple columns of the reconstructed video frame that encompass pixels immediately adjacent to at least two sides of the block boundary and at least one pixel not immediately adjacent to the block boundary. The boundary filtering strength, which contributes to the level of detail, is determined as outlined in step 6 of the deblocking filtering algorithm, as specified in the H.265 (HEVC) Standard.[65] The boundary filtering strength calculation first identifies whether to operate on a PU (prediction unit) boundary or TU (transform unit) boundary. Then the boundary filtering strength is determined, to decide whether to apply a strong deblocking filter or normal deblocking filter. If the boundary strength is greater than zero, then four conditions are also computed and checked as part of the level of detail to determine whether to apply a deblocking filter and whether to use the normal or strong version.[66] See images below. The four conditions are based on calculations from a region that includes pixels from multiple rows and multiple columns of the reconstructed video frame that encompass pixels immediately adjacent to at least two sides of the block boundary and includes at least one pixel that is not immediately adjacent to the block boundary.[67]

[65] H.265 (HEVC) Standard at 185-87.

[66] H.265 (HEVC) Deblocking at 1748-49.

[67] *Id.* at 1748.

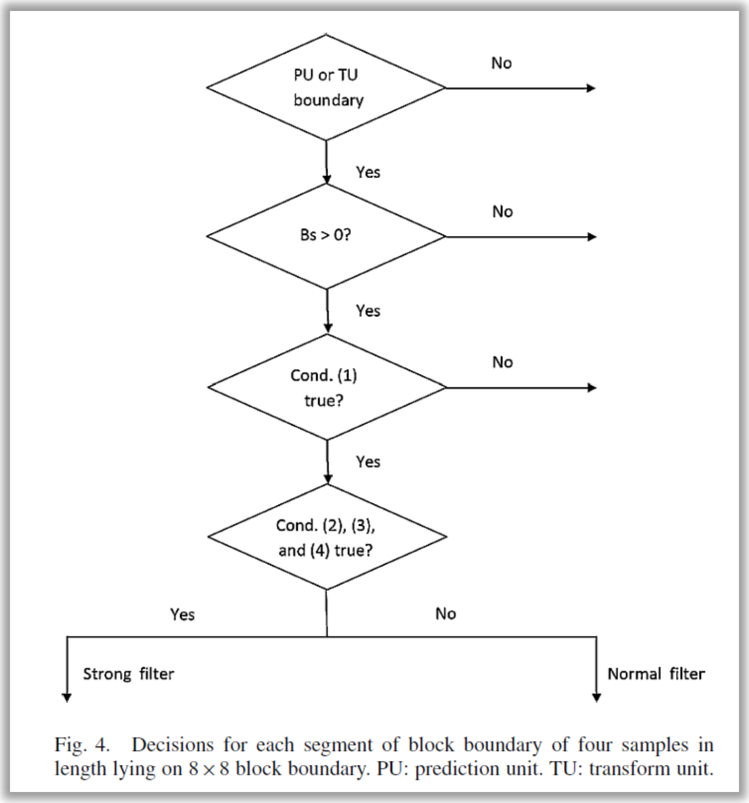


Fig. 4. Decisions for each segment of block boundary of four samples in length lying on 8 × 8 block boundary. PU: prediction unit. TU: transform unit.

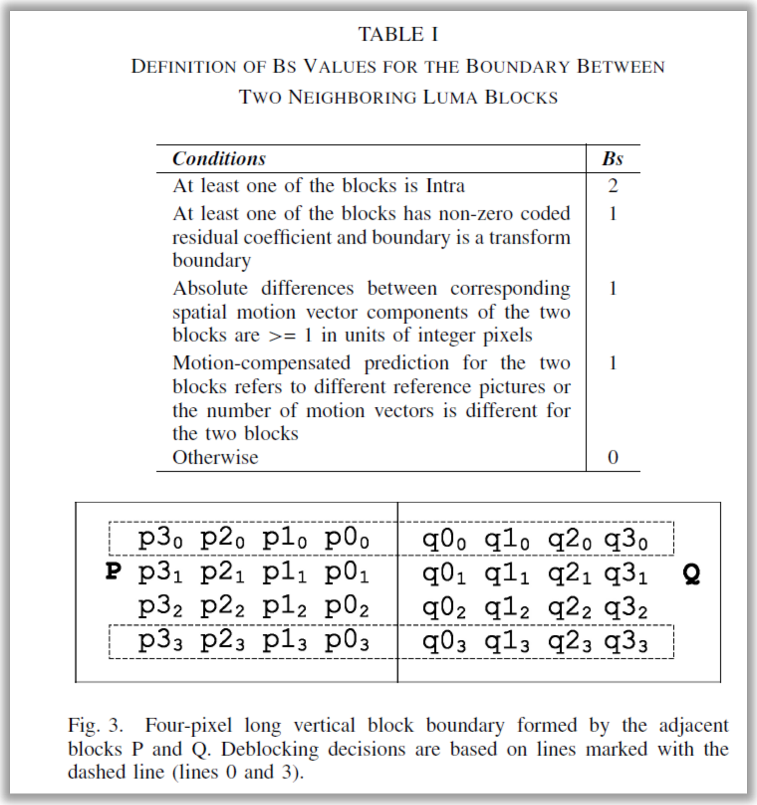


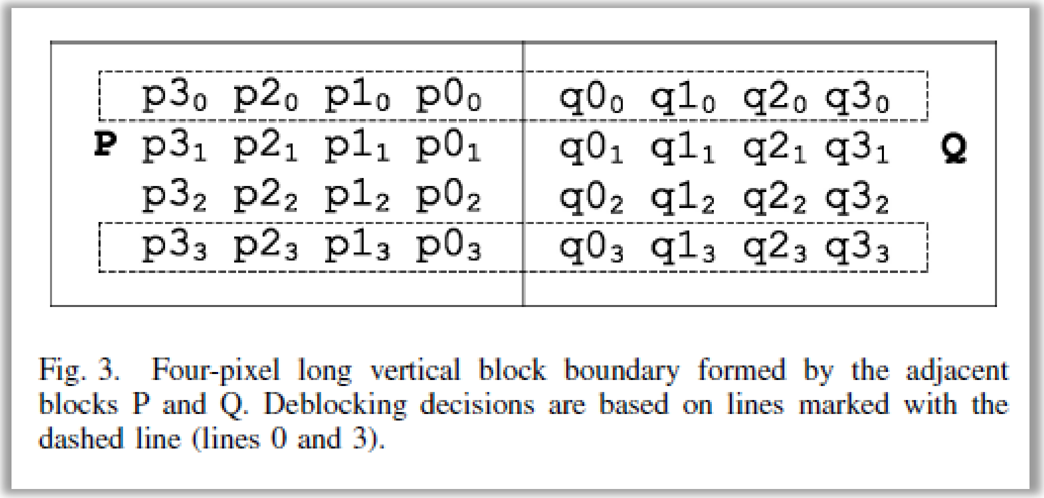
TABLE I
DEFINITION OF BS VALUES FOR THE BOUNDARY BETWEEN TWO NEIGHBORING LUMA BLOCKS

Conditions	*Bs*
At least one of the blocks is Intra	2
At least one of the blocks has non-zero coded residual coefficient and boundary is a transform boundary	1
Absolute differences between corresponding spatial motion vector components of the two blocks are >= 1 in units of integer pixels	1
Motion-compensated prediction for the two blocks refers to different reference pictures or the number of motion vectors is different for the two blocks	1
Otherwise	0

	$p3_0$	$p2_0$	$p1_0$	$p0_0$	$q0_0$	$q1_0$	$q2_0$	$q3_0$	
P	$p3_1$	$p2_1$	$p1_1$	$p0_1$	$q0_1$	$q1_1$	$q2_1$	$q3_1$	**Q**
	$p3_2$	$p2_2$	$p1_2$	$p0_2$	$q0_2$	$q1_2$	$q2_2$	$q3_2$	
	$p3_3$	$p2_3$	$p1_3$	$p0_3$	$q0_3$	$q1_3$	$q2_3$	$q3_3$	

Fig. 3. Four-pixel long vertical block boundary formed by the adjacent blocks P and Q. Deblocking decisions are based on lines marked with the dashed line (lines 0 and 3).

An illustration of the multiple rows (row 0 & 3) and multiple columns (cols 0 & 3) involved in such determination is shown below.[68] Column 0 pixels are immediately adjacent to at least two sides of the block boundary. Column 3 pixels are not.



	$p3_0$	$p2_0$	$p1_0$	$p0_0$	$q0_0$	$q1_0$	$q2_0$	$q3_0$	
P	$p3_1$	$p2_1$	$p1_1$	$p0_1$	$q0_1$	$q1_1$	$q2_1$	$q3_1$	**Q**
	$p3_2$	$p2_2$	$p1_2$	$p0_2$	$q0_2$	$q1_2$	$q2_2$	$q3_2$	
	$p3_3$	$p2_3$	$p1_3$	$p0_3$	$q0_3$	$q1_3$	$q2_3$	$q3_3$	

Fig. 3. Four-pixel long vertical block boundary formed by the adjacent blocks P and Q. Deblocking decisions are based on lines marked with the dashed line (lines 0 and 3).

173. Netflix "select[s] a filter to apply to predetermined pixels on either side of the block boundary based upon the determined level of detail" when the H.265 (HEVC) deblocking filter selects between the normal filter and the strong filter to apply to either side of the block boundary based upon the determined level of detail, for example, boundary strength and the four conditions.[69]

174. Netflix directly infringes at least claim 1, at least as described, when it tests its service using various playback devices.

175. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

[68] *Id.*

[69] *Id.* at 1749.

176. Netflix has infringed, and continues to infringe, at least claim 1 of the ’651 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’651 Infringing Products, in violation of 35 U.S.C. § 271(a).

177. Netflix has induced, and continues to induce, infringement of at least claim 1 of the ’651 patent, at least in the exemplary manner described in paragraphs 178-180, in violation of 35 U.S.C. § 271(b).

178. At least as of the date of this Complaint, Netflix knows that the ’651 patent allows its users to stream high-resolution content with smooth playback and with greater quality and efficiency. Specifically, as least as of the date of this Complaint, Netflix knows that the ’651 patent is directed to a multidimensional adaptive deblocking filter that allows for a higher-quality streaming video experience with more efficient compression and reduced bandwidth requirements.

179. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, practices the method recited in claim 1.

180. At least as of the date of this Complaint, Netflix knows and specifically intends that its end users practice the method recited in claim 1, when using its application and service as intended—namely, deblocking a reconstructed video frame, as described in paragraphs 168-171.

181. Netflix’s infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix’s wrongful acts in an amount subject to proof at trial.

COUNT III: INFRINGEMENT OF U.S. PATENT NO. 8,472,792

182. The allegations of paragraphs 1-181 of this Complaint are incorporated by reference as though fully set forth herein.

183. Pursuant to 35 U.S.C. § 282, the ’792 patent is presumed valid.

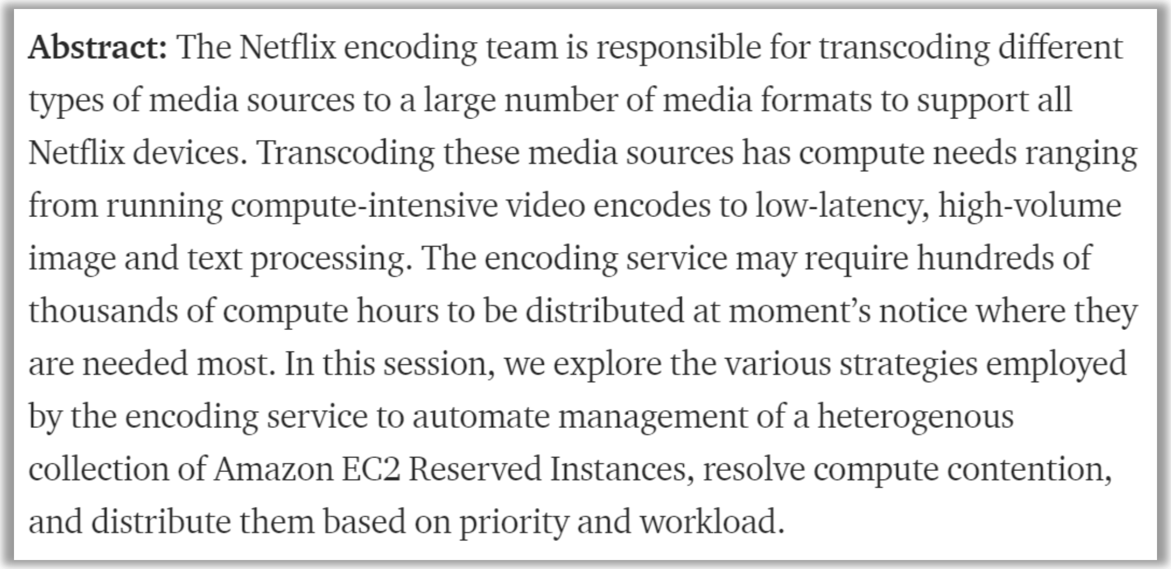
184. Upon information and belief, Netflix directly infringes the ’792 patent by making, using, offering to sell, selling, and/or importing into the United States

its Netflix service, which provides a multimedia distribution system (collectively, the "Accused '792 Infringing Products").

185. Upon information and belief, the Accused '792 Infringing Products directly infringe at least claim 9 of the '792 patent at least in the exemplary manner described in paragraphs 186-191 below.

186. Netflix provides an encoder for encoding a multimedia file, as '792 patent claim 9 recites.

187. Netflix's encoder comprises "a processor." Netflix encodes movies and other titles using computers with processors, as the Netflix Tech Blog confirms:[70]

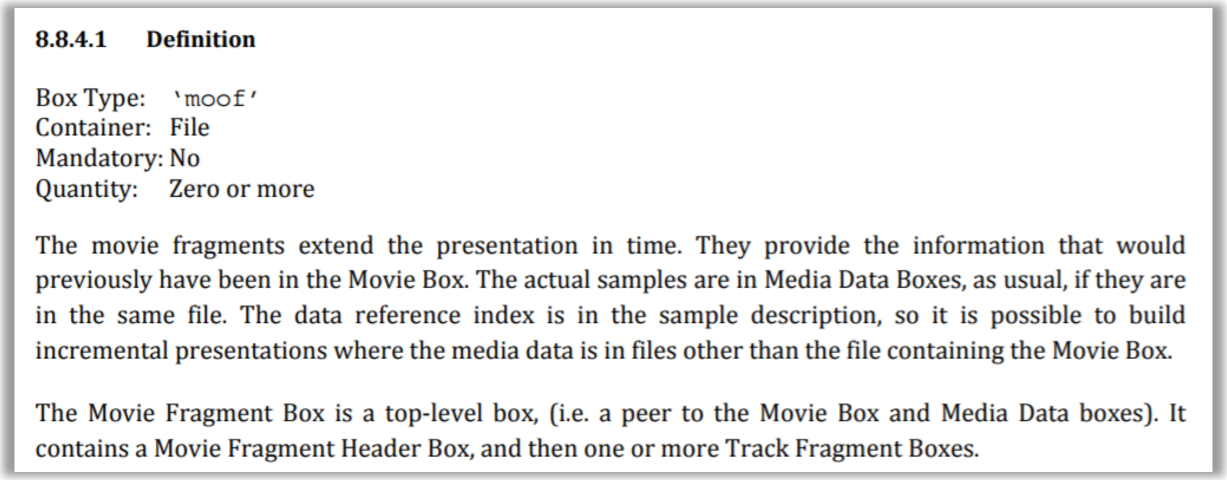


Abstract: The Netflix encoding team is responsible for transcoding different types of media sources to a large number of media formats to support all Netflix devices. Transcoding these media sources has compute needs ranging from running compute-intensive video encodes to low-latency, high-volume image and text processing. The encoding service may require hundreds of thousands of compute hours to be distributed at moment's notice where they are needed most. In this session, we explore the various strategies employed by the encoding service to automate management of a heterogenous collection of Amazon EC2 Reserved Instances, resolve compute contention, and distribute them based on priority and workload.

188. Netflix's encoder further comprises "a memory including a file containing at least one sequence of encoded video frames and a full index that includes information indicative of the location within the file and characteristics of each encoded video frame." Netflix produces multimedia files, such as MP4 files, with at least one sequence of encoded video frames stored in media data, or "mdat,"

[70] https://medium.com/netflix-techblog/netflix-at-aws-re-invent-2017-79384f525367.

boxes. Upon information and belief, Netflix multimedia streams contain mdat boxes. Netflix multimedia files (for example, MP4 files) also include at least one full index that includes information indicative of the location within the file and characteristics of each encoded video frame. A movie fragment box (or "moof") contains a number of track fragment, or "traf," boxes that each contain index information describing a sequence of video frames contained within an mdat box. A traf box contains size information for each track fragment. A traf box also contains a track run ("trun") box, which is a complete index to the location of each frame in the mdat box referenced by the traf box. The ISO/IEC definitions of the moof, traf, and trun boxes are below.[71]

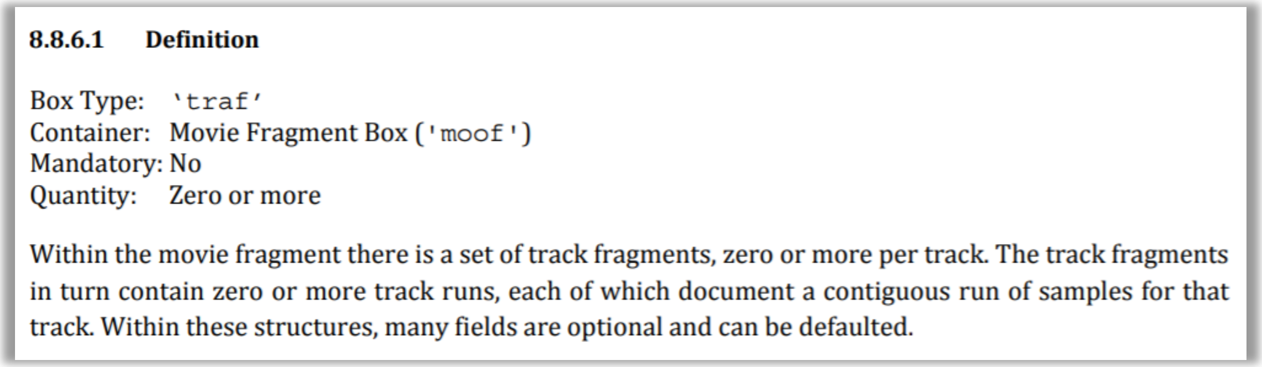


8.8.4.1 Definition

Box Type: 'moof'
Container: File
Mandatory: No
Quantity: Zero or more

The movie fragments extend the presentation in time. They provide the information that would previously have been in the Movie Box. The actual samples are in Media Data Boxes, as usual, if they are in the same file. The data reference index is in the sample description, so it is possible to build incremental presentations where the media data is in files other than the file containing the Movie Box.

The Movie Fragment Box is a top-level box, (i.e. a peer to the Movie Box and Media Data boxes). It contains a Movie Fragment Header Box, and then one or more Track Fragment Boxes.

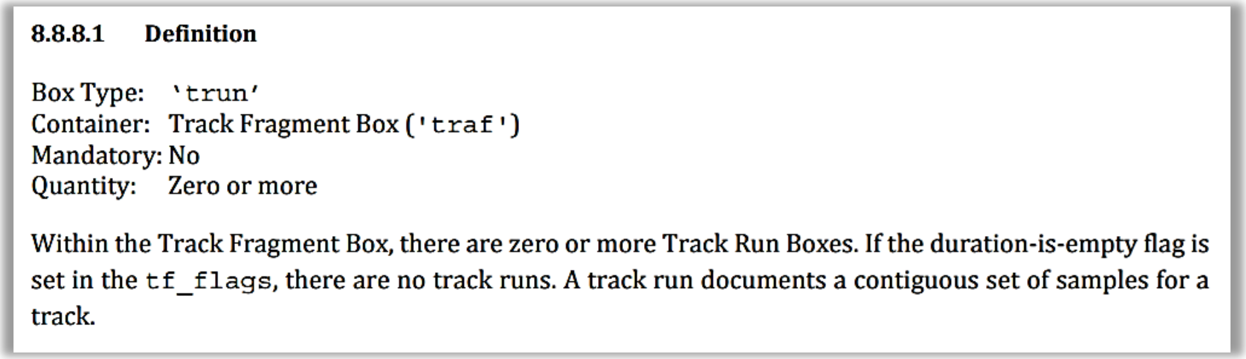


8.8.6.1 Definition

Box Type: 'traf'
Container: Movie Fragment Box ('moof')
Mandatory: No
Quantity: Zero or more

Within the movie fragment there is a set of track fragments, zero or more per track. The track fragments in turn contain zero or more track runs, each of which document a contiguous run of samples for that track. Within these structures, many fields are optional and can be defaulted.

[71] ISO/IEC 14496-12 at 56-58.



8.8.8.1 Definition

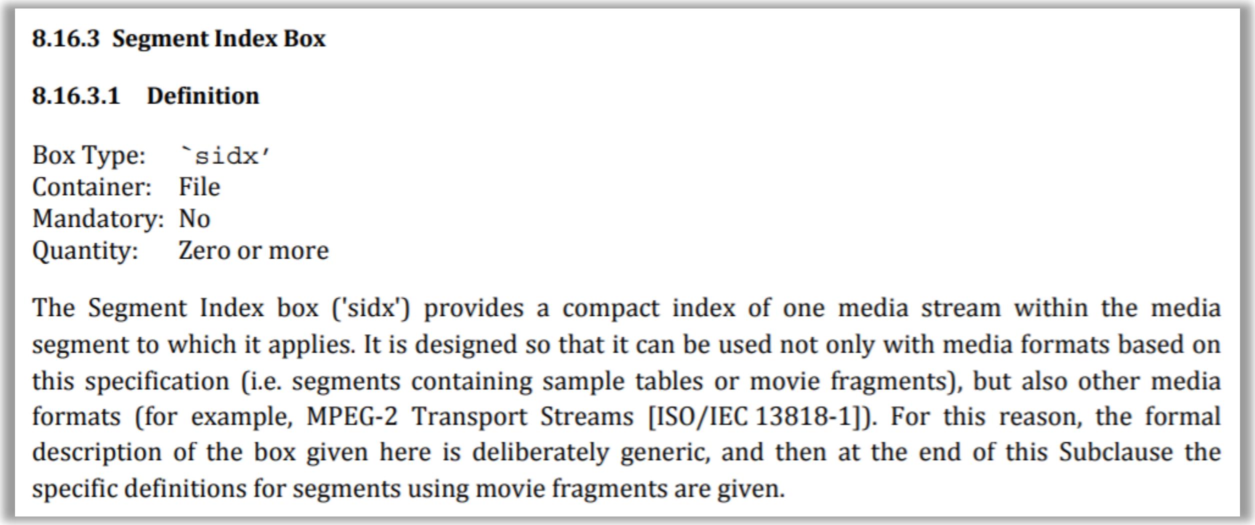
Box Type: 'trun'
Container: Track Fragment Box ('traf')
Mandatory: No
Quantity: Zero or more

Within the Track Fragment Box, there are zero or more Track Run Boxes. If the duration-is-empty flag is set in the tf_flags, there are no track runs. A track run documents a contiguous set of samples for a track.

The moof and mdat boxes are provided to the video assembler, which stores them in memory to process them. Upon information and belief, the Netflix video assembler builds a multimedia file, and it stores the file in memory containing all of the moof and mdat boxes, which collectively contain a sequence of encoded video frames and a full index including information indicative of the location within the file and characteristics of each encoded video frame.

189. Netflix's processor "is configured to generate an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames." Netflix multimedia files (for example, MP4 files) include an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames. The Netflix video contains a segment index box ("sidx") and a subsegment index box ("ssix"), either of which can be considered to be an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames. The sidx box is an index pointing to the location of each segment containing a moof box and a following mdat box, as shown below.[72]

[72] *Id.* at 105, 228.



8.16.3 Segment Index Box

8.16.3.1 Definition

Box Type: `sidx'
Container: File
Mandatory: No
Quantity: Zero or more

The Segment Index box ('sidx') provides a compact index of one media stream within the media segment to which it applies. It is designed so that it can be used not only with media formats based on this specification (i.e. segments containing sample tables or movie fragments), but also other media formats (for example, MPEG-2 Transport Streams [ISO/IEC 13818-1]). For this reason, the formal description of the box given here is deliberately generic, and then at the end of this Subclause the specific definitions for segments using movie fragments are given.

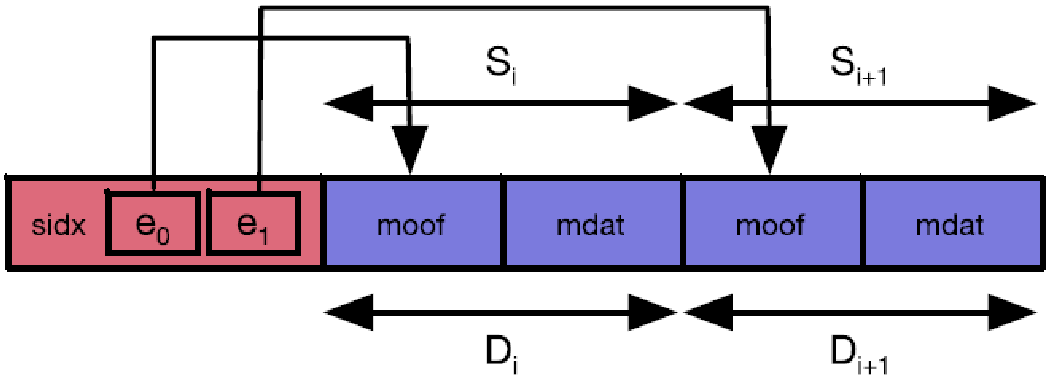


Figure K. 1: Simple Segment Index

The sequence of video frames in the mdat box contained within each of the video segments (namely, moof and following mdat box) pointed to by the abridged index contained within the sidx box is a subset of the encoded video frames contained within the sequence of encoded video frames contained within the file.[73] The ssix

[73] *Id.* at 56.

box also includes an abridged index that is an index to the locations of subsegments within segments. Upon information and belief, the ssix box includes a reference to the location of the second frame in the mdat box of each video segment (namely, moof and following mdat box), which is a subset of the sequence of encoded video frames contained within each of the individual video segments received by the video assembler.

190. Netflix's processor is further configured "to encode a multimedia file including the abridged index, the at least one sequence of encoded video frames, and a full index." The video assembler encodes a multimedia file (for example, an MP4 file) that contains the abridged index (either the sidx box or the ssix box, as described in the previous paragraph), the encoded video segments (moof and mdat boxes), and the full index (trun boxes).

191. Further, "the abridged index is located within the multimedia file prior to the series of encoded video frames, the first and second indexes enabling trick play functionality." Netflix multimedia files (for example, MP4 files) show that the abridged index is located within the multimedia file prior to the series of encoded video frames, and the multimedia file contains the first and second indexes that enable trick play functionality (for example, seeking). This is because each element in the trun, sidx, and ssix boxes enables a playback device to seek to an I-frame corresponding to a specific playback time. An I-frame is a single frame of digital content that an encoder encodes without reference to any other frames within the video sequence. The trun box, sidx box, and the ssix box enable trick play functionality because: (1) the sidx box is used to locate a video segment (namely, moof box and following mdat box) corresponding to a particular playback time that contains an I-frame, and (2) either the ssix box or the trun box within the moof box of the located video segment can be used to locate the first I-frame within the mdat box of the located video segment. The trun box can also be used in combination with the sidx box or ssix box to locate other frames within the mdat box.

Furthermore, the ssix box can be used to directly locate the first I-frame within a specific mdat box. Once the location of a frame is identified, individual frames of video from the mdat box can be extracted and provided to a decoder to commence playback at the new playback location.[74]

192. Netflix directly infringes at least claim 9, at least as described, when it tests its service using various playback devices.

193. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

194. Netflix has infringed, and continues to infringe, at least claim 9 of the ’792 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’792 Infringing Products, in violation of 35 U.S.C. § 271(a).

195. Netflix has induced, and continues to induce, infringement of at least claim 15 of the ’792 patent, at least in the exemplary manner described in paragraphs 196-203, in violation of 35 U.S.C. § 271(b).

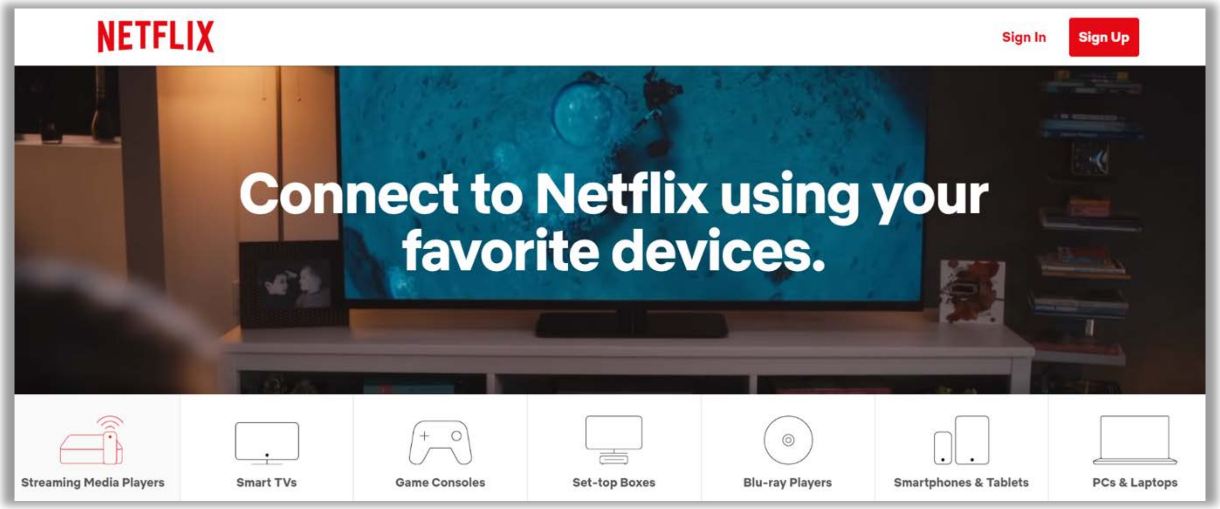
196. At least as of the date of this Complaint, Netflix knows that the ’792 patent enables playback features that video streaming users expect, enjoy, and use to navigate digital video easily, and they improve the user experience by reducing delays in loading and playing a video when it is selected by the user. Specifically, at least as of the date of this Complaint, Netflix knows that the ’792 patent is directed to providing an abridged video index that improves the user playback experience by enabling chunk-based adaptive bitrate streaming, “trick play,” and “fast start” functionality.

[74] *Id.* at 43, 59.

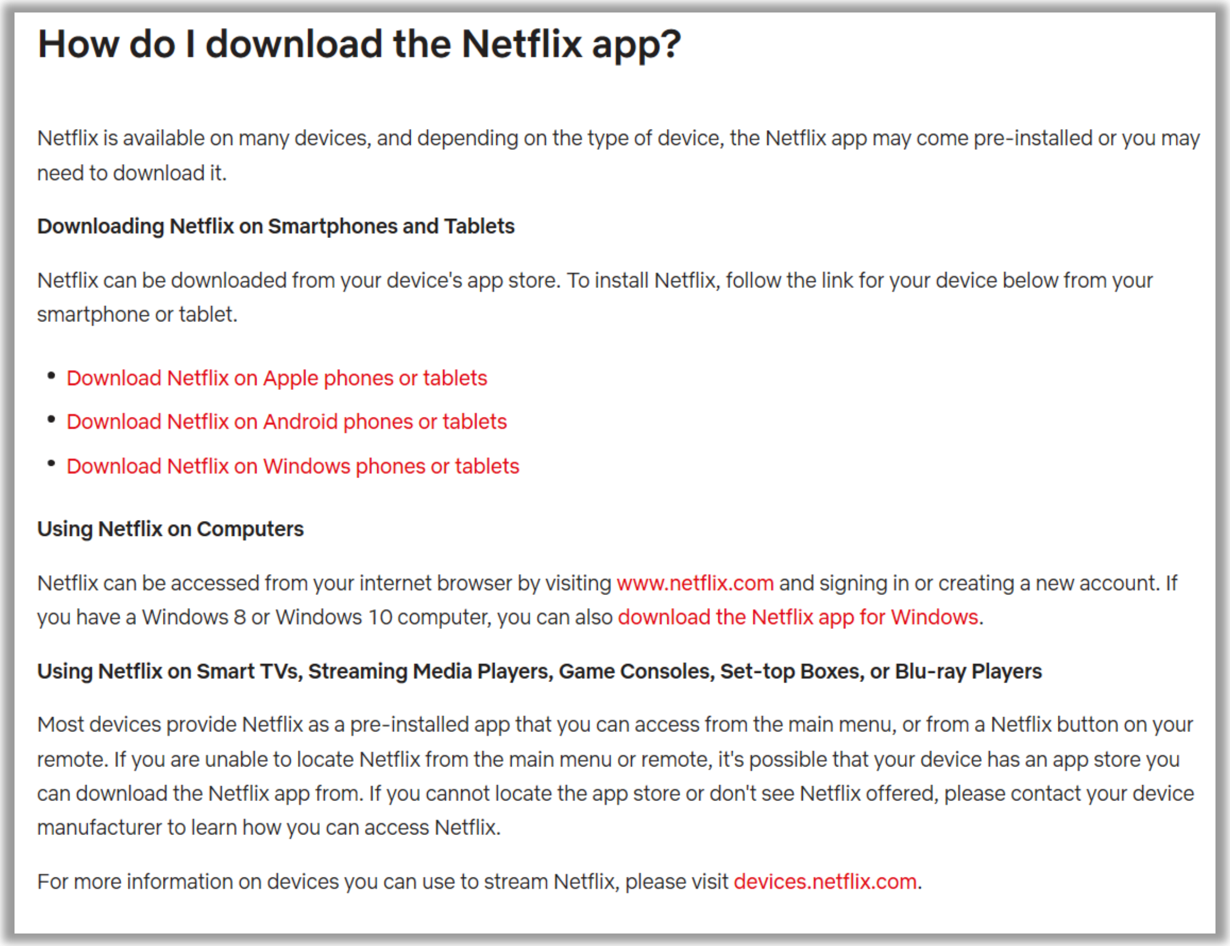
197. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, meets the limitations of claim 15.

198. At least as of the date of this Complaint, Netflix knows and specifically intends that CE playback devices enabling the Netflix application and service infringe claim 15, when enabling the application and service as intended—namely, the CE playback device serves as a decoder for decoding multimedia comprising at least one video track and at least one audio track.

199. The CE playback device enabling the Netflix application comprises "a processor configured to decode multimedia." The Netflix application runs on a device with a processor, and the application configures the processor to decode multimedia streamed from Netflix's server, as its website shows and instructs.[75]



[75] https://devices.netflix.com/en/; https://help.netflix.com/en/node/101653?ba=SwiftypeResultClick&q=install%20app%20browser.



How do I download the Netflix app?

Netflix is available on many devices, and depending on the type of device, the Netflix app may come pre-installed or you may need to download it.

Downloading Netflix on Smartphones and Tablets

Netflix can be downloaded from your device's app store. To install Netflix, follow the link for your device below from your smartphone or tablet.

- Download Netflix on Apple phones or tablets
- Download Netflix on Android phones or tablets
- Download Netflix on Windows phones or tablets

Using Netflix on Computers

Netflix can be accessed from your internet browser by visiting www.netflix.com and signing in or creating a new account. If you have a Windows 8 or Windows 10 computer, you can also download the Netflix app for Windows.

Using Netflix on Smart TVs, Streaming Media Players, Game Consoles, Set-top Boxes, or Blu-ray Players

Most devices provide Netflix as a pre-installed app that you can access from the main menu, or from a Netflix button on your remote. If you are unable to locate Netflix from the main menu or remote, it's possible that your device has an app store you can download the Netflix app from. If you cannot locate the app store or don't see Netflix offered, please contact your device manufacturer to learn how you can access Netflix.

For more information on devices you can use to stream Netflix, please visit devices.netflix.com.

200. The Netflix application, enabled on a CE playback device, configures the processor to decode multimedia “wherein the multimedia includes a sequence of encoded video frames.” The video that Netflix streams contains a sequence of encoded video frames. The video streamed from Netflix and stored at the decoder’s memory contains at least a series mdat boxes, which, as discussed, contain encoded video frames within a video fragment.[76] Upon information and belief, Netflix video streams contain mdat boxes.

201. The Netflix application, enabled on a CE playback device, configures the processor to decode multimedia wherein the multimedia further includes “a complete index referencing each encoded video frame in the sequence of encoded

[76] ISO/IEC 14496-12 at 57.

video frames." As explained, video streamed from Netflix contains moof boxes, which contain traf boxes. A traf box contains size information of each traf. The traf box also contains a trun box, which is a complete index to the location of each frame in the mdat box that follows the moof box containing the trun box.[77] Thus, Netflix video streams contain multiple sequences of encoded video frames and a complete index referencing each encoded video frame in the sequence of encoded video frames.

202. The Netflix application, enabled on a CE playback device, configures the processor to decode multimedia where the multimedia further includes "an abridged index referencing a subset of the encoded video frames in the sequence of encoded video frames." As explained, Netflix video contains an sidx box, which is an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames.

203. The Netflix application further configures the CE playback device's processor "to locate a particular encoded video frame within the multimedia using the abridged index and to playback the sequence of encoded video frame starting from the located encoded video frame, the first and second indexes enabling trick play functionality." As described in previous paragraphs, Netflix multimedia files (for example, MP4 files) show that the abridged index is located within the multimedia file before the series of encoded video frames, and the multimedia file contains the first and second indexes that enable trick play functionality (for example, seeking) because each element in trun, sidx, and ssix enables a playback device to seek to an I-frame corresponding to a specific playback time. The trun box in combination with the sidx box, and/or ssix box, enable trick play functionality, as already described.

[77] *Id.* at 56, 58.

204. Netflix's infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

COUNT IV: INFRINGEMENT OF U.S. PATENT NO. 9,184,920

205. The allegations of paragraphs 1-204 of this Complaint are incorporated by reference as though fully set forth herein.

206. Pursuant to 35 U.S.C. § 282, the '920 patent is presumed valid.

207. Upon information and belief, Netflix directly infringes the '920 patent by making, using, offering to sell, selling, and/or importing into the United States its Netflix service, which provides a federated digital rights management scheme including trusted systems (collectively, the "Accused '920 Infringing Products").

208. Upon information and belief, the Accused '920 Infringing Products directly infringe at least claim 1 of the '920 patent at least in the exemplary manner described in paragraphs 209-218 below.

209. Netflix provides a "method of decoding encrypted content using a playback device on which an active user encryption key is stored, where the content includes frames of video and at least a portion of a plurality of frames of video are encrypted using at least one frame encryption key, and the at least one frame encryption key is encrypted using a content encryption key, and one or more copies of the content encryption key are each encrypted using one or more user encryption keys including the active user encryption key." The video content distributed by Netflix in accordance with the MPEG-DASH Standard and the Microsoft PIFF Specification is encrypted by encrypting portions of frames using the AES-CTR cipher in accordance with the "cenc" scheme specified in the ISO Common Encryption Standard and Microsoft PIFF file format specification.[78] Due to

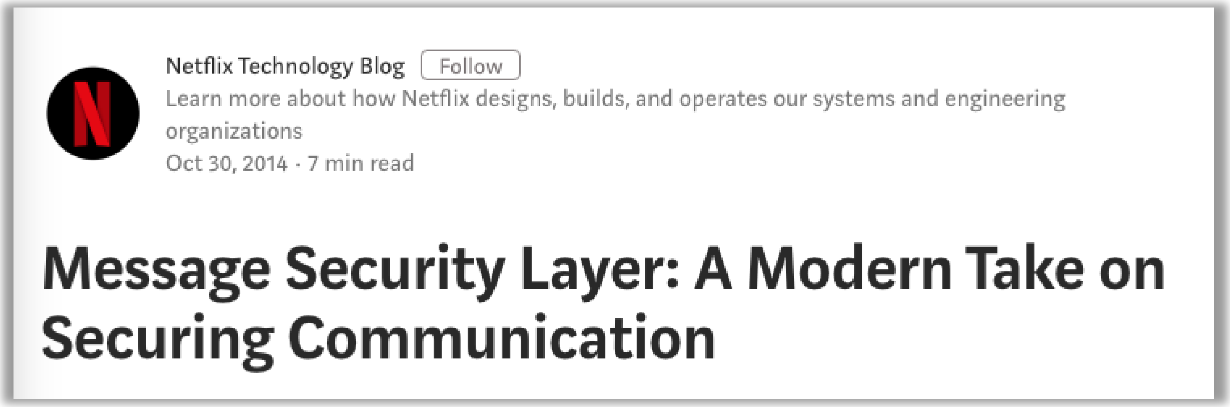
[78] *See* ISO/IEC 23009-1 (2014) Information technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and

Netflix's use of the "cenc" scheme to partially encrypt frames of video in accordance with the Microsoft PIFF Specification, playback of video streamed by Netflix on a playback device, using, for example, a Netflix-provided web-browser player or an application (for Android or iOS), involves decoding encrypted content. The process that Netflix uses to provide cryptographic keys to a playback device involves use of an active user encryption key stored on the playback device.

a. Specifically, Netflix has developed its own authentication process that involves the use of active user keys. The active user keys take the form of what Netflix describes as session keys contained in a Master Token, which become active when a user ID token is bound to the Master Token. To obtain a session key, the user must authenticate themselves to the Netflix servers. Following authentication, Master Token authentication session keys are used to encrypt and authenticate messages. Netflix's authentication process is described within the Message Security Layer ("MSL") in, for example, the Netflix Tech Blog:[79]

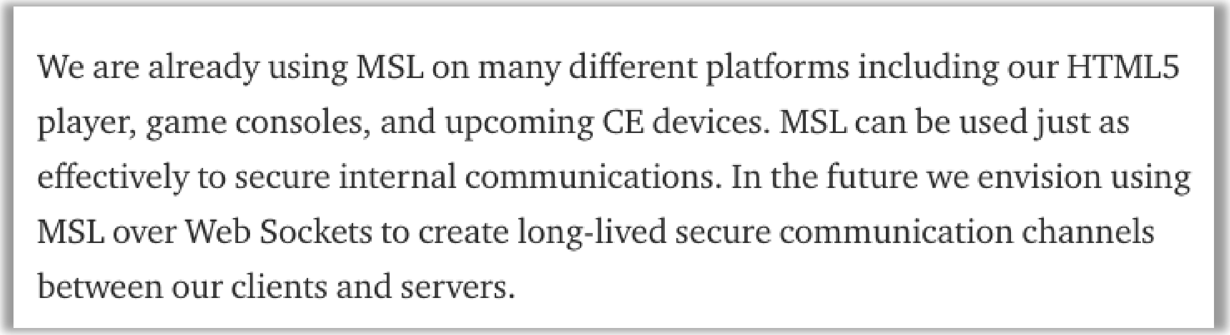
segment formats; ISO/IEC 23007-1 (2016) Information technology—MPEG systems technologies—Part 7: Common encryption in ISO base media file format files; and Portable encoding of audio-video objects: The Protected Interoperable File Format (PIFF).

[79] https://medium.com/netflix-techblog/message-security-layer-a-modern-take-on-securing-communication-f16964b79642.



Netflix Technology Blog Follow
Learn more about how Netflix designs, builds, and operates our systems and engineering organizations
Oct 30, 2014 · 7 min read

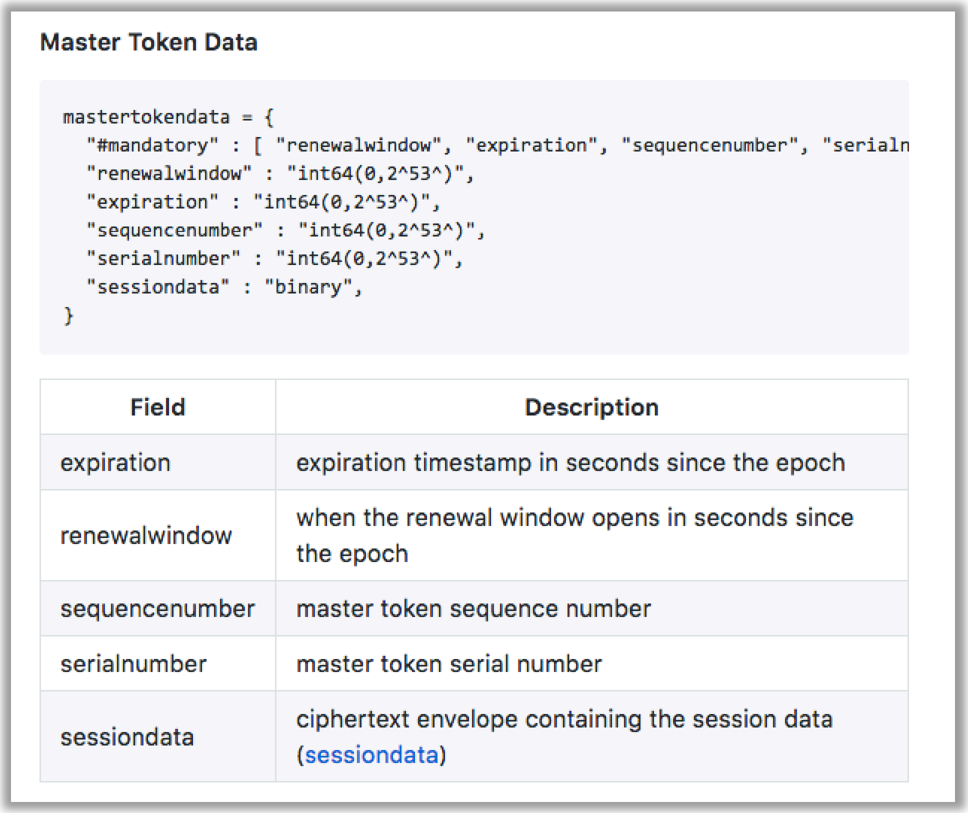
Message Security Layer: A Modern Take on Securing Communication



We are already using MSL on many different platforms including our HTML5 player, game consoles, and upcoming CE devices. MSL can be used just as effectively to secure internal communications. In the future we envision using MSL over Web Sockets to create long-lived secure communication channels between our clients and servers.

b. The Netflix MSL protocol is documented via an Open Source repository hosted on GitHub, and, in accordance with Netflix's documentation, the Master Tokens are structured as follows:[80]

[80] https://github.com/Netflix/msl/wiki/Entity-Authentication#master-tokens.



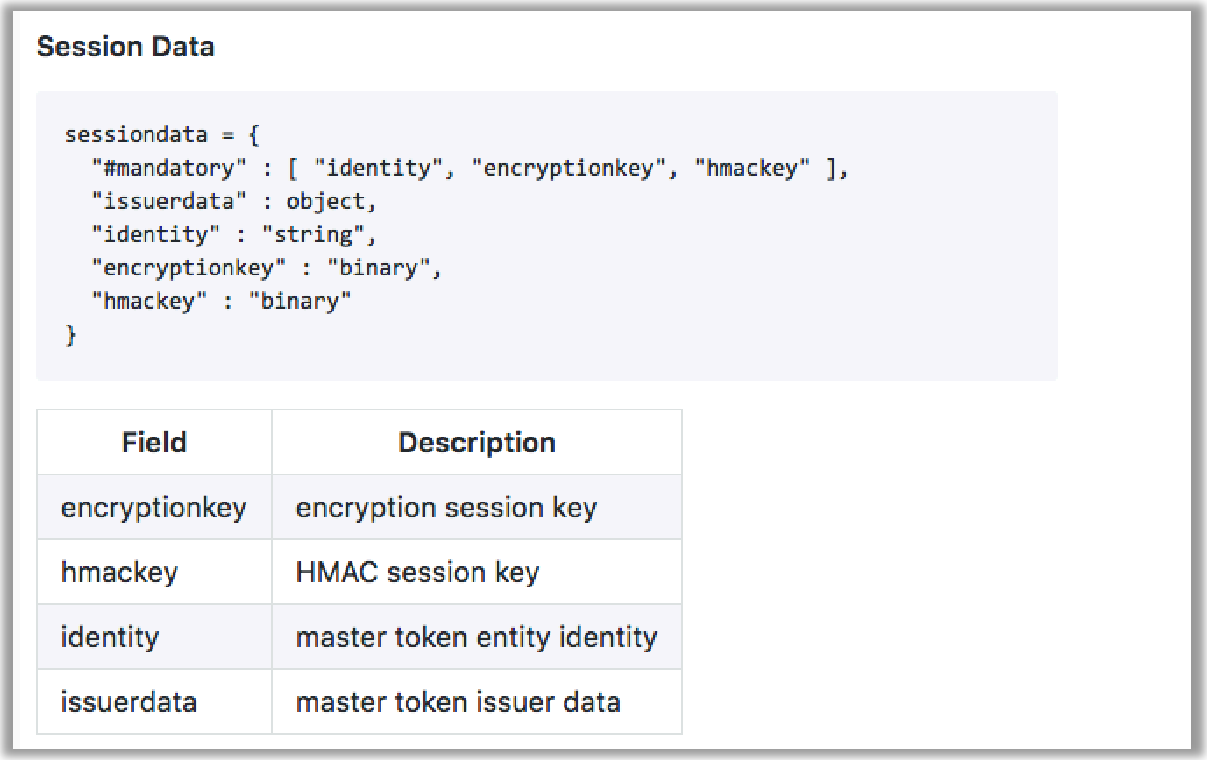
Master Token Data

```
mastertokendata = {
  "#mandatory" : [ "renewalwindow", "expiration", "sequencenumber", "serialn
  "renewalwindow" : "int64(0,2^53^)",
  "expiration" : "int64(0,2^53^)",
  "sequencenumber" : "int64(0,2^53^)",
  "serialnumber" : "int64(0,2^53^)",
  "sessiondata" : "binary",
}
```

Field	Description
expiration	expiration timestamp in seconds since the epoch
renewalwindow	when the renewal window opens in seconds since the epoch
sequencenumber	master token sequence number
serialnumber	master token serial number
sessiondata	ciphertext envelope containing the session data (sessiondata)

c. The Session Data is contained within the Master Token and is encrypted using “secret keys” and is as follows:[81]

[81] *Id.*



Session Data

```
sessiondata = {
  "#mandatory" : [ "identity", "encryptionkey", "hmackey" ],
  "issuerdata" : object,
  "identity" : "string",
  "encryptionkey" : "binary",
  "hmackey" : "binary"
}
```

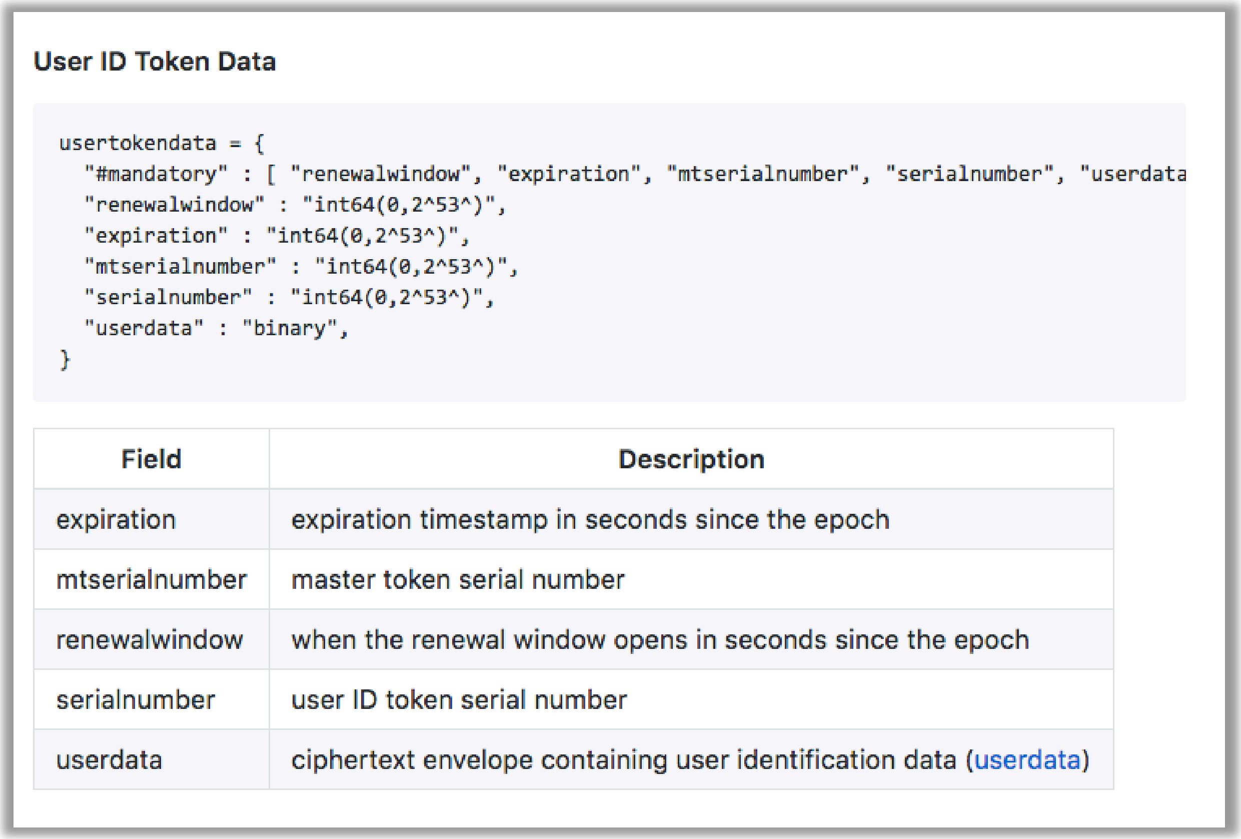
Field	Description
encryptionkey	encryption session key
hmackey	HMAC session key
identity	master token entity identity
issuerdata	master token issuer data

d. Netflix also indicates that "[o]ther entities cannot decrypt the master token session data or generate the master token verification data unless they also have access to these [secret] keys. These secret keys must be adequately protected as unauthorized access to these keys would allow communication involving master tokens to be compromised."[82]

e. Where a user login and password is not explicitly requested by the Netflix player, authentication is achieved using a user ID token. User ID token data is as follows:[83]

[82] *Id.*

[83] https://github.com/Netflix/msl/wiki/User-Authentication.



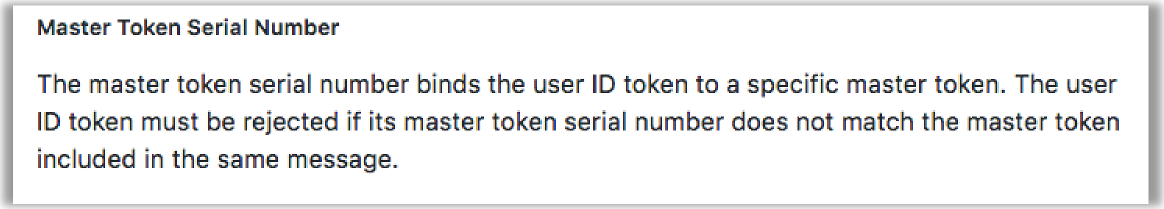
User ID Token Data

```
usertokendata = {
  "#mandatory" : [ "renewalwindow", "expiration", "mtserialnumber", "serialnumber", "userdata
  "renewalwindow" : "int64(0,2^53^)",
  "expiration" : "int64(0,2^53^)",
  "mtserialnumber" : "int64(0,2^53^)",
  "serialnumber" : "int64(0,2^53^)",
  "userdata" : "binary",
}
```

Field	Description
expiration	expiration timestamp in seconds since the epoch
mtserialnumber	master token serial number
renewalwindow	when the renewal window opens in seconds since the epoch
serialnumber	user ID token serial number
userdata	ciphertext envelope containing user identification data (userdata)

f. The Netflix MSL specifies that "[s]ecret keys are used by the issuing entity to encrypt the user ID token user data and generate the user ID token verification data. Other entities cannot decrypt the user ID token user data or generate the user ID token verification data unless they also have access to these keys. These secret keys should be adequately protected to prevent unauthorized access to the user identity." Moreover, the master token serial number binds the user ID token to the master ID token used to contain the session keys:[84]

[84] *Id.*

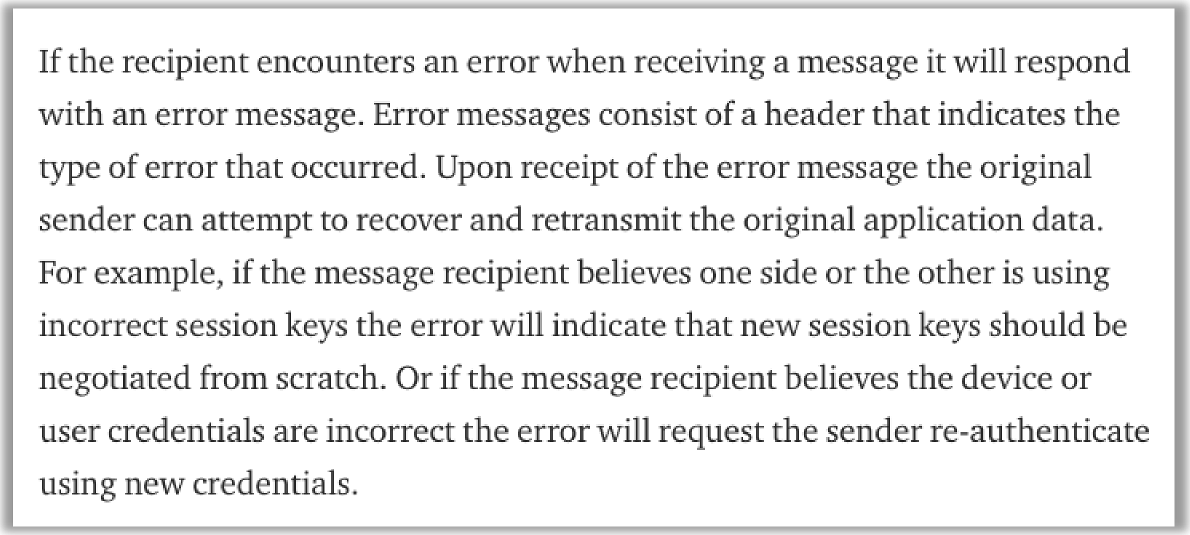


Master Token Serial Number

The master token serial number binds the user ID token to a specific master token. The user ID token must be rejected if its master token serial number does not match the master token included in the same message.

g. Accordingly, the session keys are bound to a specific user ID, and a failure of the user ID token to authenticate will cause suspension of communication via the session keys (in other words, the server will not authenticate the user). In this way, the session keys contained within the Master Token and stored by the playback device constitute an active user key that is stored by the playback device. In the event that the user ID token fails to authenticate, the Master Token session key is no longer active. When a new user ID token is bound to the Master Token, or a new Master Token is issued and bound to a user ID token, then the session key becomes an active user key again. The process is described in the Netflix Tech Blog post as follows:[85]

[85] https://medium.com/netflix-techblog/message-security-layer-a-modern-take-on-securing-communication-f16964b79642.



If the recipient encounters an error when receiving a message it will respond with an error message. Error messages consist of a header that indicates the type of error that occurred. Upon receipt of the error message the original sender can attempt to recover and retransmit the original application data. For example, if the message recipient believes one side or the other is using incorrect session keys the error will indicate that new session keys should be negotiated from scratch. Or if the message recipient believes the device or user credentials are incorrect the error will request the sender re-authenticate using new credentials.

h. The content received from Netflix's servers includes encoded "frames of video." The player receives a portion of an MP4 file from Netflix that includes an mdat box (namely, at least one video track encoded as a plurality of video chunks). Irrespective of whether the content is encoded using the H.264, H.265, or VP9 codecs, the content is stored in an MP4 container file formatted in accordance with the Microsoft PIFF Specification. VP9 content is also stored in accordance with an additional specification document published by the open source WebM project regarding the storage of VP9 content in the ISO BMFF.[86]

i. In addition, "at least a portion of a plurality of frames of video are encrypted using at least one frame encryption key." Irrespective of the codec used to encode the video, portions of each frame in the streams encoded by Netflix for delivery via MPEG-DASH are encrypted using the Advanced Encryption Standard Counter (AES-CTR) mode encryption cipher in accordance with the "cenc"

[86] *See* VP9 in ISO Media File Format, https://www.webmproject.org/vp9/mp4/.

scheme in the ISO Common Encryption Standard and the Microsoft PIFF Specification.[87] Accordingly, each frame in the received multimedia file is encrypted using at least one frame key.

j. Further, "the at least one frame encryption key is encrypted using a content encryption key." Files encoded by Netflix for distribution via MPEG-DASH use a frame encryption key (namely, a key stream output by the AES-CTR cipher) that is encrypted by a content encryption key (namely, a key indicated by the KID in the PIFF Track Encryption Box). The frame encryption key is decrypted by configuring an AES cipher using the key indicated by a KID in the PIFF Track Encryption Box and providing an initialization vector to the AES cipher in AES-CTR mode to obtain a decrypted frame key (namely, the key stream).

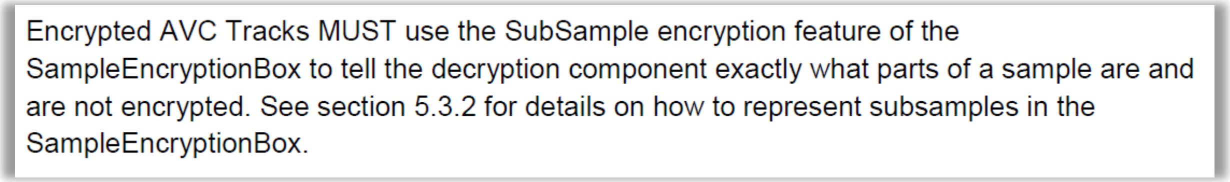
k. Netflix further provides that "one or more copies of the content encryption key are each encrypted using one or more user encryption keys including the active user encryption key." The Netflix application and browser-based players that stream H.265, H.264, and VP9 video using MPEG-DASH obtain a copy of the content encryption key from a DRM server (such as Microsoft Playready, Google Widevine, or Apple FairPlay) to play back the encrypted streams. Information that can be used to request the content encryption key from a DRM server is contained in different Protection System Header Boxes. The request and responses to

[87] *See* PIFF Specification, page 17 ("AlgorithmID . . . 0x1 –AES 128-bit in CTR mode"); VP9 ISO BMFF Specification ("If the VP9 data is encrypted, the Protection Scheme Info box ('sinf') SHALL be present, and SHALL contain a Scheme Type ('schm') box. The scheme_type field of the 'schm' box SHALL be 'cenc', indicating that AES-CTR encryption is used when samples are encrypted.").

obtain a copy of the content encryption key are communicated via MSL and, as a result, are, upon information and belief, encrypted by a user encryption key.

210. Netflix "obtain[s] encrypted content using a playback device, where the content includes frames of video and at least a portion of a plurality of frames of video are encrypted using at least one frame encryption key" when it receives, for example, content from its servers that includes encoded frames of video stored in mdat boxes within an MP4 file. Irrespective of whether the content is encoded using the H.264, H.265, or VP9 codecs, the content is stored in an MP4 file formatted in accordance with the Microsoft PIFF Specification. VP9 content is also stored in accordance with an additional specification document published by the open source WebM project regarding the storage of VP9 content in the ISO BMFF.

a. With respect to, for example, content encoded using the H.264 codec, Netflix obtains streams of video that are identified as encrypted by a PIFF Track Encryption Box. The Microsoft PIFF Specification specifies that sample encryption must be used when using the AES-CTR cipher so that the file contains information that tells the player exactly which parts of the sample are and are not encrypted:[88]



Encrypted AVC Tracks MUST use the SubSample encryption feature of the SampleEncryptionBox to tell the decryption component exactly what parts of a sample are and are not encrypted. See section 5.3.2 for details on how to represent subsamples in the SampleEncryptionBox.

b. Upon information and belief, Netflix provides initialization vectors in the PIFF Sample Encryption Box "uuid." The fields following the initialization vectors for encoded frames indicate that at least a

[88] PIFF Specification, page 23.

> portion of the frame is encrypted using at least one frame encryption key. Upon information and belief, Netflix provides similar PIFF Sample Encryption Boxes in MP4 container files used to stream H.265, H.264, and VP9 content by Netflix to playback devices.

211. Netflix "obtain[s] using the playback device a copy of the at least one frame encryption key that is encrypted using a content encryption key and obtaining one or more copies of the content encryption key that are each encrypted using one or more user encryption keys including an active user encryption key stored on the playback device." As noted above, files encoded by Netflix use a frame encryption key (namely, a key stream output by the AES-CTR based on the initialization vector and the content encryption key, namely, the key indicated by the KID in the PIFF Track Encryption Box). A decrypted frame key is obtained by providing an initialization vector (from the PIFF Sample Encryption Box) to an AES-CTR cipher configured using the content encryption key indicated by the KID (from the PIFF Track Encryption Box). The key stream output by the AES-CTR is the frame key used to decrypt one or more encrypted portions of a frame of video. The Netflix application and browser-based players can obtain a copy of the content encryption key from different DRM servers using different DRM headers contained within the multimedia file (Protection System Header Boxes). To obtain the content encryption key, the Netflix application and browser-based players communicate with the relevant DRM server. Upon information and belief, the proprietary means by which the respective DRM systems distribute the content encryption key further includes an active user key. As noted above, Netflix implements the Netflix MSL in a manner that relies on an active user key stored on the playback device to encrypt messages. Therefore, the content encryption key returned by Netflix's DRM servers is encrypted in a manner that enables decryption using the active user key stored on the playback device.

212. Netflix "decrypt[s] one of the one or more copies of the content encryption key using the playback device and the active user encryption key" by decrypting MSL message data (the content encryption key) received from the Netflix DRM servers using the active user encryption key. The content encryption key is then available along with the initialization vectors to obtain frame encryption keys in the manner described above, enabling decoding and playback of the encrypted video stream.

213. Netflix "play[s] back frames of the encrypted content using the playback device" by displaying decoded frames via the Netflix player.

214. Netflix's playback comprises "identifying any portions of a frame that are encrypted." As required by, for example, the Microsoft PIFF Specification, the frames are partially encrypted using subsample encryption, and the player software identifies the portions of the frames that are encrypted using the PIFF Sample Encryption Box contained in every MP4 video segment. Each frame has, for example, a corresponding PIFF Sub Sample Encryption Entry contained within the PIFF Sample Encryption Box that specifies an initialization vector, the number of subsamples that are encrypted, and the number of encrypted/unencrypted bytes in each encrypted subsample.

215. Netflix's playback further comprises "identifying the frame encryption key used to encrypt the identified portions of the frame." The frame encryption key for each frame is identified from the PIFF Sample Encryption Entry based on the initialization vector for the frame and is decrypted using the initialization vector and the content encryption key, the key indicated by a KID in the PIFF Track Encryption Box.

216. Netflix's playback further comprises "decrypting the identified frame encryption key using the decrypted content encryption key," because, as noted above, the ISO Common Encryption Standard specifies that the key stream output by the AES-CTR is the frame encryption key used to decrypt one or more

encrypted portions of a frame of video. The process of decrypting the identified frame encryption key involves configuring an AES-CTR cipher using the content encryption key, the key indicated by a KID in the PIFF Track Encryption Box, and providing the initialization vector from the PIFF Sample Encryption Entry to the AES-CTR cipher to obtain the frame key.

217. Netflix's playback further comprises "decrypting the encrypted portions of the frame using the decrypted identified frame encryption key" because, as noted above, the content was encrypted and must use a frame key for decryption in accordance with the ISO Common Encryption Standard.

218. Netflix's playback further comprises "decoding the unencrypted frame of video," when it plays decoded video via the Netflix player.

219. Netflix directly infringes at least claim 1, at least as described, when it tests its service using various playback devices.

220. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

221. Netflix has infringed, and continues to infringe, at least claim 1 of the '920 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused '920 Infringing Products, in violation of 35 U.S.C. § 271(a).

222. Netflix has induced, and continues to induce, infringement of at least claim 1 of the '920 patent, at least in the exemplary manner described in paragraphs 223-225, in violation of 35 U.S.C. § 271(b).

223. At least as of the date of this Complaint, Netflix knows that the '920 patent allows it to deliver video content securely to many different devices, supporting a large and diverse streaming device ecosystem. Specifically, the content security provided by the '920 inventions allows Netflix to obtain and offer

its users a library of high-quality video content. At least as of the date of this Complaint, Netflix knows that the ’920 patent is directed to a DRM architecture that enhances content security by binding active encryption keys to a user, allowing secure streaming.

224. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, practices the method recited in claim 1 of the ’920 patent.

225. At least as of the date of this Complaint, Netflix knows and specifically intends that its end users practice the method recited in claim 1, when using its application and service as intended—namely, the user engages the Netflix application to decode and play back encrypted digital video content using the playback device, as described in paragraphs 209-218.

226. Netflix’s infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix’s wrongful acts in an amount subject to proof at trial.

COUNT V: INFRINGEMENT OF U.S. PATENT NO. 9,270,720

227. The allegations of paragraphs 1-226 of this Complaint are incorporated by reference as though fully set forth herein.

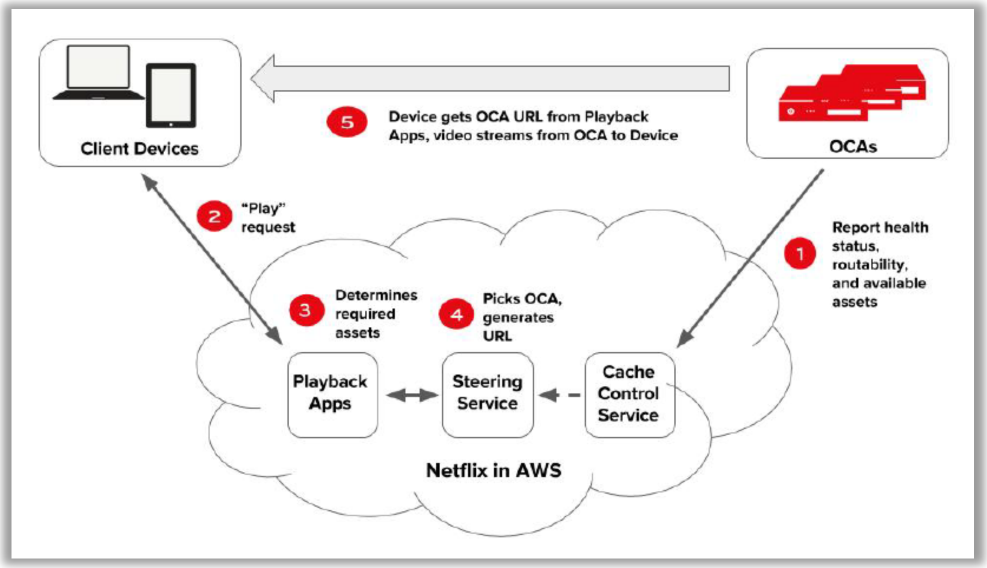
228. Pursuant to 35 U.S.C. § 282, the ’720 patent is presumed valid.

229. Upon information and belief, Netflix directly infringes the ’720 patent by making, using, offering to sell, selling, and/or importing into the United States its Netflix service, which provides a system and method for automatically generating top level index files (collectively, the “Accused ’720 Infringing Products”).

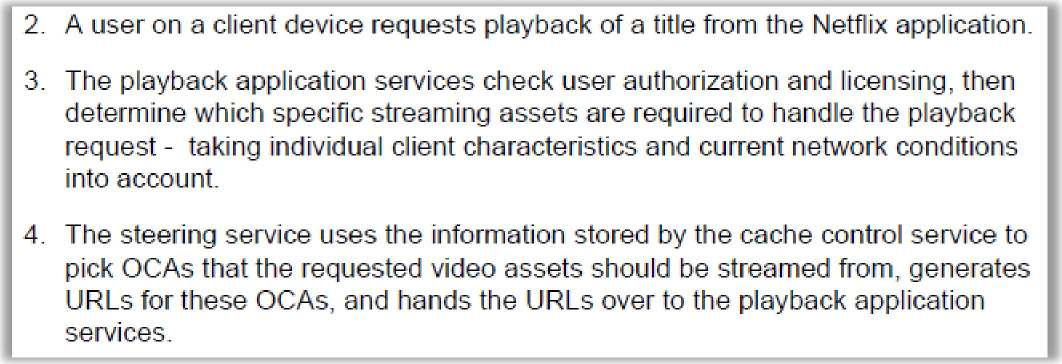
230. Upon information and belief, the Accused ’720 Infringing Products directly infringe at least claim 1 of the ’720 patent at least in the exemplary manner described in paragraphs 231-236 below.

231. Netflix practices a "method of generating a top level index file," that is, a manifest.

232. Netflix "receiv[es] a request from a playback device at a playback server system, where the request (i) identifies a piece of content and (ii) includes a product identifier" when its streaming infrastructure, that is, its playback server system, receives a request from a CE playback device, where the request (i) identifies a piece of content, and (ii) includes a product identifier. As illustrated in, for example, Netflix Open Connect documentation, which describes "the global network that is responsible for delivering Netflix TV shows and movies to our members worldwide," Netflix receives a request from a playback device that identifies requested video assets and "individual client characteristics."[89]


Client Devices
5
Device gets OCA URL from Playback Apps, video streams from OCA to Device
OCAs
2
"Play" request
1
Report health status, routability, and available assets
3
Determines required assets
4
Picks OCA, generates URL
Playback Apps
Steering Service
Cache Control Service
Netflix in AWS

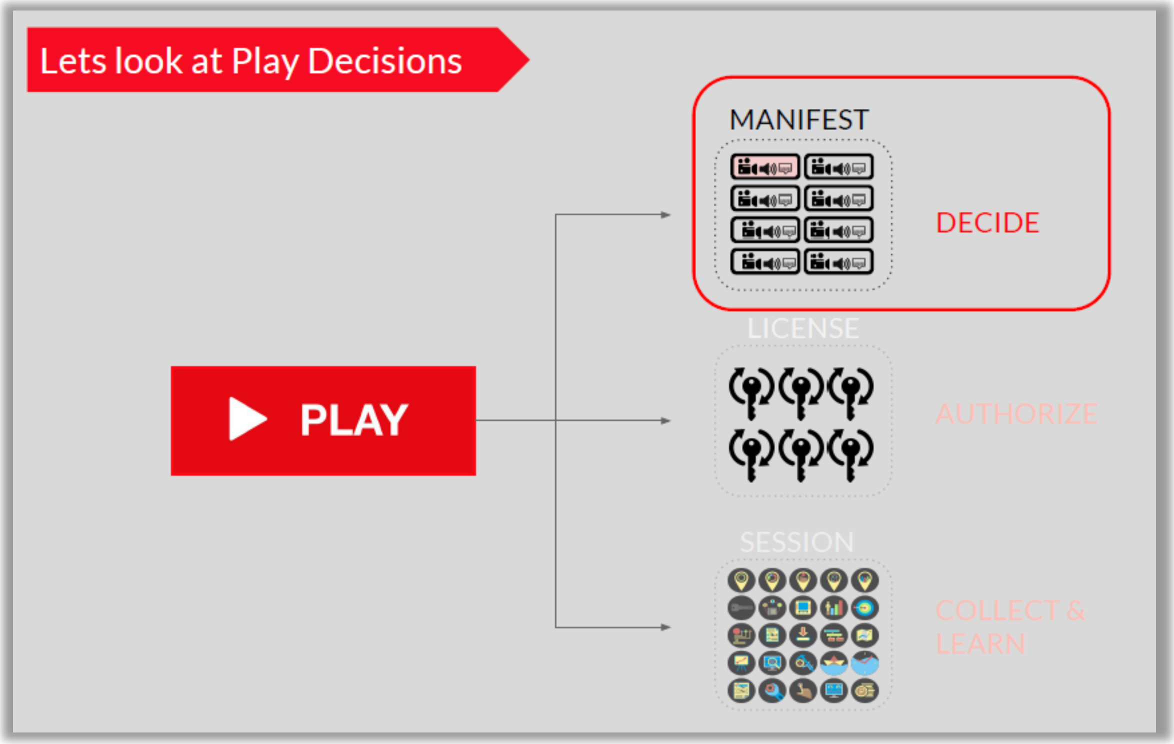
[89] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 2, 4.



2. A user on a client device requests playback of a title from the Netflix application.
3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and current network conditions into account.
4. The steering service uses the information stored by the cache control service to pick OCAs that the requested video assets should be streamed from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

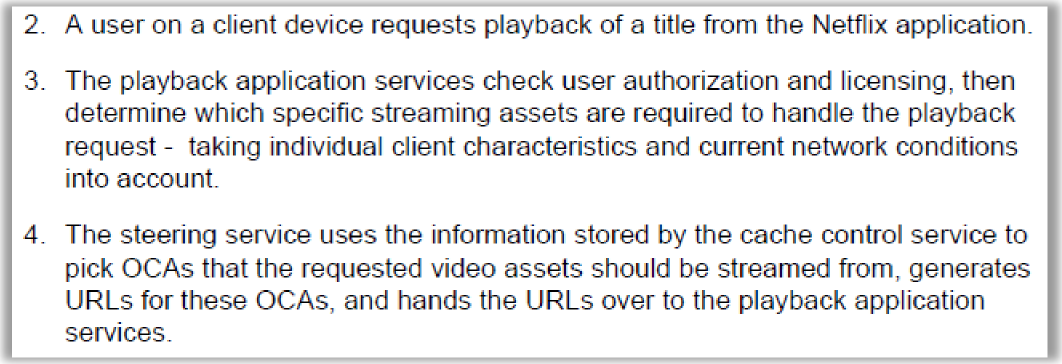
The playback device could be, for example, a PC running Windows 10 using the Edge browser from Microsoft. The request identifies a piece of content, for example, using a movieID. And the request includes a product identifier, for example, Win10 PC/Edge. The request includes information necessary to determine the playback device's version and at least one device capability based on the product identifier. The Netflix "play decision" process is illustrated in the following exemplary presentation, published on February 21, 2018, at www.slideshare.net, from Suudhan Rangarajan, a Senior Software Engineer at Netflix:[90]

[90] Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 7.



233. Netflix "retriev[es], using the playback server system, (i) a list of assets associated with the identified piece of content and (ii) at least one device capability based upon the product identifier, wherein each asset is a different stream associated with the piece of content." Netflix uses its playback server system to retrieve a list of assets associated with the identified piece of content. More specifically, and as illustrated in, for example, Netflix Open Connect documentation, Netflix's playback server system identifies and retrieves the specific streaming assets that are required to handle the playback request:[91]

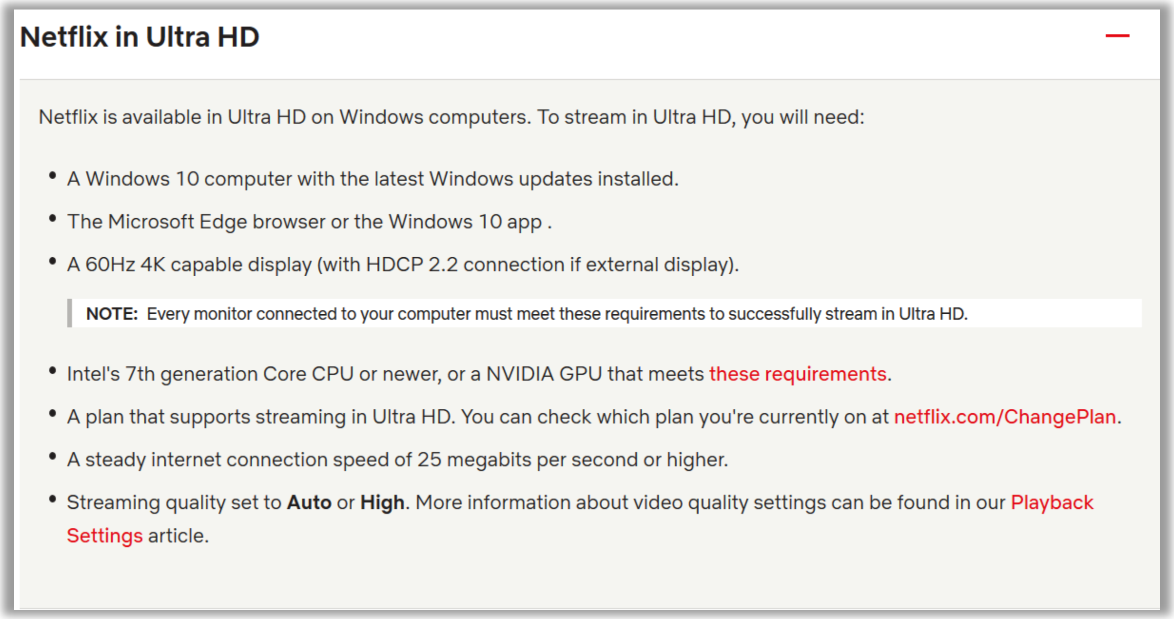
[91] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 4.



2. A user on a client device requests playback of a title from the Netflix application.
3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and current network conditions into account.
4. The steering service uses the information stored by the cache control service to pick OCAs that the requested video assets should be streamed from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

The Netflix playback server system has a list of different streams associated with the requested piece of content, for example, using the movieID—streams in different formats for different device capabilities. Multiple resolutions and bitrates exist for the content associated with the movieID. Netflix uses its playback server system to retrieve at least one device capability based on the product identifier. For example, Netflix will stream 4K/UltraHD content encoded with the H.265 codec to only a 4K/UltraHD capable PC, depending on its OS version, browser type, H.265 capability, DRM and content protection capabilities and robustness, and 60Hz HDMI.[92]

[92] https://help.netflix.com/en/node/23931; https://nvidia.custhelp.com/app/answers/detail/a_id/4583/~/4k-uhd-netflix-content-on-nvidia-gpus.



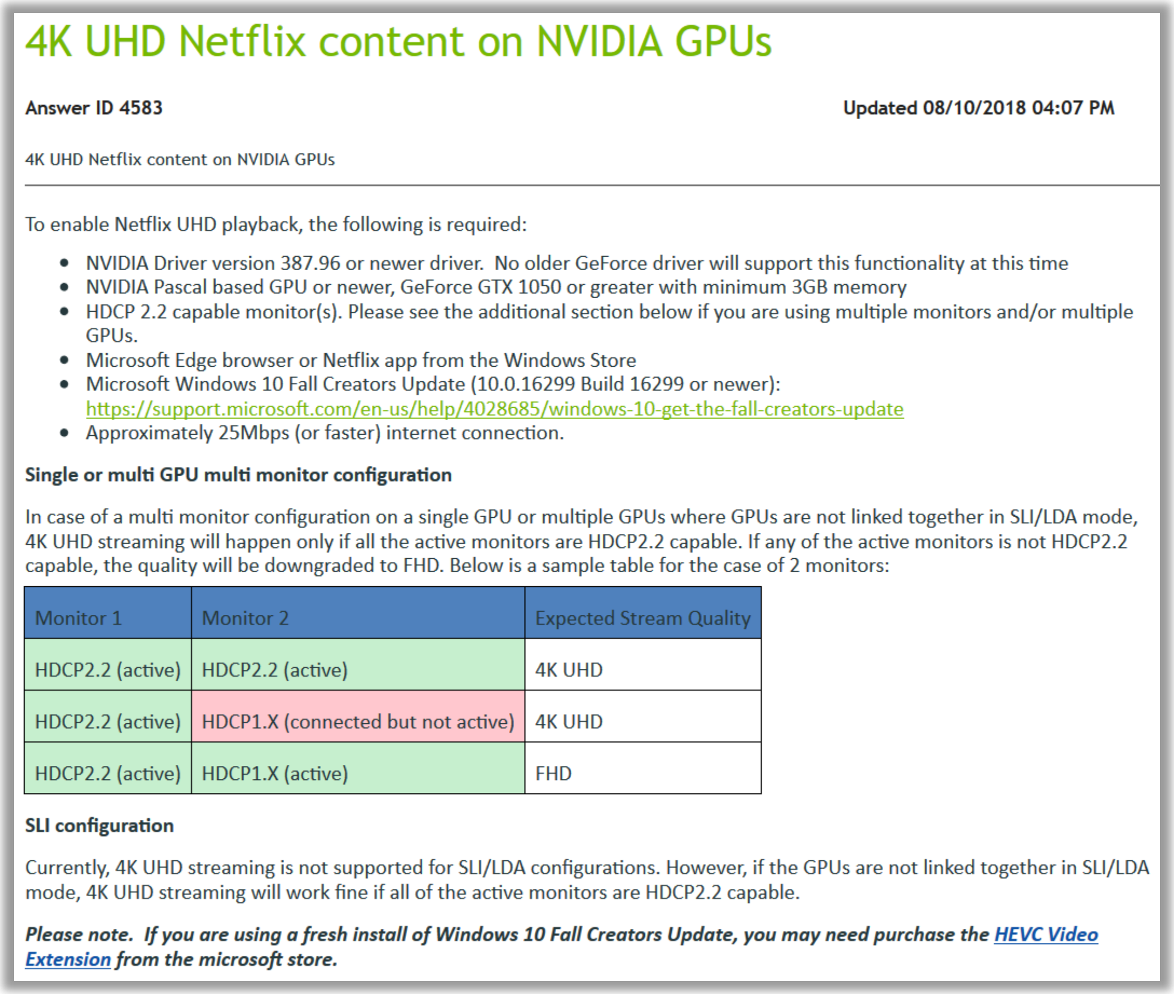
Netflix in Ultra HD

Netflix is available in Ultra HD on Windows computers. To stream in Ultra HD, you will need:

- A Windows 10 computer with the latest Windows updates installed.
- The Microsoft Edge browser or the Windows 10 app .
- A 60Hz 4K capable display (with HDCP 2.2 connection if external display).

> **NOTE:** Every monitor connected to your computer must meet these requirements to successfully stream in Ultra HD.

- Intel's 7th generation Core CPU or newer, or a NVIDIA GPU that meets these requirements.
- A plan that supports streaming in Ultra HD. You can check which plan you're currently on at netflix.com/ChangePlan.
- A steady internet connection speed of 25 megabits per second or higher.
- Streaming quality set to **Auto** or **High**. More information about video quality settings can be found in our Playback Settings article.



4K UHD Netflix content on NVIDIA GPUs

Answer ID 4583 **Updated 08/10/2018 04:07 PM**

4K UHD Netflix content on NVIDIA GPUs

To enable Netflix UHD playback, the following is required:

- NVIDIA Driver version 387.96 or newer driver. No older GeForce driver will support this functionality at this time
- NVIDIA Pascal based GPU or newer, GeForce GTX 1050 or greater with minimum 3GB memory
- HDCP 2.2 capable monitor(s). Please see the additional section below if you are using multiple monitors and/or multiple GPUs.
- Microsoft Edge browser or Netflix app from the Windows Store
- Microsoft Windows 10 Fall Creators Update (10.0.16299 Build 16299 or newer): https://support.microsoft.com/en-us/help/4028685/windows-10-get-the-fall-creators-update
- Approximately 25Mbps (or faster) internet connection.

Single or multi GPU multi monitor configuration

In case of a multi monitor configuration on a single GPU or multiple GPUs where GPUs are not linked together in SLI/LDA mode, 4K UHD streaming will happen only if all the active monitors are HDCP2.2 capable. If any of the active monitors is not HDCP2.2 capable, the quality will be downgraded to FHD. Below is a sample table for the case of 2 monitors:

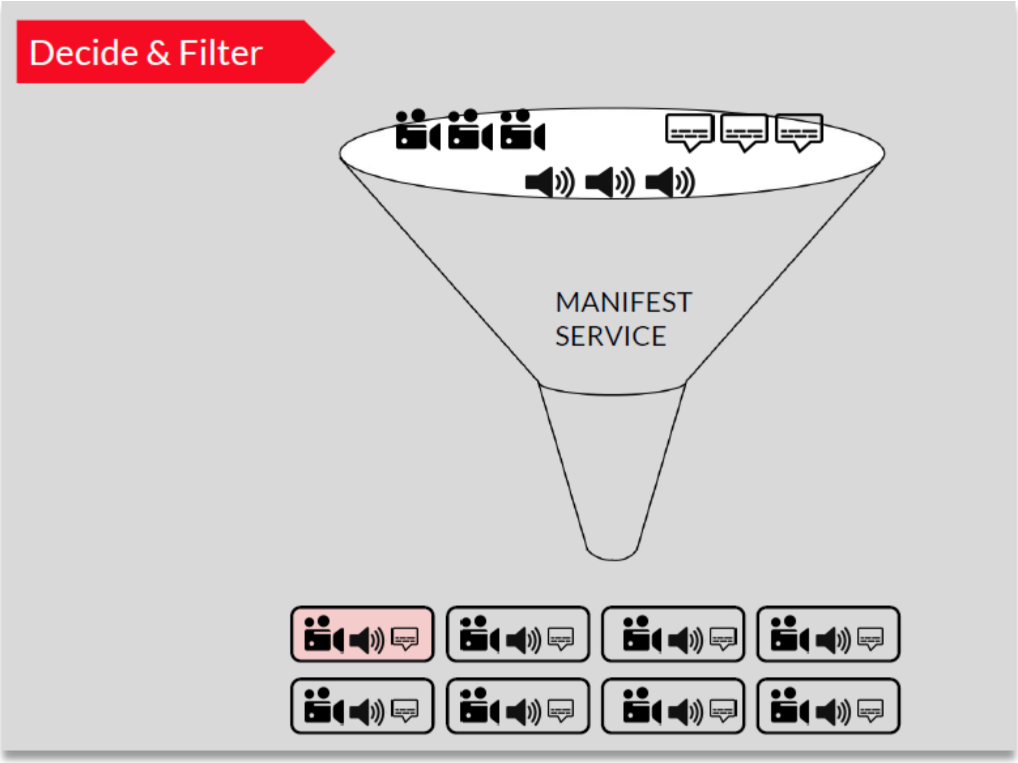
Monitor 1	Monitor 2	Expected Stream Quality
HDCP2.2 (active)	HDCP2.2 (active)	4K UHD
HDCP2.2 (active)	HDCP1.X (connected but not active)	4K UHD
HDCP2.2 (active)	HDCP1.X (active)	FHD

SLI configuration

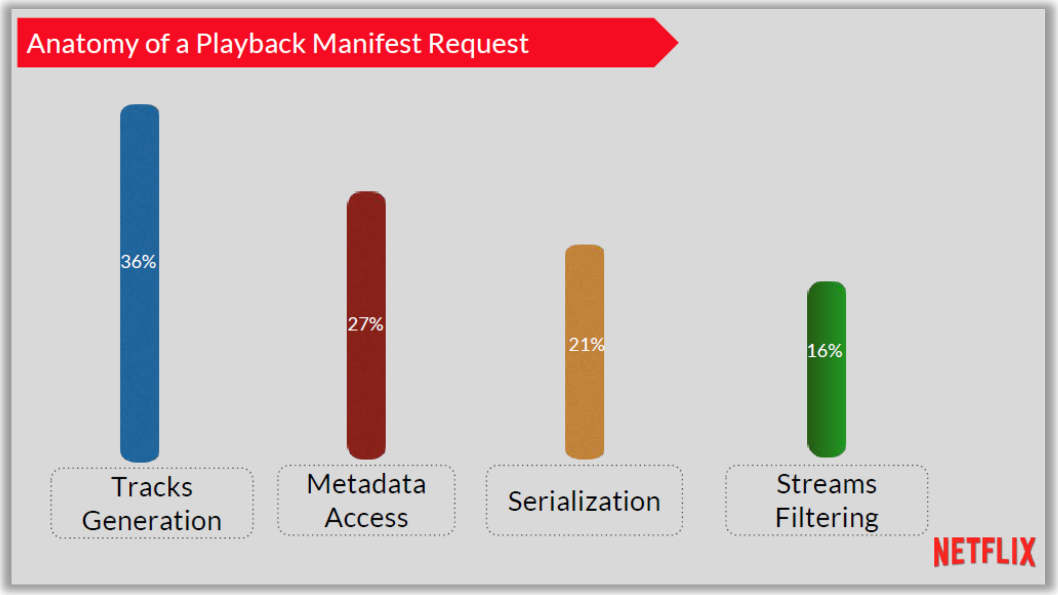
Currently, 4K UHD streaming is not supported for SLI/LDA configurations. However, if the GPUs are not linked together in SLI/LDA mode, 4K UHD streaming will work fine if all of the active monitors are HDCP2.2 capable.

Please note. If you are using a fresh install of Windows 10 Fall Creators Update, you may need purchase the HEVC Video Extension from the microsoft store.

234. Netflix "filter[s] the list of assets using the at least one device capability using the playback server system, wherein the playback server system maintains a database of product identifiers and associated device capabilities." Netflix indicates that it uses a decide-and-filter process for the manifest delivery service.[93]


Decide & Filter

[93] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 10, 23.

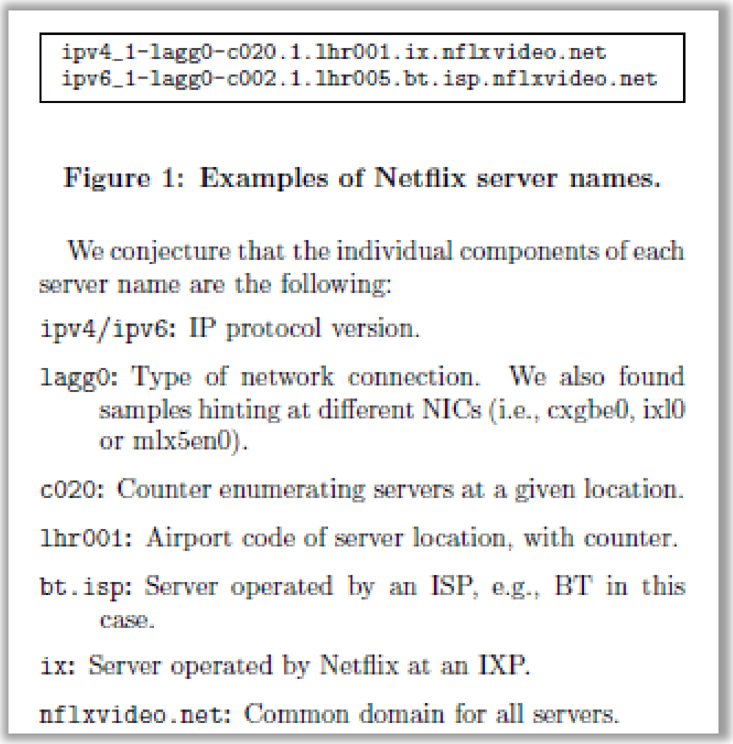


Netflix generates a different manifest based on the device capability (or capabilities) using its playback server system. Netflix filters the list of assets (bitrate/resolution/format) based on the device capabilities. For example, as described in the previous paragraph, Netflix will stream 4K/UltraHD content encoded with the H.265 codec to only a 4K/UltraHD capable PC, depending on its OS version, browser type, H.265 capability, DRM and content protection capabilities and robustness, and 60Hz HDMI.[94] Netflix's playback server system maintains a database of product identifiers and associated device capabilities.

235. Netflix "generat[es] a top level index file describing each asset in the filtered list of assets using the playback server system," that is, a manifest. Netflix generates the manifest using its playback server system, and each is specific to the playback device capabilities and contains CDN server locations for download of the assets. The distribution and naming of Netflix's CDN is documented in published papers and supports the step of generating the manifest file.[95]

[94] https://help.netflix.com/en/node/23931.

[95] *See* "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN," arXiv:1606.05519v1 [cs.NI], 17 Jun 2016, *available*



```
ipv4_1-lagg0-c020.1.lhr001.ix.nflxvideo.net
ipv6_1-lagg0-c002.1.lhr005.bt.isp.nflxvideo.net
```

Figure 1: Examples of Netflix server names.

We conjecture that the individual components of each server name are the following:

`ipv4/ipv6`: IP protocol version.

`lagg0`: Type of network connection. We also found samples hinting at different NICs (i.e., cxgbe0, ixl0 or mlx5en0).

`c020`: Counter enumerating servers at a given location.

`lhr001`: Airport code of server location, with counter.

`bt.isp`: Server operated by an ISP, e.g., BT in this case.

`ix`: Server operated by Netflix at an IXP.

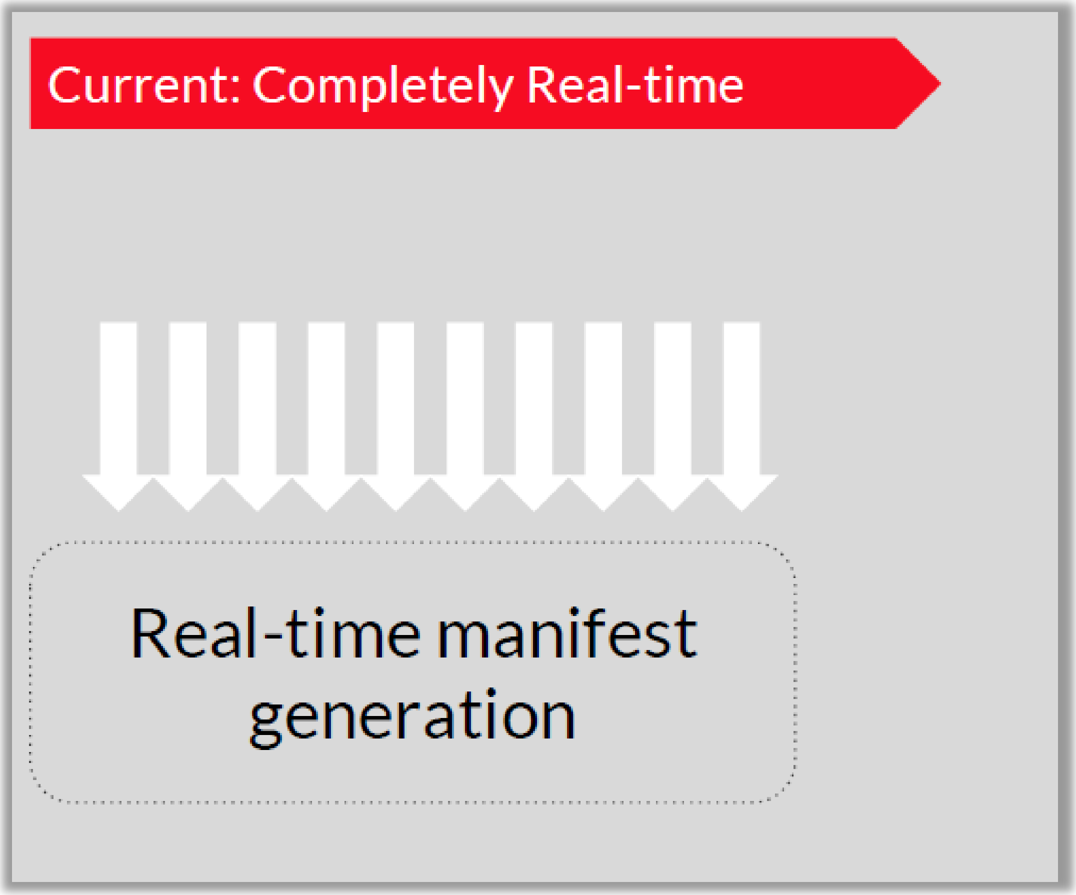
`nflxvideo.net`: Common domain for all servers.

As illustrated in the following exemplary Netflix presentation, Netflix generates the manifest, either in real-time or pre-cached:[96]

at https://arxiv.org/abs/1606.05519.

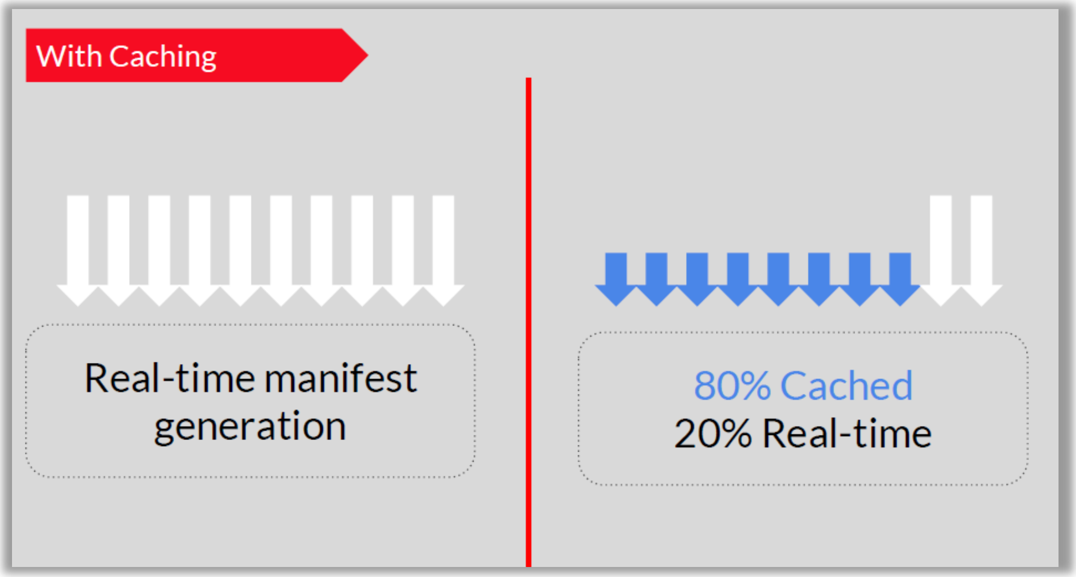
[96] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 17-18.

ROBINS KAPLAN LLP
ATTORNEYS AT LAW
LOS ANGELES



Current: Completely Real-time

Real-time manifest generation



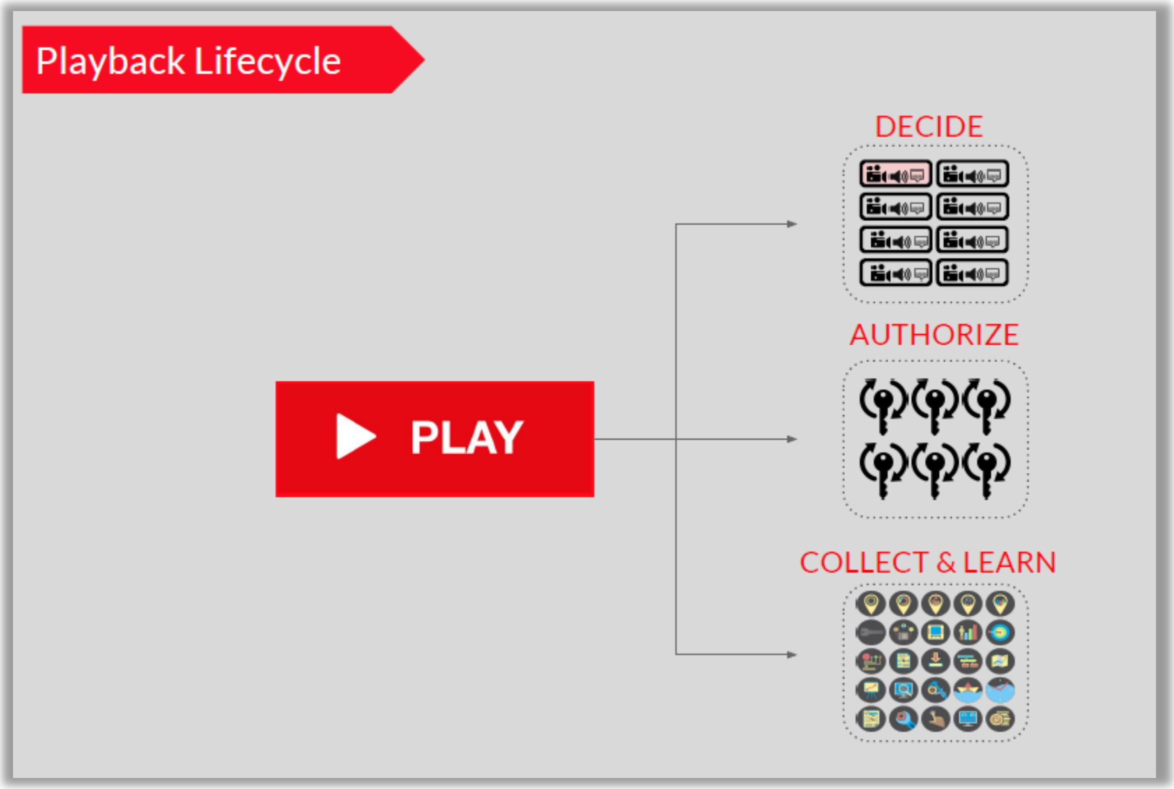
With Caching

Real-time manifest generation

80% Cached
20% Real-time

236. Netflix “send[s] the top level index file to the playback device using the playback server system, wherein the top level index file is used by the playback

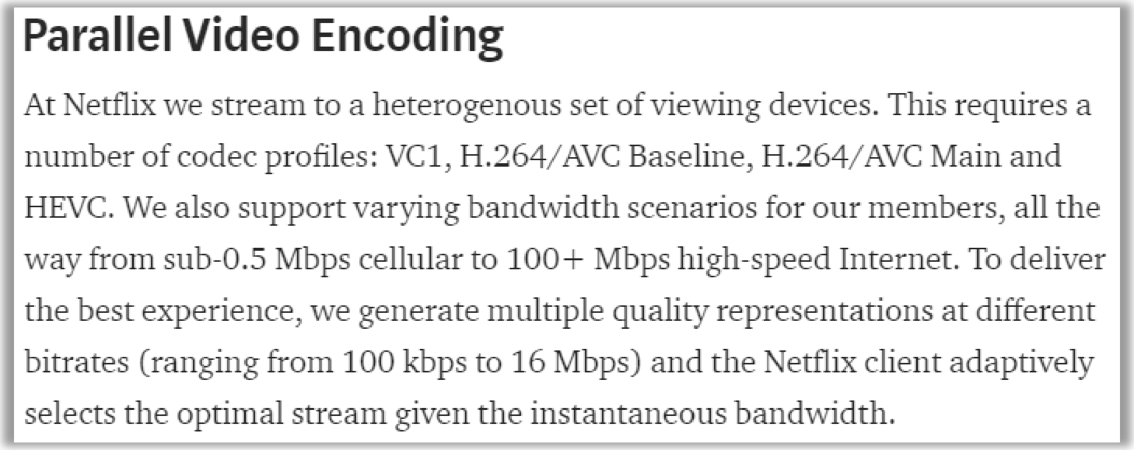
device to determine which assets to request for playback on the device." Netflix sends the top-level index file—the manifest—to the playback device using the playback server system via, for example, the MSL layer request (POST) and response (GET). The manifest is used within the playback device to request the video streams for playback. As illustrated in the following exemplary Netflix presentation, the playback server system requires a "decide process" to send the appropriate manifest to the playback device:[97]



Further, Netflix admits that it does adaptive streaming using the manifests, suggesting that the playback device uses the manifest to determine the assets to request for playback on the device:[98]

[97] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 3.

[98] https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746.



Parallel Video Encoding

At Netflix we stream to a heterogenous set of viewing devices. This requires a number of codec profiles: VC1, H.264/AVC Baseline, H.264/AVC Main and HEVC. We also support varying bandwidth scenarios for our members, all the way from sub-0.5 Mbps cellular to 100+ Mbps high-speed Internet. To deliver the best experience, we generate multiple quality representations at different bitrates (ranging from 100 kbps to 16 Mbps) and the Netflix client adaptively selects the optimal stream given the instantaneous bandwidth.

237. Netflix directly infringes at least claim 1, at least as described, when it tests its service using various playback devices.

238. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

239. Netflix has infringed, and continues to infringe, at least claim 1 of the ’720 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’720 Infringing Products, in violation of 35 U.S.C. § 271(a).

240. Netflix’s infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix’s wrongful acts in an amount subject to proof at trial.

COUNT VI: INFRINGEMENT OF U.S. PATENT NO. 9,998,515

241. The allegations of paragraphs 1-240 of this Complaint are incorporated by reference as though fully set forth herein.

242. Pursuant to 35 U.S.C. § 282, the ’515 patent is presumed valid.

243. Upon information and belief, Netflix directly infringes the ’515 patent by making, using, offering to sell, selling, and/or importing into the United States

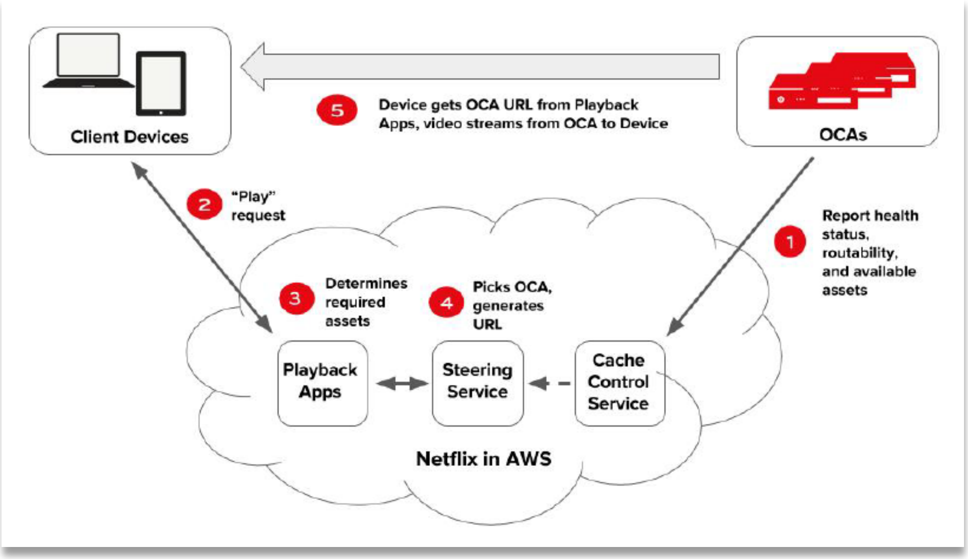
its Netflix service, which provides a system and method for automatically generating top level index files (collectively, the "Accused '515 Infringing Products").

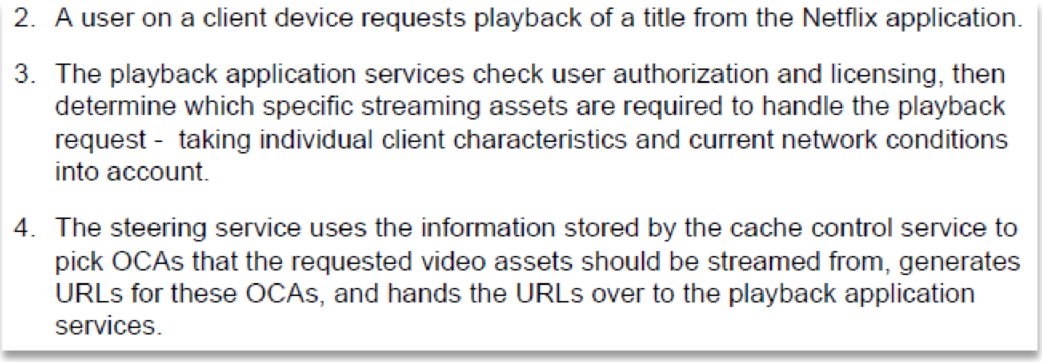
244. Upon information and belief, the Accused '515 Infringing Products directly infringe at least claim 1 of the '515 patent at least in the exemplary manner described in paragraphs 245-251 below.

245. Netflix practices a "method for authorizing playback of content," that is, its streaming service.

246. Netflix "receiv[es] a request for content from a playback device at a playback server, where the request includes a product identifier that identifies a device configuration" when its streaming infrastructure, that is, its playback server, receives a request for content from a playback device, where the request includes a product identifier that identifies a device configuration. As illustrated in, for example, Netflix Open Connect documentation, Netflix's playback server receives a request from a playback device that identifies requested video assets and "individual client characteristics."[99]

[99] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 4.


Client Devices
5 Device gets OCA URL from Playback Apps, video streams from OCA to Device
OCAs
2 "Play" request
1 Report health status, routability, and available assets
3 Determines required assets
4 Picks OCA, generates URL
Playback Apps
Steering Service
Cache Control Service
Netflix in AWS



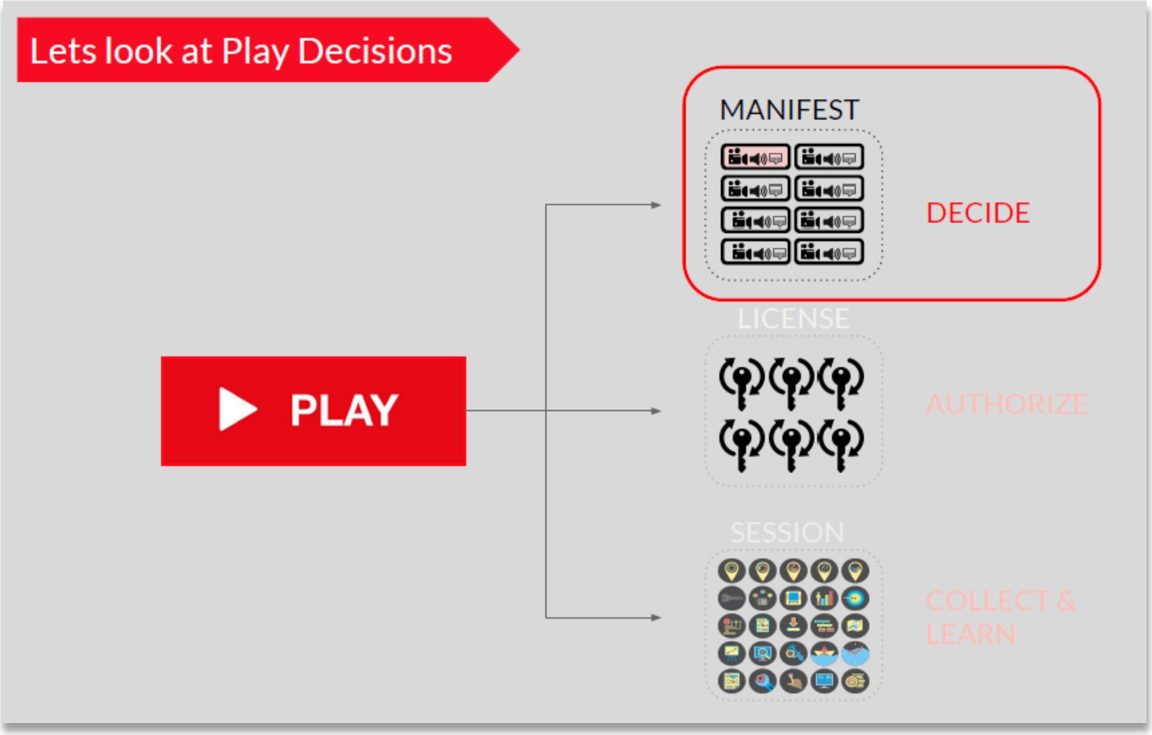
2. A user on a client device requests playback of a title from the Netflix application.

3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and current network conditions into account.

4. The steering service uses the information stored by the cache control service to pick OCAs that the requested video assets should be streamed from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

The playback device could be, for example, a PC running Windows 10 using the Edge browser from Microsoft. The request for content includes a product identifier, for example, Win10 PC/Edge. And the request includes information necessary to determine the playback device's version and at least one device capability based on the product identifier and, as a result, identifies a device configuration. The Netflix "play decision" process is illustrated below in an exemplary presentation, from Suudhan Rangarajan, a Senior Software Engineer at Netflix:[100]

[100] Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 8.



Lets look at Play Decisions

PLAY

MANIFEST

DECIDE

LICENSE

AUTHORIZE

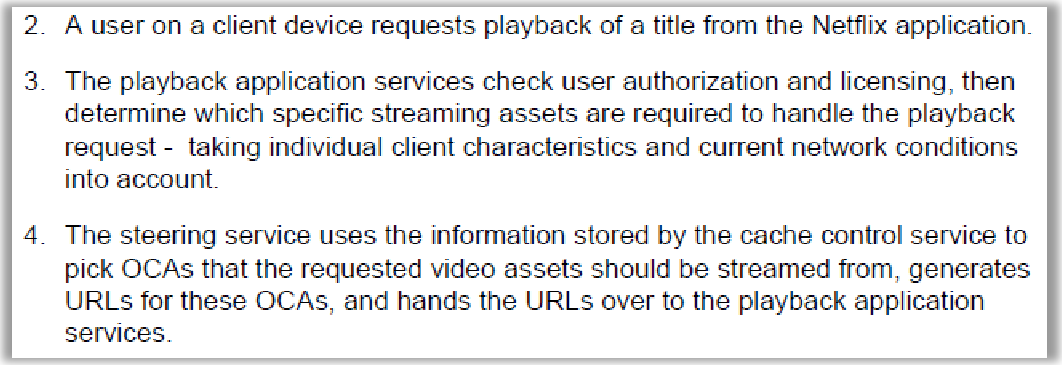
SESSION

COLLECT & LEARN

247. Netflix "identif[es], using the playback server, based on the product identifier, a plurality of device capabilities including a device type and a device software version indicating a version number for an adaptive streaming software component implemented on the playback device." For example, the playback device could be a PC running Windows 10 using the Edge browser from Microsoft. Netflix uses its playback server to identify the device type, based on the product identifier, identified by the flag "MicrosoftEnableDeviceInfo." Furthermore, a device software version indicating a version number for an adaptive streaming software component is also included, which is the version number of the Edge browser. And as discussed in the previous paragraph, the request for content includes a product identifier, for example, Win10 PC/Edge. And the request includes, for example, the capabilities and version of the playback device.

248. Netflix "retriev[es], using the playback server, a list of assets associated with the identified piece of content, wherein each asset is a different stream associated with the piece of content." Netflix uses its playback server to

retrieve a list of assets associated with the identified piece of content. More specifically, as illustrated in Netflix Open Connect documentation, Netflix's playback server determines which specific streaming assets are required to handle the playback request.[101]



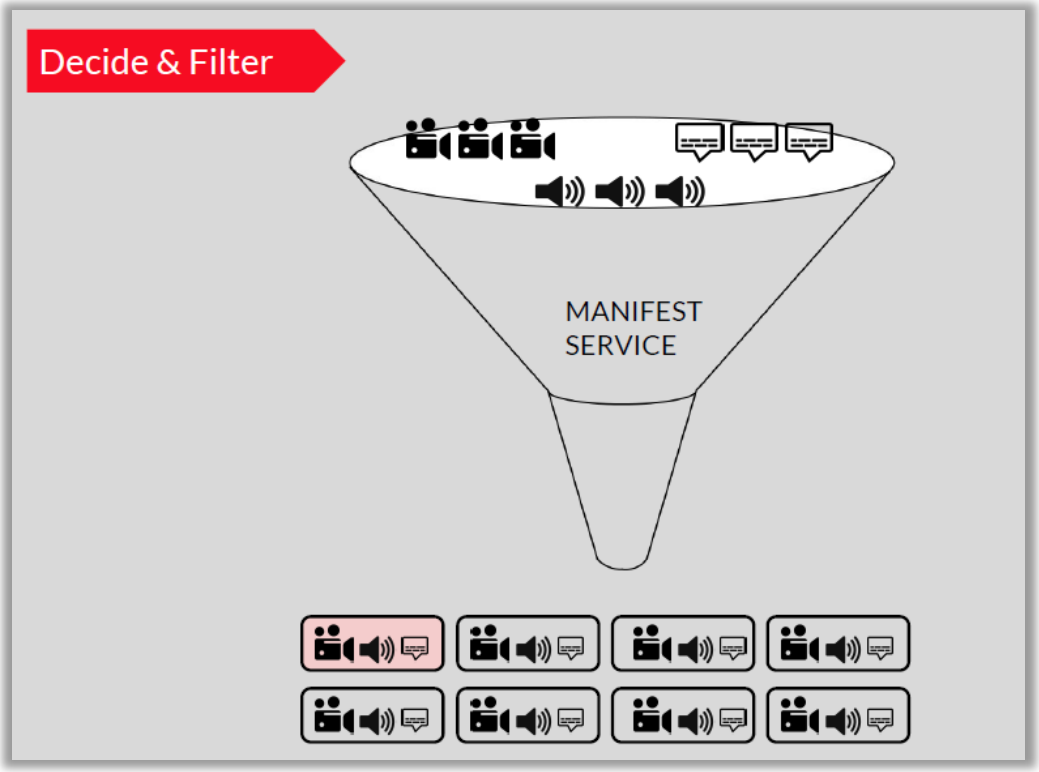
2. A user on a client device requests playback of a title from the Netflix application.
3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and current network conditions into account.
4. The steering service uses the information stored by the cache control service to pick OCAs that the requested video assets should be streamed from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

The Netflix playback server has a list of different streams associated with the requested piece of content, for example, using the movieID—streams in different formats for different device capabilities. Multiple resolutions and bitrates exist for the content associated with the movieID.

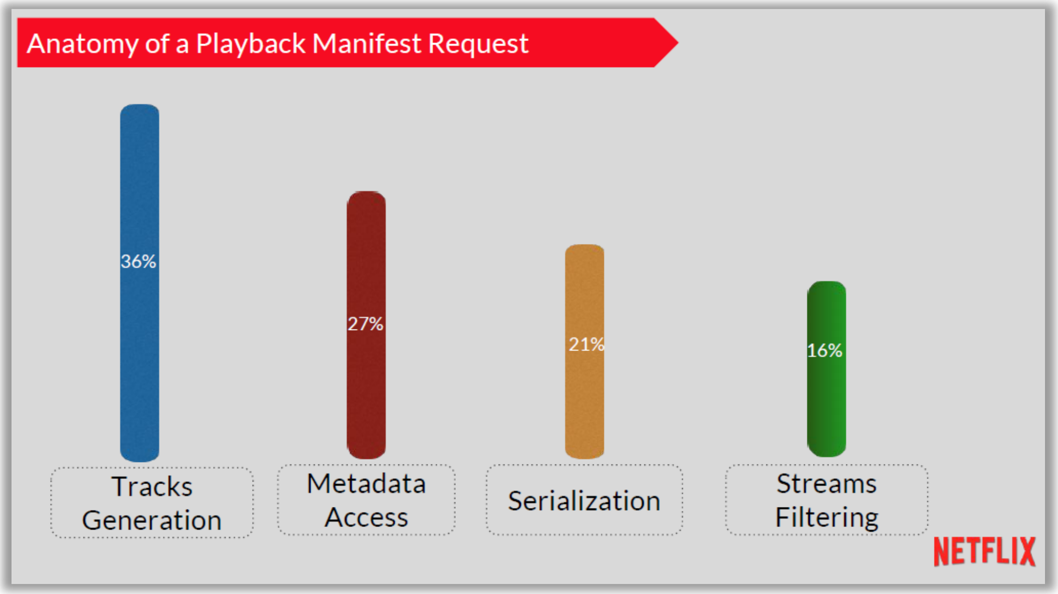
249. Netflix "filter[s], using the playback server, the list of assets based on the plurality of device capabilities." Netflix indicates that it uses a decide-and-filter process for the manifest delivery service.[102]

[101] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 4.

[102] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 10, 23.



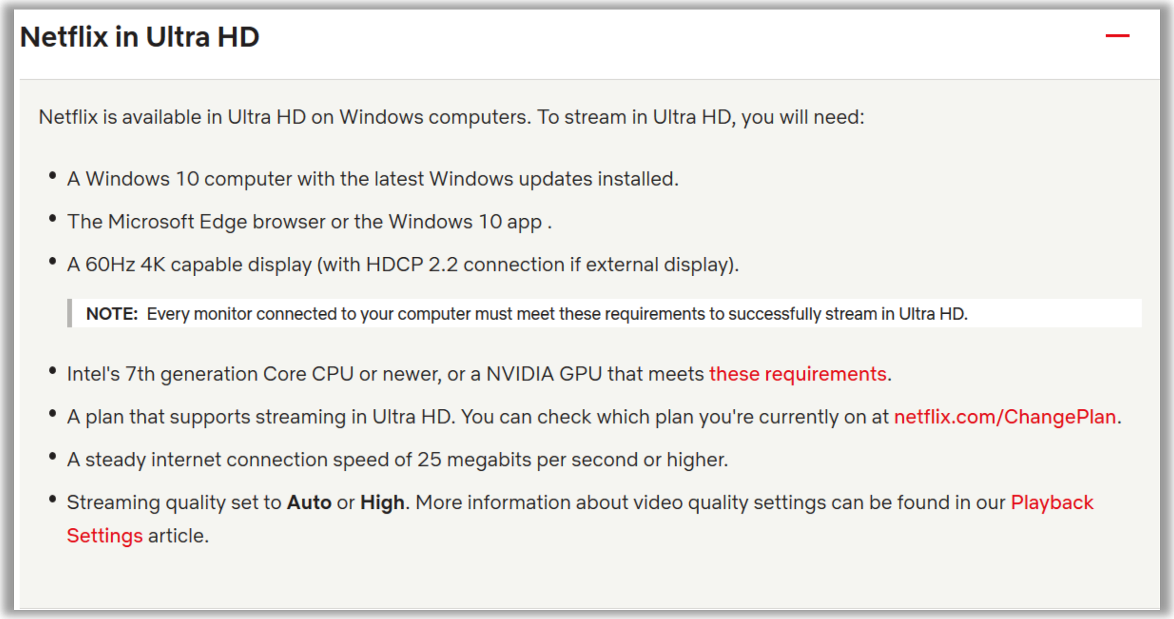
Decide & Filter



Anatomy of a Playback Manifest Request

Netflix generates a different manifest based on the device capabilities using its playback server. Netflix filters the list of assets (bitrate/resolution/format) based on the device capabilities. For example, Netflix will stream 4K/UltraHD content encoded with the H.265 codec to only a 4K/UltraHD capable PC, depending on its

OS version, browser type, H.265 capability, DRM and content protection capabilities and robustness, and 60Hz HDMI.[103]



Netflix in Ultra HD

Netflix is available in Ultra HD on Windows computers. To stream in Ultra HD, you will need:

- A Windows 10 computer with the latest Windows updates installed.
- The Microsoft Edge browser or the Windows 10 app .
- A 60Hz 4K capable display (with HDCP 2.2 connection if external display).

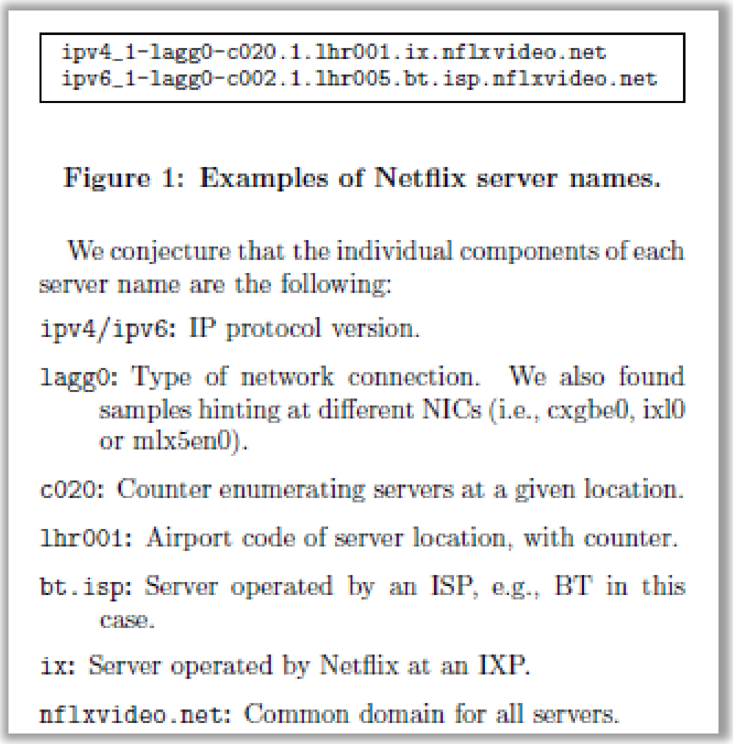
> **NOTE:** Every monitor connected to your computer must meet these requirements to successfully stream in Ultra HD.

- Intel's 7th generation Core CPU or newer, or a NVIDIA GPU that meets these requirements.
- A plan that supports streaming in Ultra HD. You can check which plan you're currently on at netflix.com/ChangePlan.
- A steady internet connection speed of 25 megabits per second or higher.
- Streaming quality set to **Auto** or **High**. More information about video quality settings can be found in our Playback Settings article.

250. Netflix "generat[es], using the playback server, a top level index file describing each asset in the filtered list of assets, wherein the top level index file identifies locations and bitrates of a plurality of alternative streams capable of being used to perform adaptive streamlining of the content." Netflix, using its playback server, generates a manifest, that is, a top-level index file. Each is specific to the playback device capabilities and contains CDN server locations for download of the assets. The distribution and naming of Netflix's CDN is documented in published papers and supports the step of generating the manifest file.[104]

[103] https://help.netflix.com/en/node/23931.

[104] *See* "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN," arXiv:1606.05519v1 [cs.NI], 17 Jun 2016, *available at* https://arxiv.org/abs/1606.05519.



```
ipv4_1-lagg0-c020.1.lhr001.ix.nflxvideo.net
ipv6_1-lagg0-c002.1.lhr005.bt.isp.nflxvideo.net
```

Figure 1: Examples of Netflix server names.

We conjecture that the individual components of each server name are the following:

`ipv4/ipv6`: IP protocol version.

`lagg0`: Type of network connection. We also found samples hinting at different NICs (i.e., cxgbe0, ixl0 or mlx5en0).

`c020`: Counter enumerating servers at a given location.

`lhr001`: Airport code of server location, with counter.

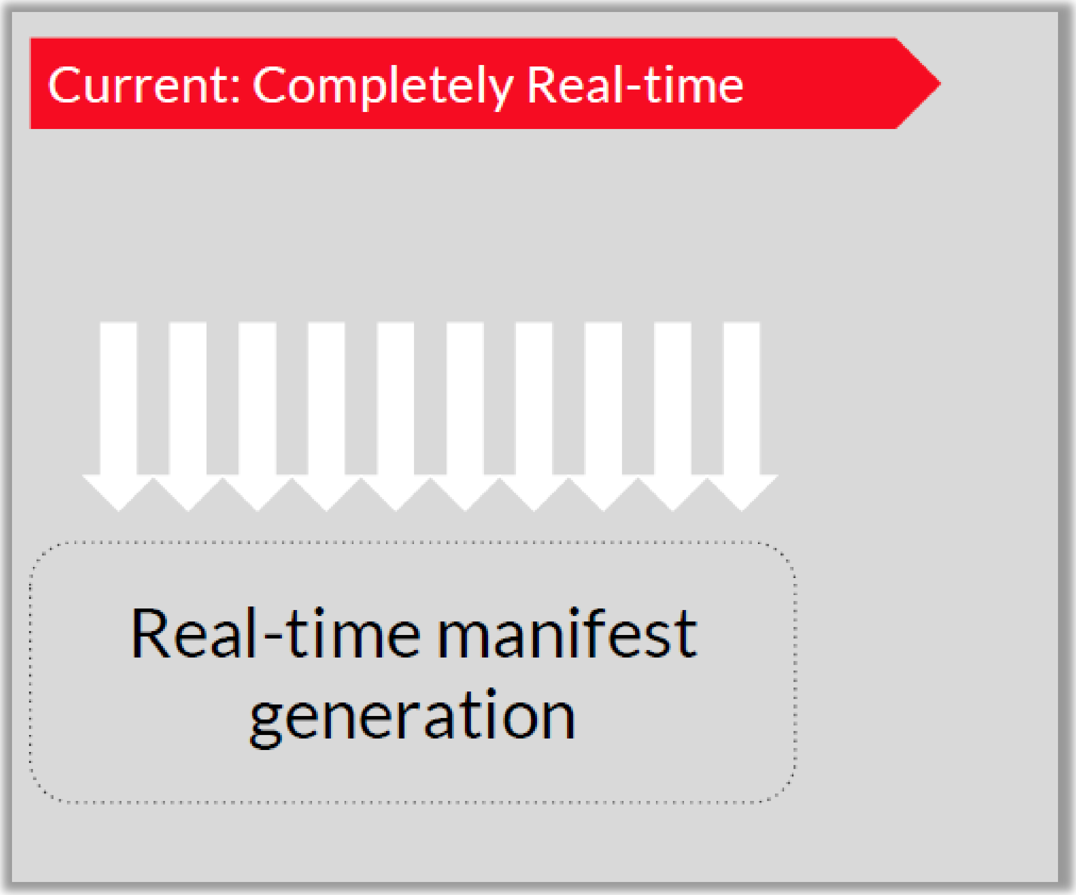
`bt.isp`: Server operated by an ISP, e.g., BT in this case.

`ix`: Server operated by Netflix at an IXP.

`nflxvideo.net`: Common domain for all servers.

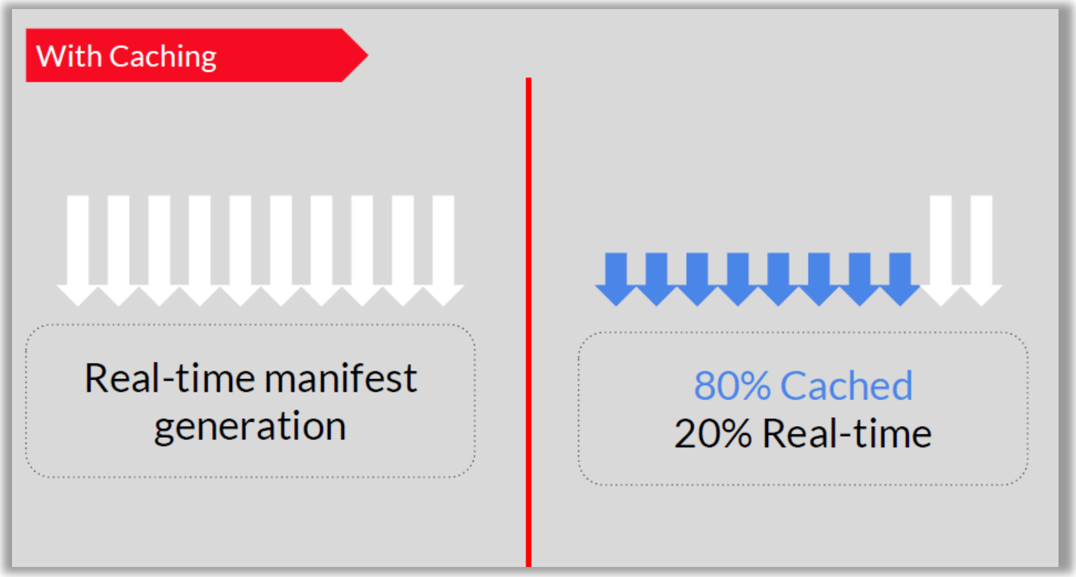
As illustrated in the following exemplary Netflix presentation, Netflix generates the manifest, either in real-time or pre-cached:[105]

[105] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 17-18.



Current: Completely Real-time

Real-time manifest generation



With Caching

Real-time manifest generation

80% Cached
20% Real-time

251. Netflix "send[s] the top level index file from the playback server to the playback device." Netflix sends the top-level index file—the manifest—to the

playback device using the playback server via, for example, the MSL layer request (POST) and response (GET).

252. Netflix directly infringes at least claim 1, at least as described, when it tests its service using various playback devices.

253. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

254. Netflix has infringed, and continues to infringe, at least claim 1 of the ’515 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’515 Infringing Products, in violation of 35 U.S.C. § 271(a).

255. Netflix has induced, and continues to induce, infringement of at least claim 16 of the ’515 patent, at least in the exemplary manner described in paragraphs 256-265, in violation of 35 U.S.C. § 271(b).

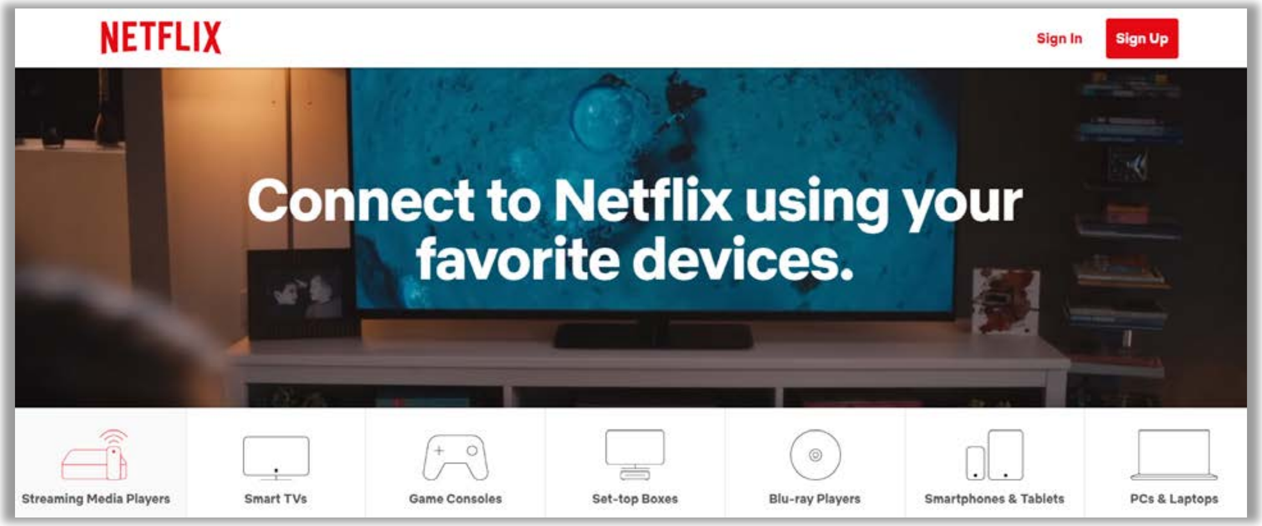
256. At least as of the date of this Complaint, Netflix knows that the ’515 patent enables it to offer adaptive bitrate streaming services that perform smoothly and without stalls when switching among video streams of different resolution during playback on a user’s device. Specifically, at least as of the date of this Complaint, Netflix knows that the ’515 patent is directed to a playback server system that automatically generates a top-level index file tailored to a particular playback device that the playback device uses to request a streaming file, improving adaptive bitrate streaming.

257. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service to be used with a CE playback device that, when used as intended, meets the limitations of claim 16.

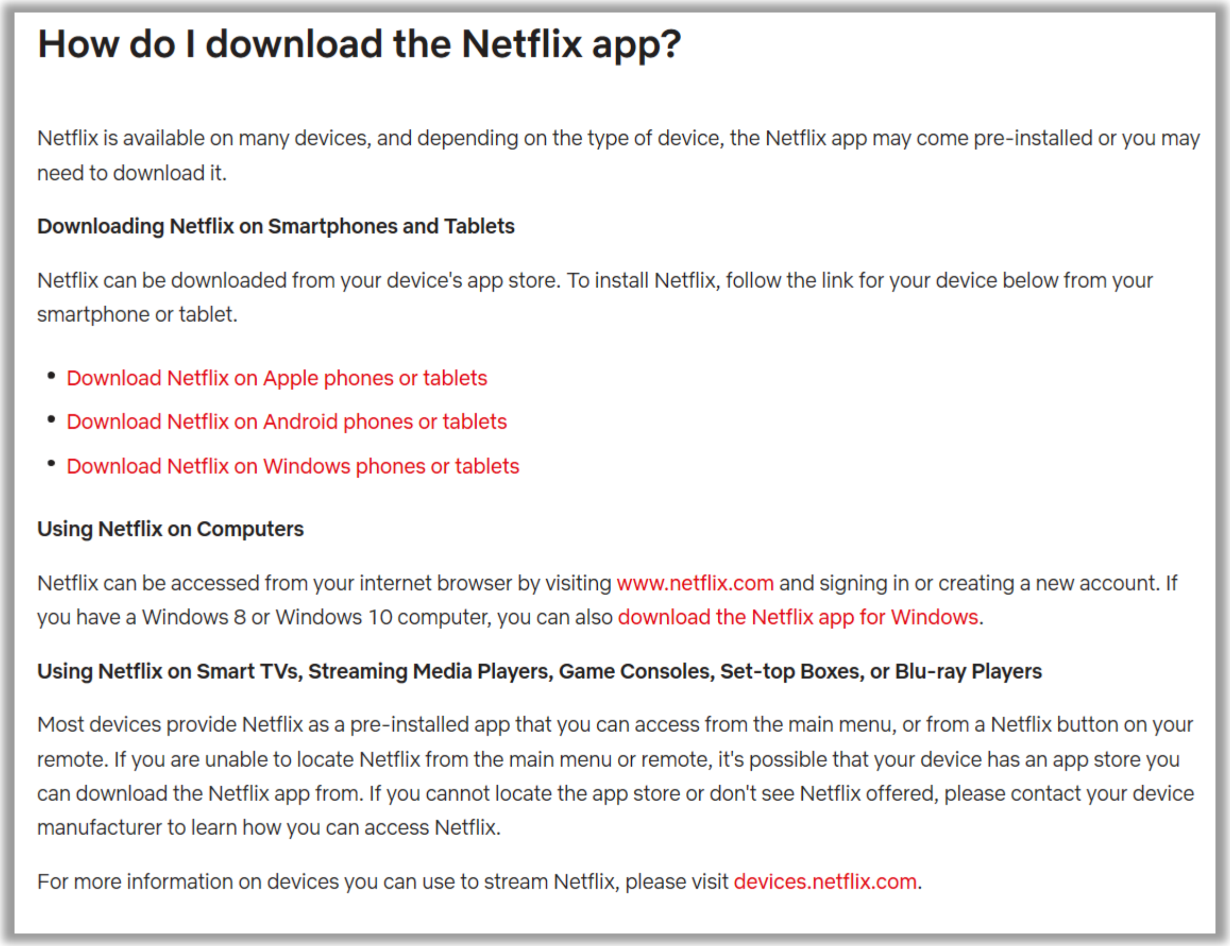
258. At least as of the date of this Complaint, Netflix knows and specifically intends that end-user CE playback devices be a device that meets all of

the limitations of claim 16, when the Netflix application is enabled on the playback device as intended.

259. The CE playback device enabling the Netflix application comprises "memory containing information used to identify capabilities of the playback device." The Netflix application runs on a device with memory containing information used to identify capabilities of the playback device, as illustrated on Netflix's website:[106]



[106] https://devices.netflix.com/en/; https://help.netflix.com/en/node/101653?ba=SwiftypeResultClick&q=install%20app%20browser.



How do I download the Netflix app?

Netflix is available on many devices, and depending on the type of device, the Netflix app may come pre-installed or you may need to download it.

Downloading Netflix on Smartphones and Tablets

Netflix can be downloaded from your device's app store. To install Netflix, follow the link for your device below from your smartphone or tablet.

- Download Netflix on Apple phones or tablets
- Download Netflix on Android phones or tablets
- Download Netflix on Windows phones or tablets

Using Netflix on Computers

Netflix can be accessed from your internet browser by visiting www.netflix.com and signing in or creating a new account. If you have a Windows 8 or Windows 10 computer, you can also download the Netflix app for Windows.

Using Netflix on Smart TVs, Streaming Media Players, Game Consoles, Set-top Boxes, or Blu-ray Players

Most devices provide Netflix as a pre-installed app that you can access from the main menu, or from a Netflix button on your remote. If you are unable to locate Netflix from the main menu or remote, it's possible that your device has an app store you can download the Netflix app from. If you cannot locate the app store or don't see Netflix offered, please contact your device manufacturer to learn how you can access Netflix.

For more information on devices you can use to stream Netflix, please visit devices.netflix.com.

The playback device could be, for example, a PC running Windows 10 with the Edge browser from Microsoft. The request from the playback device to the playback server includes a product identifier, for example, Win10 PC/Edge. And the request includes information necessary to determine the playback device’s version and at least one device capability based on the product identifier. That information is stored in the playback device’s memory.

260. The CE playback device enabling the Netflix application further comprises “a processor configured by a client application,” namely, the Netflix application. The Netflix application or JavaScript-implemented and browser-

enabled playback runs on a device with a processor, and the processor is configured by the Netflix application or JavaScript-implemented player.[107]

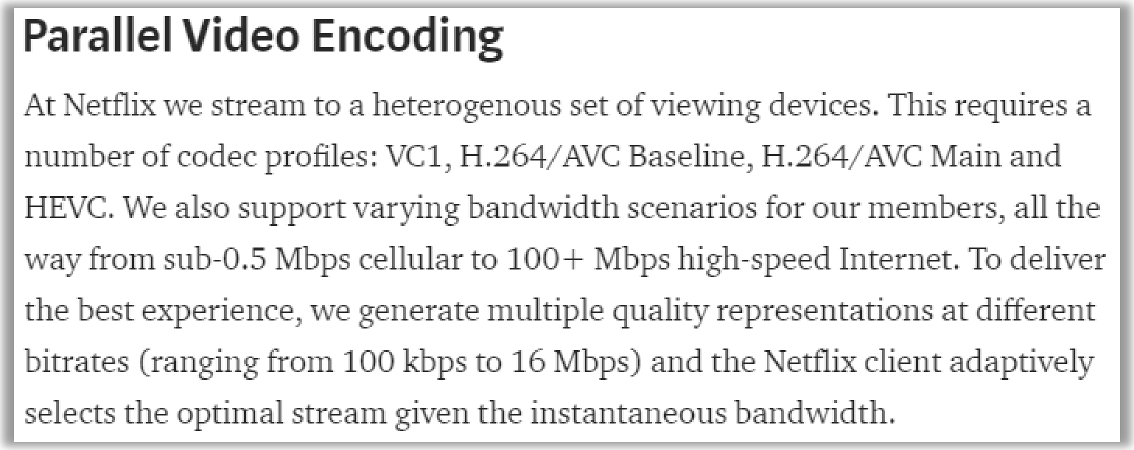
261. The Netflix application "configures the processor to request, using the playback device, a top level index file from a playback server, where the request identifies a piece of content and includes a software version indicating a version number for an adaptive streaming software component implemented on the device." The processor, configured by the Netflix application, uses the playback device to request a top-level index file—the manifest. This is realized via, for example, the MSL layer request (POST) and response (GET). The request identifies a piece of content and includes a software version indicating a version number for an adaptive streaming software component implemented on the device. The playback device could be, for example, a PC running Windows 10 using the Edge browser from Microsoft. The request identifies a piece of content, for example, using the movieID. The request further includes a version number for an adaptive streaming software component implemented on the device, for example, the Edge browser.

262. The Netflix application further configures the processor to "receive, using the playback device, a top level index file from the playback server, where the top level index file identifies locations and bitrates of a plurality of different alternative streams capable of being used to perform adaptive streaming of the identified piece of content and accessible to the playback device." The processor, configured by the Netflix application, uses the playback device to request and receive a manifest from the Netflix playback server. The manifest includes the locations and bitrates of a plurality of different alternative streams.

263. The Netflix application further configures the processor to "select, using the playback device, an initial stream from the plurality of different

[107] *Id.*

alternative streams." The processor, configured by the Netflix application, uses the playback device to select an initial stream from the urls listed in the manifest.[108]



Parallel Video Encoding

At Netflix we stream to a heterogenous set of viewing devices. This requires a number of codec profiles: VC1, H.264/AVC Baseline, H.264/AVC Main and HEVC. We also support varying bandwidth scenarios for our members, all the way from sub-0.5 Mbps cellular to 100+ Mbps high-speed Internet. To deliver the best experience, we generate multiple quality representations at different bitrates (ranging from 100 kbps to 16 Mbps) and the Netflix client adaptively selects the optimal stream given the instantaneous bandwidth.

264. The Netflix application further configures the processor to "retrieve, using the playback device, at least a portion of the initial stream from the locations identified in the top level index file." The processor, configured by the Netflix application, uses the playback device to request and receive the manifest. As discussed in previous paragraphs, the manifest includes the locations and bitrates of a plurality of different alternative streams. And at least a portion of the initial stream from one of the locations identified in the top-level index file—the manifest—is retrieved.

265. The Netflix application further configures the processor to "play back, using the playback device, the portion of the initial stream." After the processor retrieves the at least portion of the initial stream from one of the locations identified in the top-level index file—the manifest—the playback device plays the file.

266. Netflix's infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

[108] https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746.

COUNT VII: INFRINGEMENT OF U.S. PATENT NO. 10,212,486

267. The allegations of paragraphs 1-266 of this Complaint are incorporated by reference as though fully set forth herein.

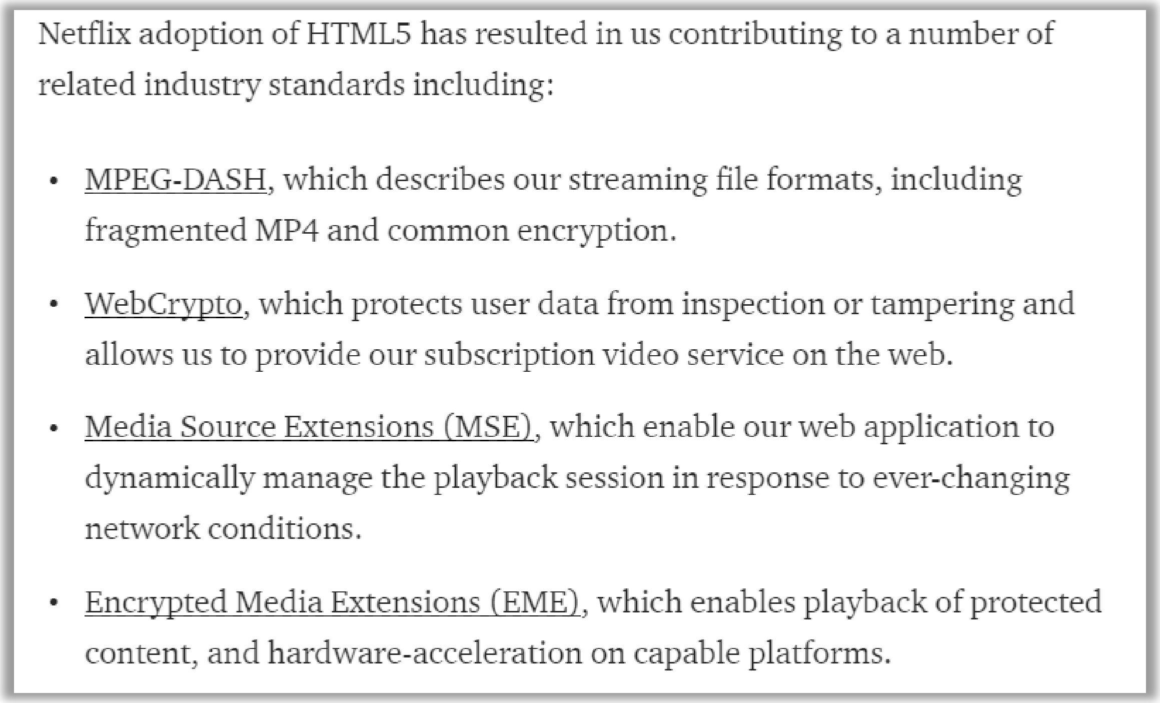
268. Pursuant to 35 U.S.C. § 282, the ’486 patent is presumed valid.

269. On information and belief, Netflix directly infringes the ’486 patent by making, using, offering to sell, selling, and/or importing into the United States its Netflix service, which provides playback devices and methods for deciphering frame keys within a secure video decoder, efficiently enhancing content security (collectively, the “Accused ’486 Infringing Products”).

270. On information and belief, the Accused ’486 Infringing Products directly infringe at least claim 1 of the ’486 patent at least as shown in the exemplary manner described in paragraphs 271-288 below.

271. Netflix provides “[a] playback device for playing back encrypted video” by providing applications that enable playback utilizing, for example, the MPEG-DASH Standard on a heterogeneous set of viewing devices.[109] On information and belief, at least the Netflix Microsoft Windows 10 Application, Netflix Android Application, and Netflix Android TV Application (“Netflix Apps”) use the MPEG-DASH Standard.

[109] https://medium.com/netflix-techblog/update-on-html5-video-for-netflix-fbb57e7d7ca0.



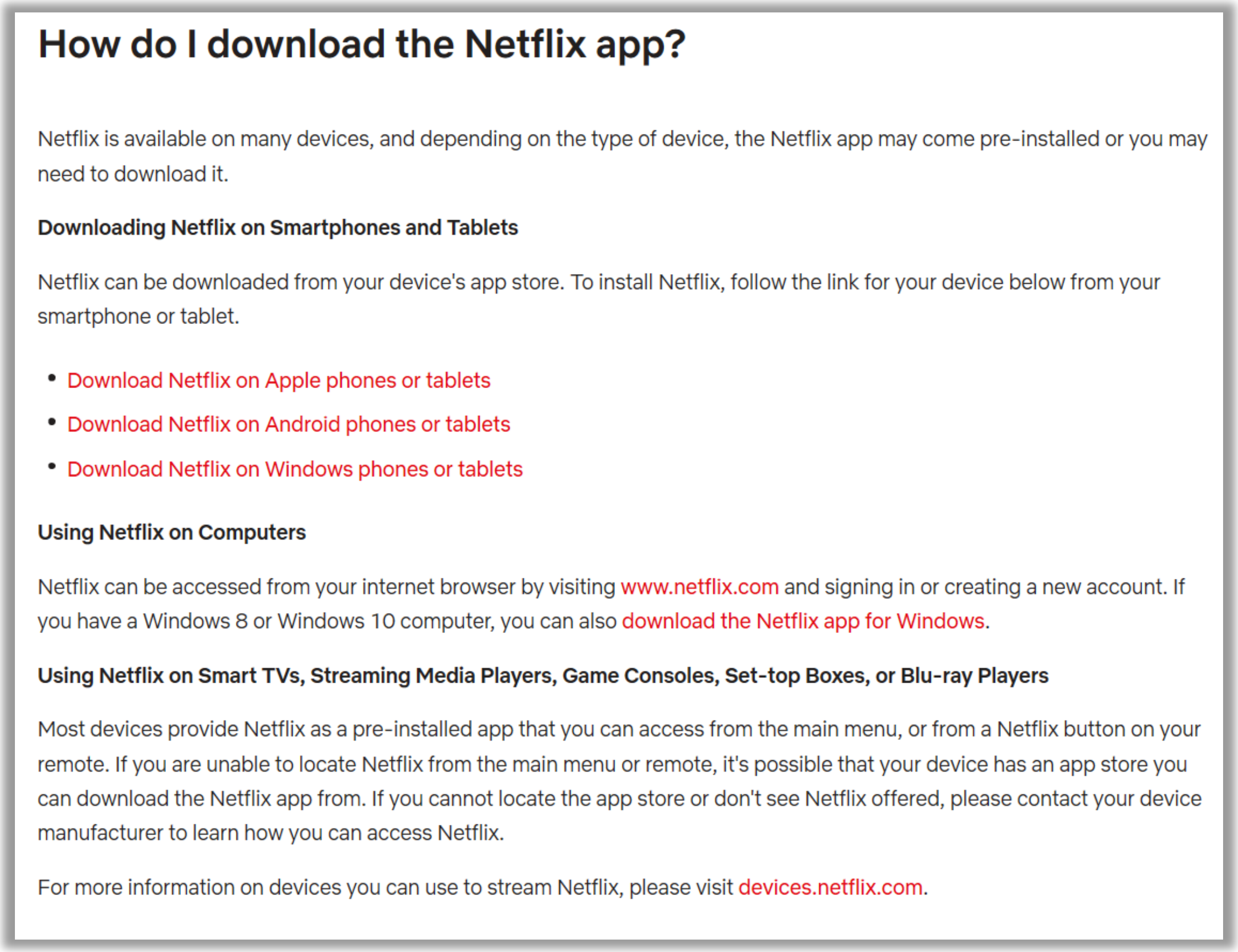
Netflix adoption of HTML5 has resulted in us contributing to a number of related industry standards including:

- MPEG-DASH, which describes our streaming file formats, including fragmented MP4 and common encryption.
- WebCrypto, which protects user data from inspection or tampering and allows us to provide our subscription video service on the web.
- Media Source Extensions (MSE), which enable our web application to dynamically manage the playback session in response to ever-changing network conditions.
- Encrypted Media Extensions (EME), which enables playback of protected content, and hardware-acceleration on capable platforms.

272. Netflix's playback device comprises "a set of one or more processors" because all playback devices that run the Netflix player application or other client applications that access the Netflix service include a set of one or more processors.

273. Netflix's playback device further comprises "a non-volatile storage containing a playback application" because in order to play back content, the playback device uses, for example, a Netflix player application that is either pre-installed or downloaded and stored in non-volatile memory. Netflix provides details on how to access the Netflix application on numerous devices—including smartphones, tablets, computers, smart TVs, streaming media players, game consoles, set-top boxes, and Blu-ray players—and states that "[t]he Netflix app may come pre-installed":[110]

[110] https://help.netflix.com/en/node/101653?ba=SwiftypeResultClick&q=install%20app%20browser.



How do I download the Netflix app?

Netflix is available on many devices, and depending on the type of device, the Netflix app may come pre-installed or you may need to download it.

Downloading Netflix on Smartphones and Tablets

Netflix can be downloaded from your device's app store. To install Netflix, follow the link for your device below from your smartphone or tablet.

- Download Netflix on Apple phones or tablets
- Download Netflix on Android phones or tablets
- Download Netflix on Windows phones or tablets

Using Netflix on Computers

Netflix can be accessed from your internet browser by visiting www.netflix.com and signing in or creating a new account. If you have a Windows 8 or Windows 10 computer, you can also download the Netflix app for Windows.

Using Netflix on Smart TVs, Streaming Media Players, Game Consoles, Set-top Boxes, or Blu-ray Players

Most devices provide Netflix as a pre-installed app that you can access from the main menu, or from a Netflix button on your remote. If you are unable to locate Netflix from the main menu or remote, it's possible that your device has an app store you can download the Netflix app from. If you cannot locate the app store or don't see Netflix offered, please contact your device manufacturer to learn how you can access Netflix.

For more information on devices you can use to stream Netflix, please visit devices.netflix.com.

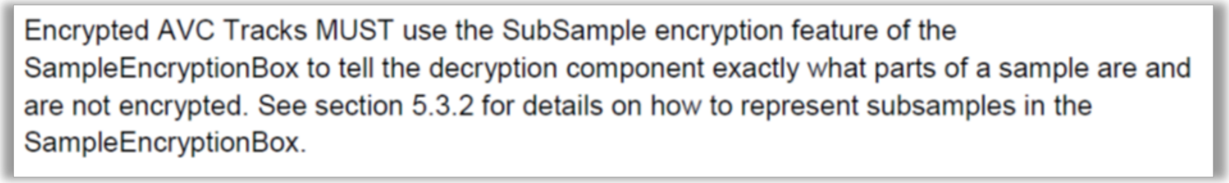
274. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . receiving a container file with video data at a parser."

a. Netflix's applications receive data from MP4 container files that contain video streams encrypted in accordance with the ISO Common Encryption Standard and Microsoft PIFF Specification. Netflix's applications include certain code— a parser—responsible for extracting information utilized in the decryption and playback of the video.

b. For example, upon information and belief, the Netflix Windows 10 App is stored locally in non-volatile memory and contains code written in JavaScript that includes a parser.

275. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . extracting portions of the container file using the parser." As noted above, the parser component in each of the Netflix Apps extracts data from received portions of MP4 container files that contain streams of video. *See* ¶ 274.

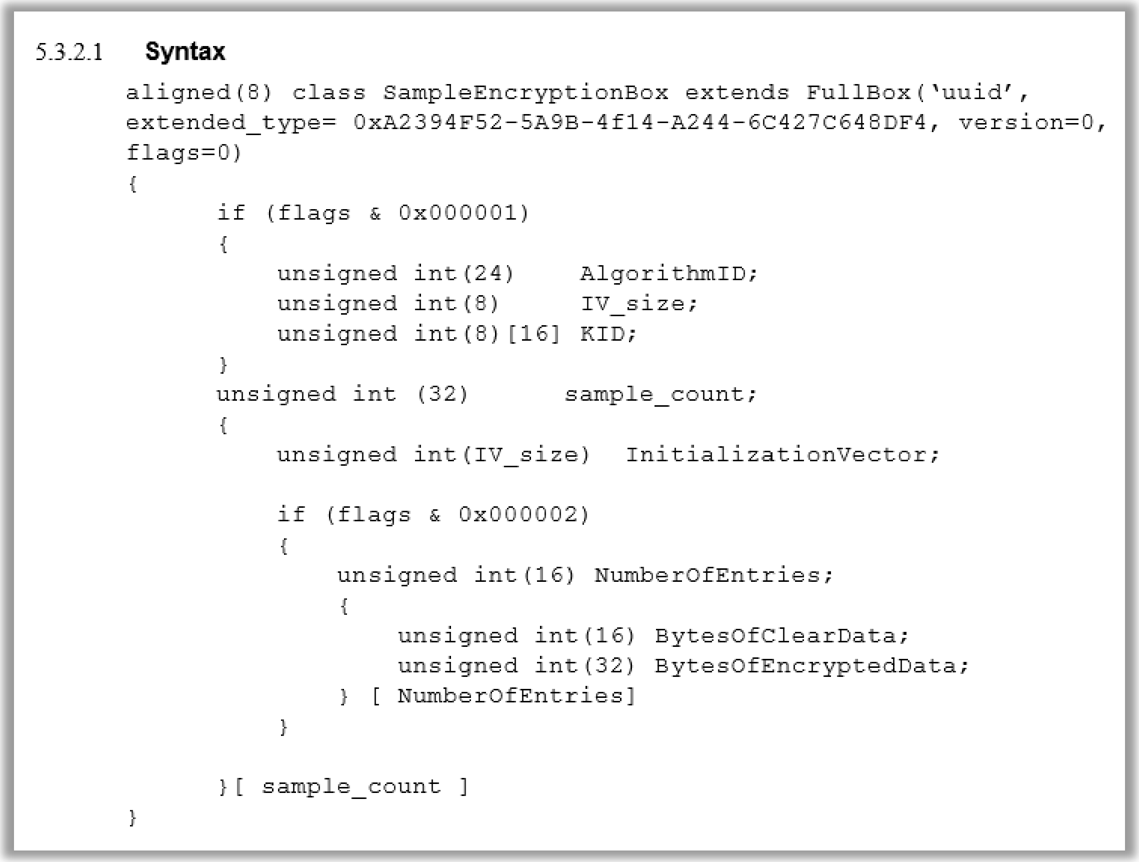
276. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . extracting portions of the container file using the parser, wherein the container file comprises: video data with a plurality of partially encrypted frames." The ISO Common Encryption Standard[111] and Microsoft PIFF Specification[112] utilized by Netflix specify the use of partially encrypted frames (referred to as sub-sample encryption).



Encrypted AVC Tracks MUST use the SubSample encryption feature of the SampleEncryptionBox to tell the decryption component exactly what parts of a sample are and are not encrypted. See section 5.3.2 for details on how to represent subsamples in the SampleEncryptionBox.

[111] ISO/IEC CD 23001-7 (3rd Ed.) at 6.

[112] Microsoft PIFF Specification at 16.



5.3.2.1 **Syntax**

```
aligned(8) class SampleEncryptionBox extends FullBox(‘uuid’,
extended_type= 0xA2394F52-5A9B-4f14-A244-6C427C648DF4, version=0,
flags=0)
{
      if (flags & 0x000001)
      {
          unsigned int(24)    AlgorithmID;
          unsigned int(8)     IV_size;
          unsigned int(8)[16] KID;
      }
      unsigned int (32)      sample_count;
      {
          unsigned int(IV_size)  InitializationVector;

          if (flags & 0x000002)
          {
              unsigned int(16) NumberOfEntries;
              {
                  unsigned int(16) BytesOfClearData;
                  unsigned int(32) BytesOfEncryptedData;
              } [ NumberOfEntries]
          }

      }[ sample_count ]
}
```

a. For example, upon information and belief, Netflix Windows 10 App streaming data shows that the retrieved data includes video data that conforms with the Microsoft PIFF Specification and includes a plurality of partially encrypted frames. MP4Box analysis shows that the downloaded exemplar videos are encoded in accordance with the H.264/AVC or H.265/HEVC standards and that portions of the encrypted frames are indicated within a PIFF Sample Encryption Box (“uuid”).

277. Netflix’s playback device further comprises “a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . extracting portions of the container file using the parser, wherein each partially encrypted frame contains encrypted portions and

unencrypted portions of data." As noted above, each partially encrypted frame includes encrypted portions and unencrypted portions. *See* ¶ 276.

278. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . extracting portions of the container file using the parser, wherein the container file comprises: a set of cryptographic information describing the encrypted portion of each partially encrypted frame." For example, the PIFF Sample Encryption Box ("uuid") in the MP4 files that contain H.265 (HEVC) or H.264 (AVC) encoded video and the Sample Encryption Box ("senc") in the MP4 files that contain VP9 encoded video received from Netflix servers by the Netflix Apps includes cryptographic information for each frame, including information describing the encrypted and unencrypted portion of each frame.

279. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . extracting portions of the container file using the parser, wherein the container file comprises: a set of cryptographic information describing the encrypted portion of each partially encrypted frame, where cryptographic information for a partially encrypted frame comprises: cryptographic material for the encrypted portion of the partially encrypted frame."

a. The Microsoft PIFF Specification and ISO Common Encryption Standard utilized by Netflix relies on the use of an AES-CTR cipher to generate a frame key to decrypt partially encrypted frames based upon cryptographic material provided in the container file. The cryptographic material for each partially encrypted frame is provided in a PIFF Sample Encryption Box ("uuid")[113] or a Sample

[113] Microsoft PIFF specification at 22.

Encryption Box ("senc")[114] in the MP4 files received from Netflix servers by the Netflix Apps.

b. For example, upon information and belief, the MP4 container file downloaded by the Netflix Windows 10 App shows that the downloaded video is encoded in accordance with the H.264 (AVC) standard and that cryptographic material are contained within a PIFF Sample Encryption Box ("uuid").

280. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . extracting portions of the container file using the parser, wherein the container file comprises: a set of cryptographic information describing the encrypted portion of each partially encrypted frame, where cryptographic information for a partially encrypted frame comprises: a block reference that identifies the encrypted portion of the partially encrypted frame." As noted above, the PIFF Sample Encryption Box ("uuid") and the Sample Encryption Box ("senc") in the MP4 files received from Netflix servers by the Netflix Apps include cryptographic information for each frame including a number of encrypted subsamples, a number of unencrypted bytes, and a number of encrypted bytes. *See* ¶ 276.

281. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . providing each partially encrypted frame, the cryptographic material for each partially encrypted frame, and the block reference for each partially encrypted frame from the parser to a video decoder."

a. To decrypt the partially encrypted streams received from Netflix's servers, the Netflix Apps provide partially encrypted frames, the

[114] ISO/IEC CD 23001-7 (3rd Ed.) at 14.

cryptographic material for each partially encrypted frame, and the block reference for each partially encrypted frame from the parser to a video decoder.[115]

b. For example, the Netflix Windows 10 App leverages Encrypted Media Extensions to configure a Content Decryption Module (CDM) to decrypt video encrypted in accordance with the ISO Common Encryption ("cenc") Standard.[116] The ISO Common Encryption Standard specifies that "[s]amples can be partially encrypted, specified by subsample information referenced by Sample Auxiliary Information Sizes Box ('saiz') and Sample Auxiliary Information Offsets Box ('saio') boxes."[117] Upon information and belief, streams downloaded or streamed to the Netflix Windows 10 App include "saiz" and "saio" boxes and that the "saio" box points to the first byte of within the PIFF Sample Encryption Box ("uuid") in accordance with the ISO Common Encryption standard.

282. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . identifying the encrypted portion of each partially encrypted frame using the block reference for each partially encrypted frame." For example, the encrypted portion of the partially encrypted frame is identified using the block reference contained within the "PIFFSubSampleEncryptionEntries" from the MP4 container files.

[115] Microsoft PIFF Specification at 20.

[116] https://w3c.github.io/encrypted-media/format-registry/stream/mp4.html.

[117] ISO/IEC CD 23001-7 (3rd Ed.) at 3-4.

283. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . deciphering a frame key for each partially encrypted frame using the cryptographic material for each partially encrypted frame to produce a frame key for each partially encrypted frame."

a. Netflix uses the AES-CTR cipher as part of its encryption method, in accordance with the ISO Common Encryption Standard and Microsoft PIFF Specification.[118]

b. The AES-CTR cipher employs a frame encryption key (that is, "key stream" output by the AES-CTR cipher) to encrypt each partially encrypted frame. The at least one frame encryption key for a given frame is deciphered according to the following process:[119]

[118] Microsoft PIFF specification at 17.

[119] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf.

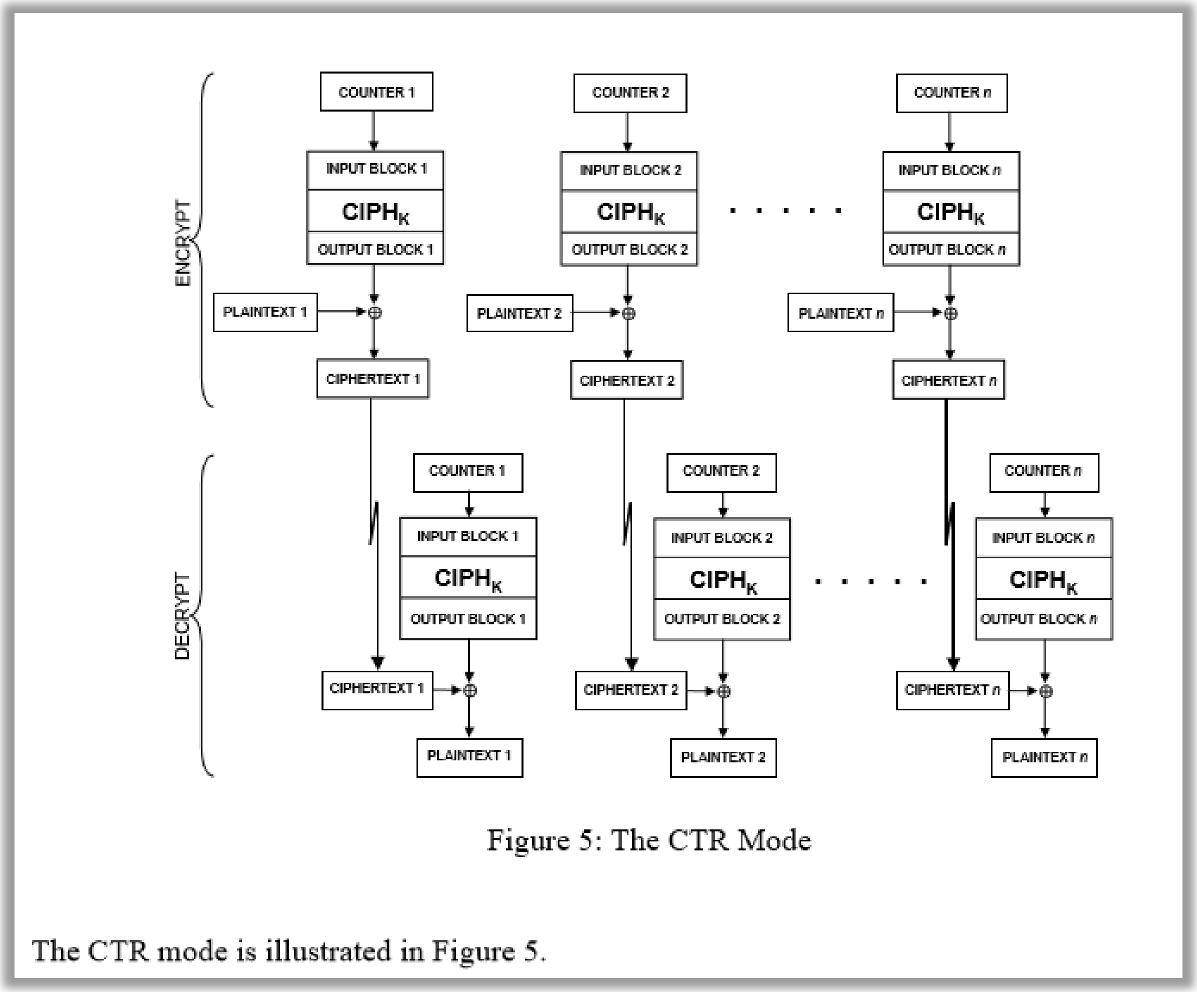


Figure 5: The CTR Mode

The CTR mode is illustrated in Figure 5.

c. The output blocks of an AES cipher in AES-CTR mode are conventionally referred to as a "key stream."[120] The "key stream" (that is, the frame key) for a particular frame is used to decrypt encrypted blocks within the frame by combining the encrypted block with the "key stream" using an exclusively-OR operation.

d. The Netflix Windows 10 App parses data obtained from MP4 files to determine the encryption method utilized. The App deciphers "key streams" using the AES cipher in a manner

[120] https://www.ietf.org/rfc/rfc3686.txt.pdf.

compatible with the encryption specification present in the Protection Scheme Info Box ("sinf") and PIFF Sample Encryption. A key referenced by the KID present in the "sinf" box is used to configure the AES cipher in AES-CTR mode to decipher the at least one frame key. Additionally, the initialization vectors specified in the PIFF Sample Encryption for each frame entry is used to configure the AES cipher in AES-CTR mode to generate the frame key.

284. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . decrypting the encrypted portion of each partially encrypted frame based upon the frame key for each partially encrypted frame using the video decoder." As noted above, the Netflix Apps decrypts the encrypted portion of the partially encrypted frame using the frame key deciphered using the AES-CTR cipher. The decryption process involves combining the frame key (namely, the key stream) with the encrypted block of data using an exclusive-OR process. *See* ¶ 283.

285. Netflix's playback device further comprises "a non-volatile storage containing a playback application for causing the set of one or more processors to perform the step[] of . . . decoding each decrypted frame for rendering on a display device using the video decoder." The decrypted frame is decoded for rendering on a display device using the video decoder.

286. Netflix directly infringes at least claim 1 when it tests its service using various playback devices.

287. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly

use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

288. Netflix has infringed, and continues to infringe, at least claim 1 of the ’486 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’486 Infringing Products in violation of 35 U.S.C. § 271(a).

289. Netflix has induced and continues to induce infringement of at least claim 1 of the ’486 patent, at least in the exemplary manner described in paragraphs 290-292, in violation of 35 U.S.C. § 271(b).

290. At least as of the date of this Complaint, Netflix knows that the ’486 patent is directed to a content security architecture that deciphers frame keys within a secure video decoder, efficiently enhancing content security. Netflix knows that it provides and specifically intends to provide an application and service to be used with a playback device that, when used as intended, practices the method recited in claim 1.

291. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, meets the limitations of claim 1.

292. At least as of the date of this Complaint, Netflix knows and specifically intends that its end users infringe claim 1, when using its application and service as intended—namely, providing playback devices and methods for deciphering frame keys within a secure video decoder, efficiently enhancing content security, as described in paragraphs 271-288.

293. Netflix’s infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix’s wrongful acts in an amount subject to proof at trial.

COUNT VIII: INFRINGEMENT OF U.S. PATENT NO. 10,225,588

294. The allegations of paragraphs 1-293 of this Complaint are incorporated by reference as though fully set forth herein.

295. Pursuant to 35 U.S.C. § 282, the ’588 patent is presumed valid.

296. Upon information and belief, Netflix directly infringes the ’588 patent by making, using, offering to sell, selling, and/or importing into the United States its Netflix service, which provides playback devices and methods for playing back alternative streams of content protected using a common set of cryptographic keys (collectively, the “Accused ’588 Infringing Products”).

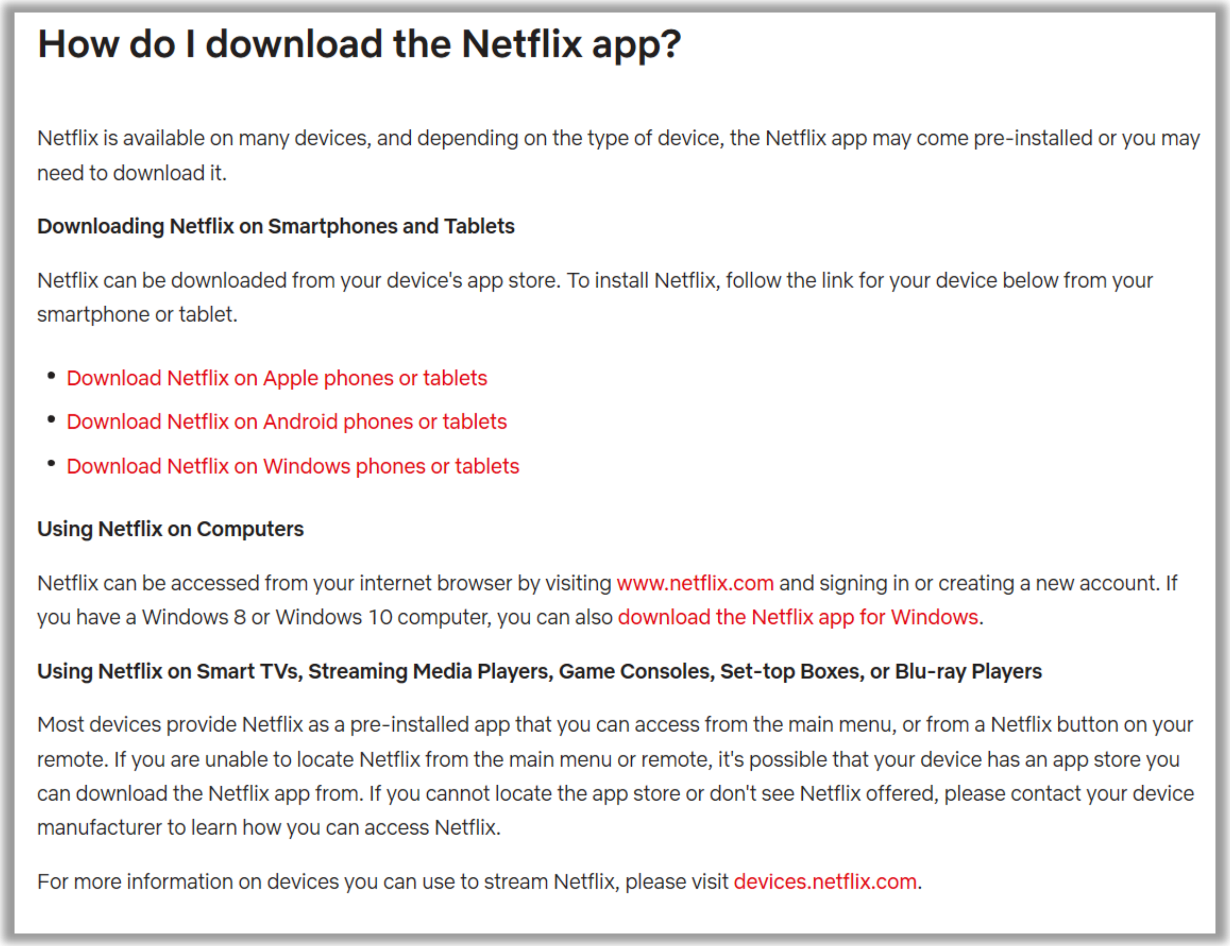
297. Upon information and belief, the Accused ’588 Infringing Products directly infringe at least claim 1 of the ’588 patent at least in the exemplary manner described in paragraphs 298-312 below.

298. Netflix provides a “playback device for playing protected content from a plurality of alternative streams” by providing applications that enable playback utilizing the MPEG-DASH Standard and the Microsoft PIFF Specification on a heterogeneous set of viewing devices.

299. Netflix’s playback device comprises “a set of one or more processors” because all playback devices that run the Netflix player application or other client applications that access the Netflix service include a set of one or more processors.

300. Netflix’s playback device further comprises “a non-volatile storage containing an application” because to play back content, the playback device uses, for example, a Netflix player application that is either pre-installed or downloaded and stored in non-volatile memory. Netflix provides details on how to access the Netflix application on numerous devices—including smartphones, tablets, computers, smart TVs, streaming media players, game consoles, set-top boxes, and Blu-ray players—and states that “[m]ost devices provide Netflix as a pre-installed app that you can access from the main menu”:[121]

[121] https://help.netflix.com/en/node/101653?ba=SwiftypeResultClick&q=%20install%20app%20browser.



How do I download the Netflix app?

Netflix is available on many devices, and depending on the type of device, the Netflix app may come pre-installed or you may need to download it.

Downloading Netflix on Smartphones and Tablets

Netflix can be downloaded from your device's app store. To install Netflix, follow the link for your device below from your smartphone or tablet.

- Download Netflix on Apple phones or tablets
- Download Netflix on Android phones or tablets
- Download Netflix on Windows phones or tablets

Using Netflix on Computers

Netflix can be accessed from your internet browser by visiting www.netflix.com and signing in or creating a new account. If you have a Windows 8 or Windows 10 computer, you can also download the Netflix app for Windows.

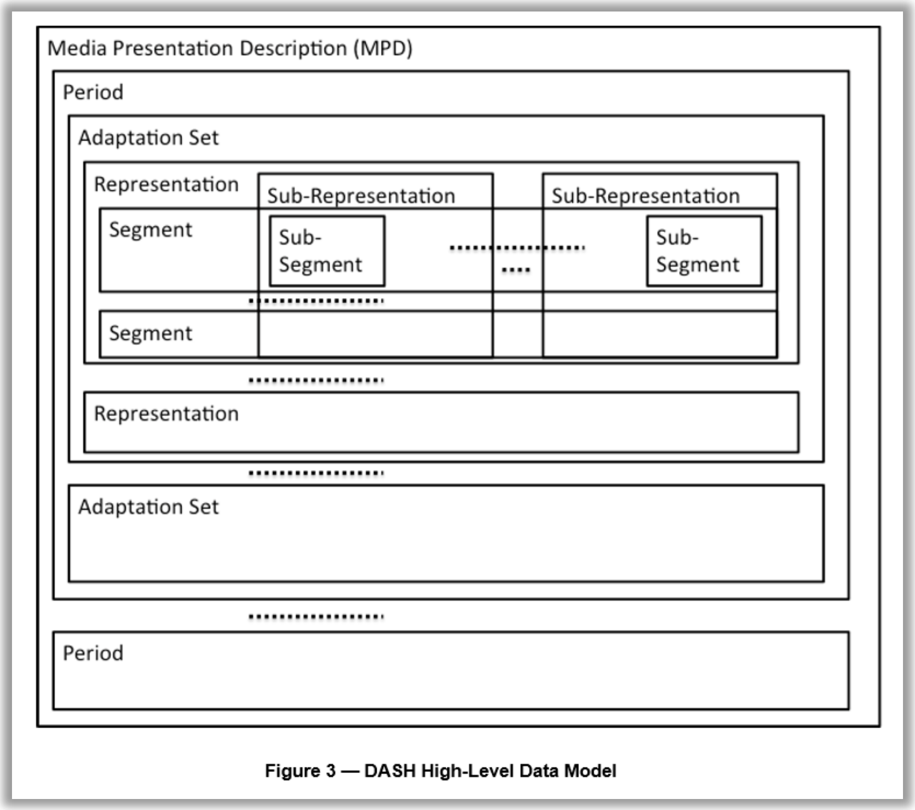
Using Netflix on Smart TVs, Streaming Media Players, Game Consoles, Set-top Boxes, or Blu-ray Players

Most devices provide Netflix as a pre-installed app that you can access from the main menu, or from a Netflix button on your remote. If you are unable to locate Netflix from the main menu or remote, it's possible that your device has an app store you can download the Netflix app from. If you cannot locate the app store or don't see Netflix offered, please contact your device manufacturer to learn how you can access Netflix.

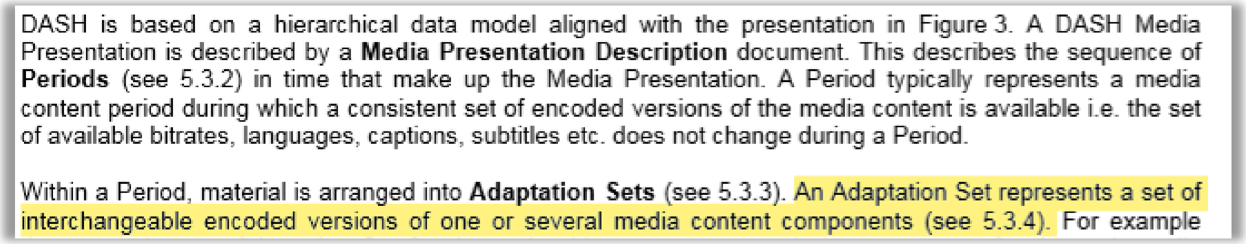
For more information on devices you can use to stream Netflix, please visit devices.netflix.com.

301. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of obtaining a top level index file identifying a plurality of alternative streams of protected video, wherein each of the alternative streams of protected video includes partially encrypted video frames that are encrypted using a set of common keys comprising at least one key, and wherein the partially encrypted video frames contain encrypted portions and unencrypted portions of data."

a. For example, the Netflix application downloads a manifest file, which is a top-level index identifying a plurality of alternative streams of protected video. Many Netflix players utilize the MPEG-DASH Standard to adaptively stream content by obtaining a top-level index file that describes multiple alternative streams of video encrypted in accordance with the ISO Common Encryption Standard or the Microsoft PIFF Specification and then selecting between the protected streams based upon network streaming conditions. The MPEG-DASH Standard includes requirements for a Media Presentation Description or MPD file (that is, a top-level index file) that includes descriptions of different Representations (namely, alternative streams) in an Adaptation Set.[122] The Netflix manifest includes the information contained within an MPD file.



[122] ISO/IEC 23009-1 (2014) Information technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and segment formats, at 9-10 (yellow highlighting added).



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

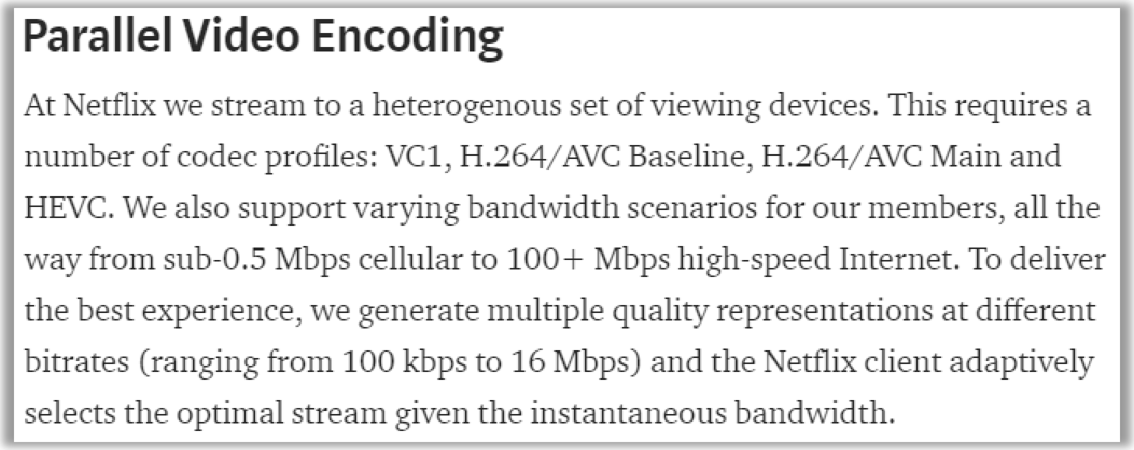
Within a Period, material is arranged into **Adaptation Sets** (see 5.3.3). An Adaptation Set represents a set of interchangeable encoded versions of one or several media content components (see 5.3.4). For example

b. In addition, "each of the alternative streams of protected video includes partially encrypted video frames that are encrypted using a set of common keys comprising at least one key, and wherein the partially encrypted video frames contain encrypted portions and unencrypted portions of data." As noted above, the plurality of streams of video are encrypted in accordance with the ISO Common Encryption Standard and the Microsoft PIFF Specification. Specifically, Netflix uses an AES-CTR cipher to partially encrypt video frames using a set of common keys comprising at least one key. Furthermore, Netflix encodes a plurality of alternative streams described in the top-level index files so that each of the plurality of alternative streams of protected video includes partially encrypted video frames that are encrypted using a set of common keys comprising at least one key. In some instances, Netflix encodes a plurality of alternative streams that each have the same resolution and encrypts them using the same key. In many instances, Netflix encrypts all streams (irrespective of resolution) using the same key.

302. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . obtaining a copy of the set of common keys." To play back streamed content, the Netflix player application obtains the key indicated by the

KID specified, for example, in the PIFF Track Encryption Boxes of the plurality of alternative protected video streams that share a common KID.

303. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . detecting streaming conditions for the playback device" because, for example, the Netflix player application detects streaming conditions and selects a stream from the plurality of alternative streams of protected video. For example, Netflix documentation clearly indicates that the Netflix player application ("client") detects streaming conditions because it "adaptively selects the optimal stream"[123] and takes "current network conditions into account" during device playback:[124]

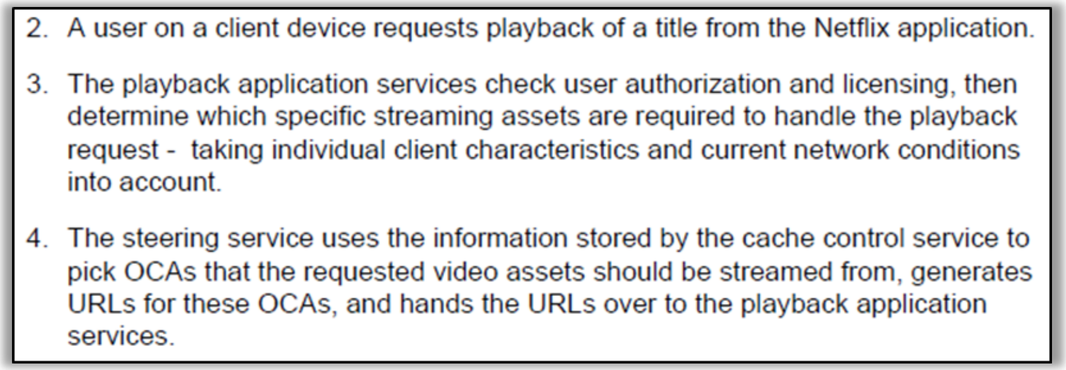


Parallel Video Encoding

At Netflix we stream to a heterogenous set of viewing devices. This requires a number of codec profiles: VC1, H.264/AVC Baseline, H.264/AVC Main and HEVC. We also support varying bandwidth scenarios for our members, all the way from sub-0.5 Mbps cellular to 100+ Mbps high-speed Internet. To deliver the best experience, we generate multiple quality representations at different bitrates (ranging from 100 kbps to 16 Mbps) and the Netflix client adaptively selects the optimal stream given the instantaneous bandwidth.

[123] https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746.

[124] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 4.

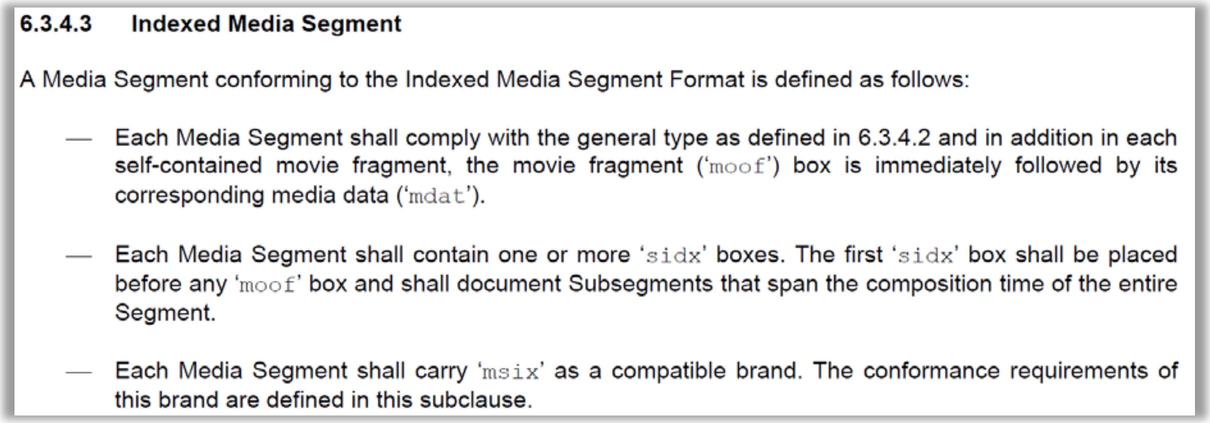


2. A user on a client device requests playback of a title from the Netflix application.
3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and current network conditions into account.
4. The steering service uses the information stored by the cache control service to pick OCAs that the requested video assets should be streamed from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

304. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . selecting a stream from the plurality of alternative streams of protected video based on the detected streaming conditions." *See* ¶¶ 301-303.

305. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file" because, for example, MP4 container files encoded by Netflix include container indexes in the form of an sidx box, which provides byte ranges for portions of a stream of protected video within the container file. For example, the MPEG-DASH Standard requires including an sidx box within the MP4 container file:[125]

[125] ISO/IEC 23009-1 at 87 (Section 6.3.4.3).



6.3.4.3 Indexed Media Segment

A Media Segment conforming to the Indexed Media Segment Format is defined as follows:

— Each Media Segment shall comply with the general type as defined in 6.3.4.2 and in addition in each self-contained movie fragment, the movie fragment (‘moof’) box is immediately followed by its corresponding media data (‘mdat’).

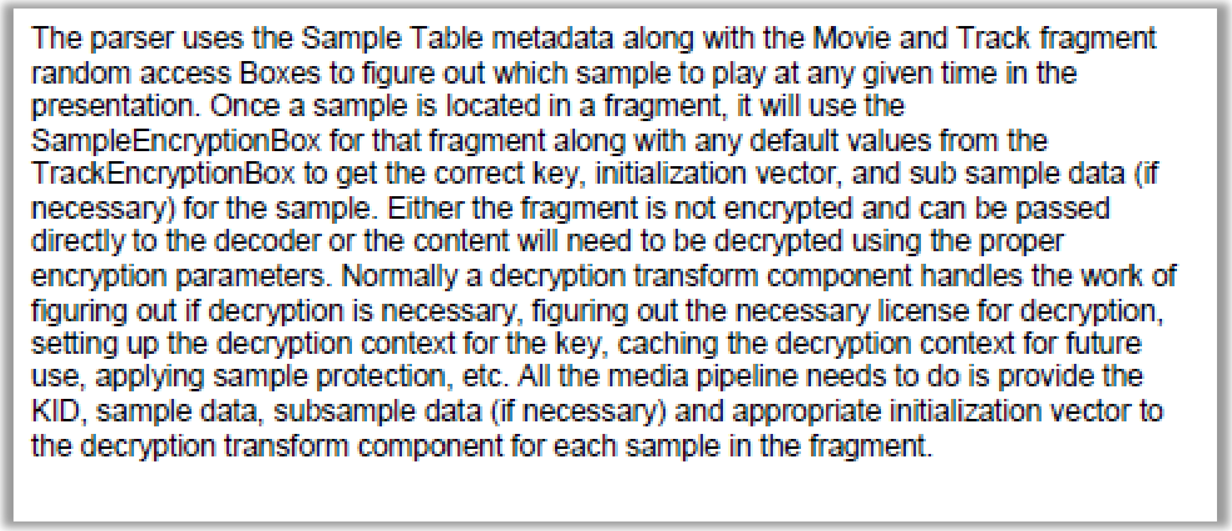
— Each Media Segment shall contain one or more ‘sidx’ boxes. The first ‘sidx’ box shall be placed before any ‘moof’ box and shall document Subsegments that span the composition time of the entire Segment.

— Each Media Segment shall carry ‘msix’ as a compatible brand. The conformance requirements of this brand are defined in this subclause.

306. Netflix’s playback device further comprises “a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . requesting portions of the selected stream of protected video based on the provided byte ranges” because Netflix applications use, for example, the sidx box to make HTTP byte range requests for content.

307. Netflix’s playback device further comprises “a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . locating encryption information that identifies encrypted portions of frames of video within the requested portions of the selected stream of protected video.” For example, to decrypt the partially encrypted streams received from Netflix, Netflix player applications locate encryption information that identifies encrypted portions of frames of video within the selected stream (for example, the received PIFF Sample and Subsample Encryption Boxes “uuid”). The process utilized is in accordance with the Microsoft PIFF Specification:[126]

[126] PIFF Specification, page 2.



The parser uses the Sample Table metadata along with the Movie and Track fragment random access Boxes to figure out which sample to play at any given time in the presentation. Once a sample is located in a fragment, it will use the SampleEncryptionBox for that fragment along with any default values from the TrackEncryptionBox to get the correct key, initialization vector, and sub sample data (if necessary) for the sample. Either the fragment is not encrypted and can be passed directly to the decoder or the content will need to be decrypted using the proper encryption parameters. Normally a decryption transform component handles the work of figuring out if decryption is necessary, figuring out the necessary license for decryption, setting up the decryption context for the key, caching the decryption context for future use, applying sample protection, etc. All the media pipeline needs to do is provide the KID, sample data, subsample data (if necessary) and appropriate initialization vector to the decryption transform component for each sample in the fragment.

308. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys" because Netflix player applications decrypt, or cause the decryption of, the encrypted portion of the partially encrypted frame (for example, the demultiplexed, encoded samples from MediaExtractor) using the common keys (for example, the common key indicated by the KID in the PIFF Track Encryption Box for the plurality of alternative video streams).

309. Netflix's playback device further comprises "a non-volatile storage containing an application for causing the set of one or more processors to perform the step[] of . . . playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video" because the Netflix application causes, or Netflix client software in conjunction with another application(s) causes, the processor(s) and the hardware elements of the client device under the processor's control to play back decrypted video.

310. Netflix directly infringes at least claim 1, at least as described, when it tests its service using various playback devices.

311. Upon information and belief, testing Netflix-compatible CE devices is critical to ensuring the success of the Netflix streaming service. Testing allows Netflix to ensure that the largest ecosystem of CE devices possible may seamlessly use the service. It further ensures that iterative versions, updates, and subsequent releases of the application and service remain compatible with CE devices.

312. Netflix has infringed, and continues to infringe, at least claim 1 of the ’588 patent in the United States by making, using, offering for sale, selling, and/or importing the Accused ’588 Infringing Products in violation of 35 U.S.C. § 271(a).

313. Netflix has induced, and continues to induce, infringement of at least claim 1 of the ’588 patent, at least in the exemplary manner described in paragraphs 314-315, in violation of 35 U.S.C. § 271(b).

314. At least as of the date of this Complaint, Netflix knows that the ’588 patent enables Netflix to offer its users an improved experience for adaptive bitrate streaming while maintaining the content security that it and other content providers require to make video content available over the internet. Specifically, the ’588 patent is directed to a DRM architecture that uses common frame encryption keys to encode alternate video streams, reducing playback stalls during adaptive bitrate streaming.

315. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, meets the limitations of claim 1, as described in paragraphs 298-312.

316. Netflix’s infringement has caused and continues to cause damage to DivX, and DivX is entitled to recover damages sustained as a result of Netflix’s wrongful acts in an amount subject to proof at trial.

JURY TRIAL DEMANDED

DivX hereby demands a trial by jury on all claims and issues so triable.

PRAYER FOR RELIEF

WHEREFORE, DivX respectfully requests that the Court:

A. Enter judgment that Netflix has directly infringed one or more claims of one or more of the DivX Patents, either literally or under the doctrine of equivalents, in violation of 35 U.S.C. § 271(a);

B. Enter judgment that Netflix has induced infringement of one or more claims of the DivX Patents in violation of 35 U.S.C. § 271(b);

C. Enter an order, pursuant to 35 U.S.C. § 284, awarding to DivX damages adequate to compensate for Netflix's infringement of the DivX Patents (and, if necessary, related accountings), in an amount to be determined at trial, but not less than a reasonable royalty;

D. Enter an order, pursuant to 35 U.S.C. § 285, deeming this to be an "exceptional case" and thereby awarding to DivX its reasonable attorneys' fees, costs, and expenses;

E. Enter an order that Netflix account for and pay to DivX the damages to which DivX is entitled as a consequence of the infringement;

F. Enter an order for a post-judgment equitable accounting of damages for the period of infringement of the DivX Patents following the period of damages established at trial;

G. Enter an order awarding to DivX pre- and post-judgment interest at the maximum rates allowable under the law; and

H. Enter an order awarding to DivX such other and further relief, whether at law or in equity, that this Court deems just and proper.

DATED: March 5, 2019

ROBINS KAPLAN LLP

By: */s/ Roman M. Silberfeld*

Roman M. Silberfeld

Roman M. Silberfeld, SBN 62783
RSilberfeld@RobinsKaplan.com
Daniel L. Allender, SBN 264651
DAllender@RobinsKaplan.com
ROBINS KAPLAN LLP
2049 Century Park East, Suite 3400
Los Angeles, CA 90067
Telephone: (310) 552-0130
Facsimile: (310) 229-5800

Christopher A. Seidl (*pro hac vice* to be filed)
CSeidl@RobinsKaplan.com
Aaron R. Fahrenkrog (*pro hac vice* to be filed)
AFahrenkrog@RobinsKaplan.com
Bryan J. Mechell (*pro hac vice* to be filed)
BMechell@RobinsKaplan.com
William E. Manske (*pro hac vice* to be filed)
WManske@RobinsKaplan.com
Shui Li (*pro hac vice* to be filed)
SLi@RobinsKaplan.com
Emily J. Tremblay (*pro hac vice* to be filed)
ETremblay@RobinsKaplan.com
Mary Pheng (*pro hac vice* to be filed)
MPheng@RobinsKaplan.com
Rajin Singh Olson (*pro hac vice* to be filed)
ROlson@RobinsKaplan.com
ROBINS KAPLAN LLP
800 LaSalle Avenue, Suite 2800
Minneapolis, MN 55402
Telephone: (612) 349-8500
Facsimile: (612) 339-4181

Christine Yun Sauer, SBN 314307
CYunSauer@RobinsKaplan.com

ROBINS KAPLAN LLP
2440 West El Camino Real, Suite 100
Mountain View, CA 94040
Telephone: (650) 784-4040
Facsimile: (650) 784-4041

ATTORNEYS FOR PLAINTIFF DIVX, LLC