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*)
CSeidl@RobinsKaplan.com
Aaron R. Fahrenkrog (*pro hac vice*)
AFahrenkrog@RobinsKaplan.com
Bryan J. Mechell (*pro hac vice*)
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

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 PSG (JCx)

FIRST AMENDED COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

Plaintiff DivX, LLC ("Plaintiff" or "DivX"), by its attorneys, for its first amended complaint ("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

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

---

[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.

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 invented technologies for video compression, transmission, playback, and security

[4] *Id.*

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.

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.

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

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 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

---

[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.*

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

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]

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

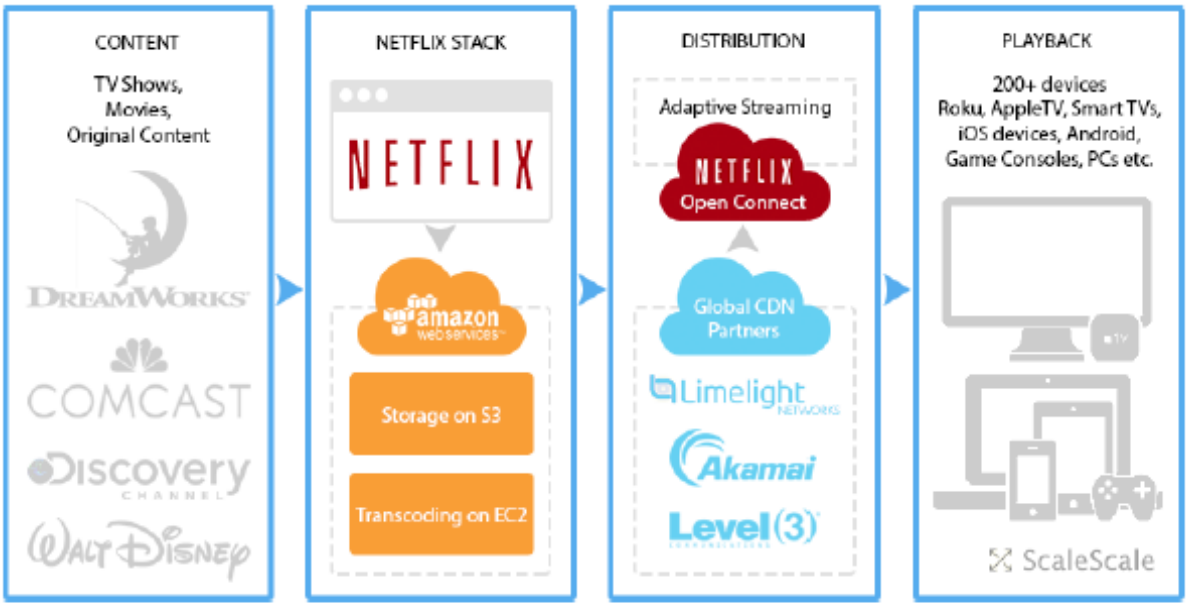
[15] Netflix, Inc., 2017 10-K, *available at* https://www.sec.gov/Archives/edgar/data/1065280/000106528018000069/q4nflx201710k.htm.

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

[17] *Id.*

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

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



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 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

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

[21] *Id.*

[22] *Id.*

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

---

[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.

[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-

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]

THE DIVX PATENTS[33]

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

---

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.*

[33] DivX files this first amended complaint pursuant to the Court's August 8, 2019 Order (Dkt. 59) granting in part Netflix's motion to dismiss under 35 U.S.C. § 101. In that Order, the Court found that DivX's initial complaint in this matter did not provide sufficient factual allegations to support arguments made by DivX in

I. Technical Background of Streaming Video

73. The DivX Patents are directed to improvements to computer systems for video streaming. Video "streaming" refers to the computing process of continuously providing digital video to an end user through a computing device.

74. Video streaming is accomplished by providing digital video files over the internet, from server computers that host (store) the video files, over the interconnected computer networks that make up the internet, to client computers (consumer devices, such as desktop computers, laptop computers, smartphones, and smart televisions) that can interpret the video files and convert them to pixels displayed on the screen during playback.

75. The ability to perform video streaming, and the level of performance that can be provided to an end user (such as high-resolution, smooth playback, without stalls or errors), depends on the computing resources of the computing devices—server computers, network computers, and client computers—used in the video streaming system. Those computing resources include the processing power of the computers, the input/output (I/O) and data transmission capabilities of the computers, and the memory (storage) available on the computers.

76. Before digital video, video was stored on analog media such as tape. Transition from analog media to digital video brought new challenges. For example, the amount of data required to represent a video in digital form at its full recorded resolution is massive. The computing resources of servers, networks, and

---

opposition to Netflix's motion to dismiss. Dkt. 59 at 16, 18-19, 20, 21. Although DivX respectfully disagrees with the Court's conclusion, DivX submits this first amended complaint to incorporate additional facts, supported by each patent's intrinsic record—its claims, specification, and file history—demonstrating that the claims of the asserted patents each recite technological improvements and/or non-routine and unconventional inventive concepts. A redline comparing this first amended complaint to the initially filed complaint (Dkt. 1) is attached as Exhibit 17.

client computers, however, are limited. Streaming digital video, therefore, requires computing techniques to reduce the amount of data that must be processed by server computers, transmitted over networks, and interpreted and converted to displayed video by client computers. These techniques are generally referred to as "encoding" (converting the data to a particular digital format) and "decoding" (translating the digital format to a format that can be rendered and displayed on a display device).

77. Video encoding and decoding rely on a computing technique called "compression" to reduce the size of the digital video files that must be processed and transmitted while simultaneously preserving sufficient playback performance and quality on the client device. Video compression employs data compression techniques specific to digital video content to reduce file sizes while maintaining playback quality. Because digital video is frequently represented as a series of still image "frames" played back quickly (for example, at a rate of 30 frames per second), video compression techniques take advantage of similarities among pixels in a single frame (spatial redundancy) and similarities among pixels across different frames (temporal redundancy) to reduce the amount of data that must be stored in the digital video file, transmitted over computer networks, and decoded by the client computer and converted to pixel data for display during playback.

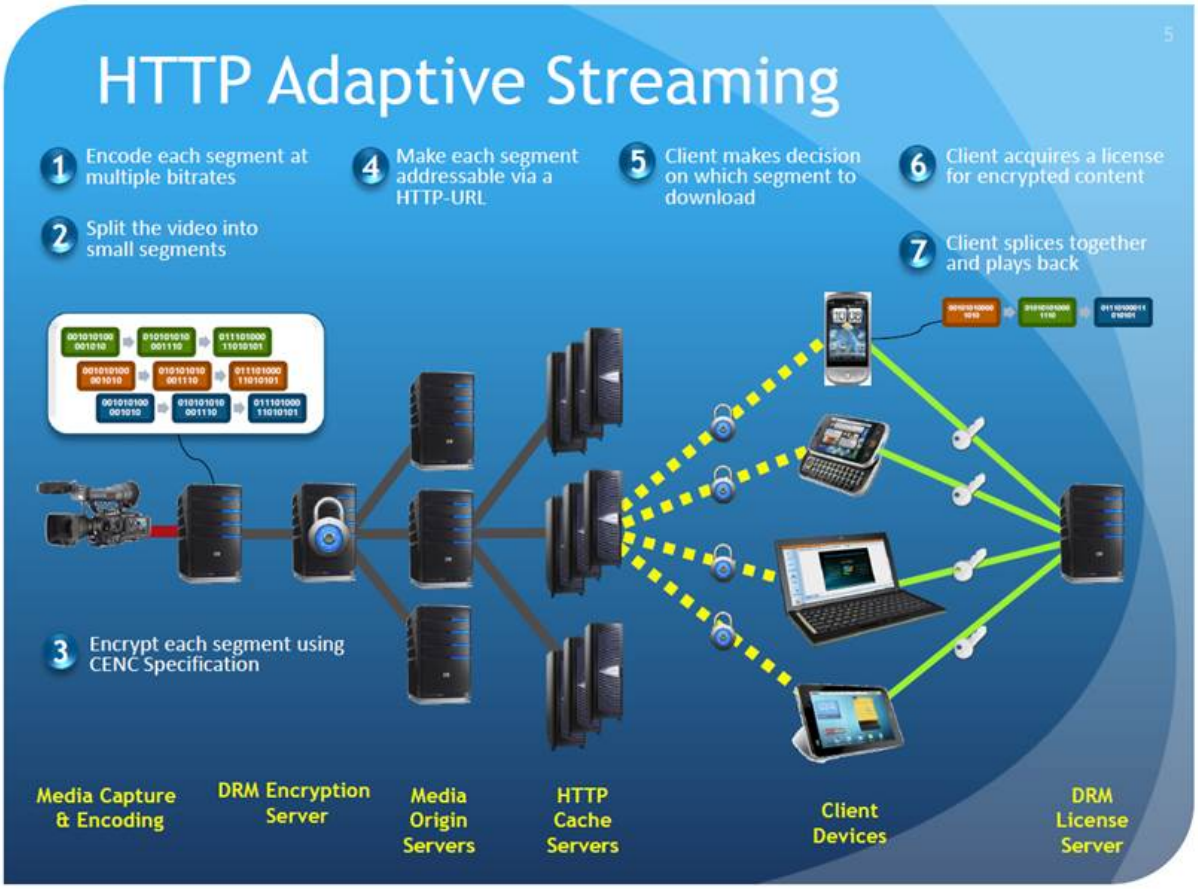
78. Video compression presents unique computing challenges different from other data compression techniques (for example, audio, text, or pictures). Consumers expect the same high-quality experience from video streamed on smartphones, televisions, and personal computers as they do from cable programming and physical media (such as DVD or Blu-ray discs). Video compression, digital video files, encoding, and decoding techniques, therefore, must overcome the technical challenge of delivering superior video quality on all types of devices in the most bandwidth-efficient way possible with the least latency (the time it takes to transmit the digital video files over the network).

79. Video compression techniques produce specific types of computer files for representing video data. These files include data structured in a certain, defined way to represent both the video data and other information required to effectively decompress, decode, and play back the video on the display of a playback device (client computer). Examples of compressed video files used for video streaming include DivX files, AVI files, MP4 files, and Matroska files.

80. Preventing piracy of digital video is another significant technical challenge for streaming video. Digital rights management (DRM) is an access control method that has been developed to protect digital media. DRM is designed to prevent the end user that has obtained digital media from modifying, copying, converting, or using the digital media in any way other than that permitted by the digital content provider. DRM often includes encryption of digital video data in specific ways using specific encryption structures and encryption and decryption mechanisms. Video streaming involves sending portions of files over a network for decryption and decoding on devices on which other software may be executing. Video streaming technology providers face unique technical challenges in providing adequate security of the video content and control over access rights while reducing the burdens on the encoding and decoding computers relating to encryption and decryption.

81. Adaptive bitrate streaming (ABS) is a specific technique used when streaming multimedia over computer networks to playback devices. ABS differs from other types of streaming because it involves detecting the streaming conditions in real time and adjusting the quality of the streamed media accordingly so the user does not experience stalls in video playback caused by changes in bandwidth or processing capabilities. For ABS, the playback server system encodes a particular video title in separate, multiple streams, at different bitrates, to be streamed consistent with the capabilities of the network and playback device, including bandwidth. If available bandwidth changes, for example, ABS allows the

device to switch to a lower-resolution stream of the same video data, which requires less data transmission and processing. This allows the video content to keep playing the video without any stall. The process of stream switching in ABS requires the ability to seek to a particular location and commence playback without access to all of the preceding portions of the file.



[34]

82. In sum, streaming digital video data presents unique technical challenges relating to video compression and content protection that affect the computing systems that encode and encrypt digital video, the digital video file types created by those computing systems, and the computing systems that process those file types to decrypt and decode the digital video to provide streaming users with a high-quality experience. DivX's patented inventions provide technical solutions, through computing improvements, to these technical challenges.

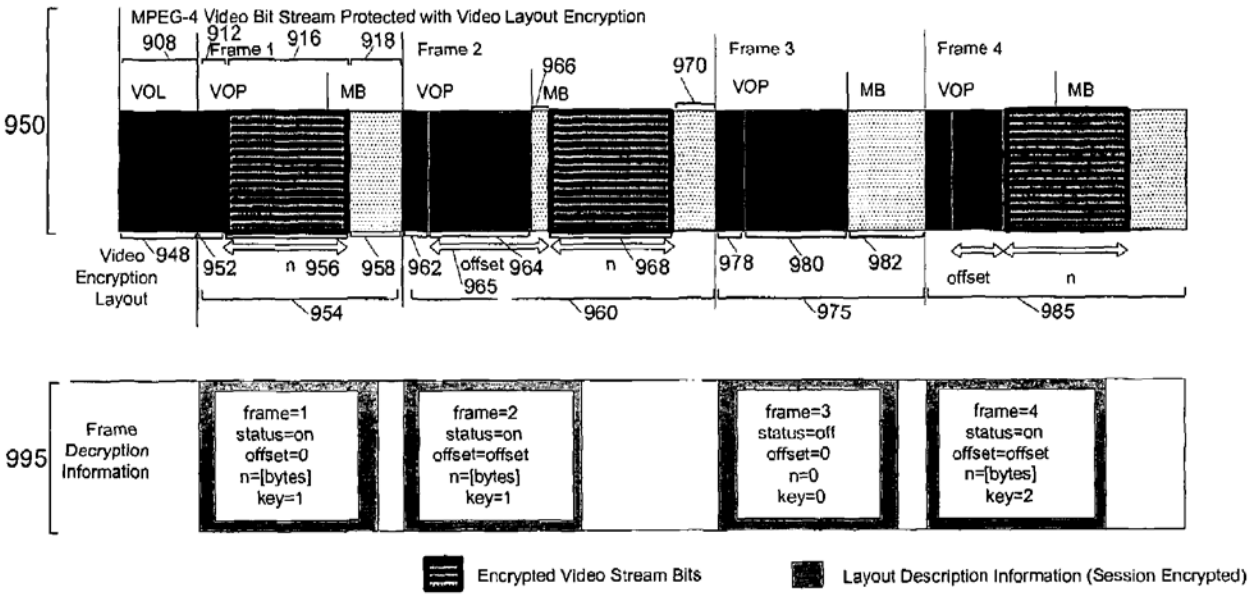
[34] https://dashif.org/docs/DASH264-v1.5.pdf.

II. The '673 Patent

83. 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.

Summary of the '673 Invention

84. The '673 claims are directed to a new structure of encrypted video data that includes partial encryption of the video frames in the stream and also includes frame decryption information synchronized in the data with the encrypted video frames. An example illustration of this new structure is provided in FIG. 9.



'673 patent, FIG. 9, 9:23-10:17. The frame decryption information, synchronized with the encrypted video frames, is illustrated in video stream 995. Certain claims of the '673 patent allow for full or partial encryption of each individual encrypted frame, and other claims require partial encryption of at least some of the encrypted frames.

85. This new structure provides "the encryption and efficient decryption of video information." *Id.* at 1:14-20. "More specifically, the present invention is directed to a method and system for generating a protected stream of compressed digital video and for decrypting the protected stream in a bounded-bandwidth fashion." *Id.* The new video stream formats, encoding and encryption processes, and decryption and decoding processes of the '673 claims provide video content security while reducing the computing resources needed to decrypt and decode the video stream. The '673 claims are directed to a partial frame encryption architecture that enables improved, more efficient streaming of encrypted video to any device, providing secure decryption without decoding. The inventions claimed 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.

Technical Problems Addressed by the '673 Invention

86. As existed in the prior art and continues to be the case today, a stream of compressed digital video content has a specific structure arranged so that it can be interpreted properly by a playback device and converted to pixels on the display screen. *See, e.g.*, *id.* at 1:24-49 (describing display of digital video as pixels), 3:3-11 (describing standards used for video compression and decompression), 5:55-6:24 (describing, with respect to FIG. 5, "types of frames within a video stream . . . formatted consistently with the MPEG-4 standard"), 7:15-28 (describing specific organization of MPEG-4 stream), 9:6-10:17 (describing, with respect to FIG. 9, "the structure of an unencrypted video stream and of a video stream encrypted in accordance with the present invention").

87. Decoding a digital video stream on a playback device, such as a television, tablet or smartphone, is "very computationally intensive, with the degree of computational intensity varying directly with the extent of compression." *Id.* at

1:63-2:9. Therefore, "[a]nything that adds to computational intensity over and above the processing overhead associated with the applicable decoding process is undesirable, since this leads to increased system complexity and expense." *Id.* In particular, "[a]ny processing of frames required in addition to decoding (*e.g.*, decryption) consumes yet further processing resources." *Id.* at 3:12-19 (describing FIG. 3).

88. Decryption adds to the computational overhead associated with decoding. "[T]he processing power necessary required [sic] to both decrypt and decode a sequence of frames" that have been encrypted is higher than "the relatively smaller amount of processing power required to decode unprotected (i.e., unencrypted) frames." *Id.* at 3:34-51 (describing FIG. 4). Further, "the maximum processing power required to both decrypt and decode a frame increases proportionally to its size." *Id.*

89. Yet, decryption and decoding are both necessary to play back video on a computing device such as a smartphone, television, or tablet. "As a consequence, adequate processing power needs to be provided to ensure that even the largest frames expected to be received may be successfully decrypted and decoded." *Id.* Some frames are larger (contain more data) than others, yet need to be decrypted and decoded at the same speed and quality as other, smaller frames. "This requirement may significantly increase system cost and complexity, even though only a relatively small percentage of received frames may necessitate use of the full extent of available peak processing power." *Id.*

90. 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. *See, e.g., id.* at 3:12-19, 3:34-51. Specifically, at the time of the '673 invention, "a need exist[ed] for an adequately secure technique for bounding the resources consumed during decryption, thereby reducing peak processing

requirements." *Id.* at 3:49-51. The '673 invention provides for these efficiencies while also providing the requisite content security.

91. The '673 patent, therefore, addresses a technical problem: allowing adequate content security while limiting the resources consumed during video decryption. *See, e.g., id.* at 3:39-51. 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., id.* at 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.

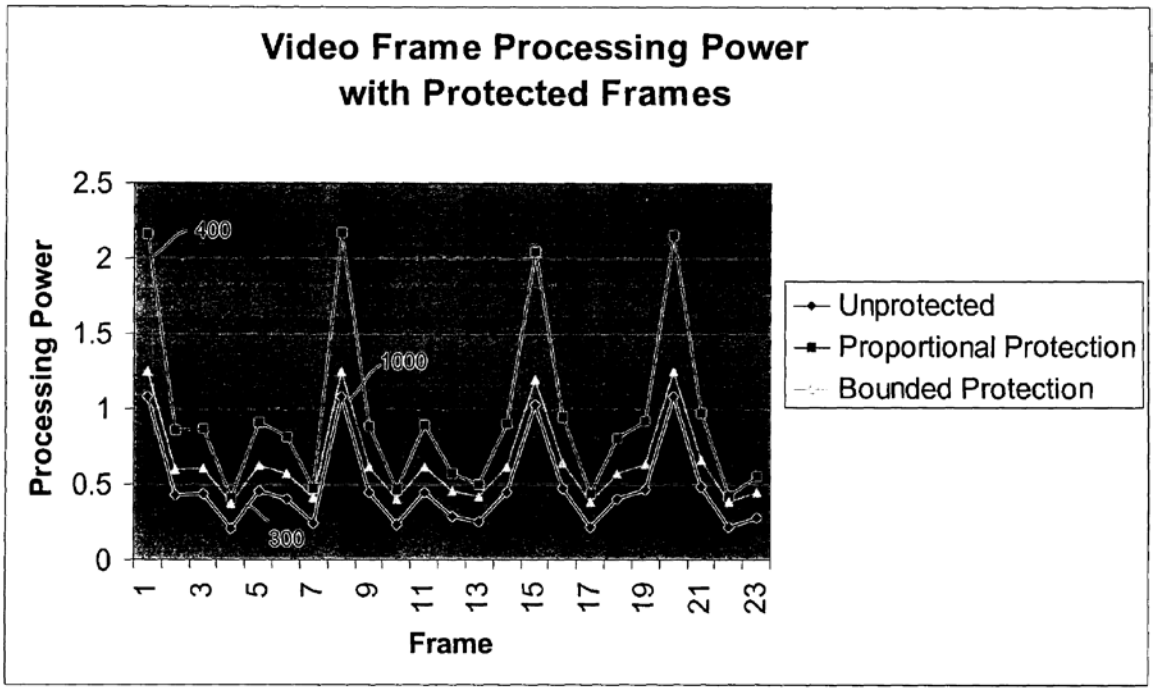
Technical Solutions and Benefits Provided by the '673 Invention

92. The '673 patent claims specific ways to solve these technical problems with compressed digital video content that provides sufficient security but requires less processing power to decrypt. The '673 claims are directed to improvements to the functionality of computer systems that perform digital video encoding, encryption, decryption, and decoding. The '673 claims are directed to a new structure of encrypted video data, how that new structure is encoded (claim 1 and dependents), how that new structure is decrypted (claim 14 and dependents), how a video encoder is configured to create that new structure (claim 21 and dependents), and how a video decoder is configured to decrypt and decode that new structure (claim 29 and dependents).

93. The new structure of encrypted video data of the '673 invention includes frame decryption information synchronized with encrypted frames in the video data. *See, e.g., id.* at 3:55-4:42; 5:25-32, 6:39-7:14 (describing FIG. 6, including new process for creating the new structure of encrypted video data), 7:15-8:42 (describing FIG. 7, including new process for generating "frame decryption information" for the new structure of encrypted video data), 8:43-9:5 (describing FIG. 8, including new process for decrypting and decoding the new structure of encrypted video data), 9:6-10:17 (describing FIG. 9, including structure of the new

video data format). In claim 1, the video data includes encryption of some but not all frames. For each encrypted frame, "at least selected portions of selected frames" are encrypted. *Id.* at 11:45-48. Claim 1 allows for full or partial encryption for each encrypted frame. Other claims of the '673 patent require partial encryption for at least some of the encrypted frames. *See, e.g.*, *id.* at 12:47-64 (claim 14), 14:18-45 (claim 29). Neither structure (encrypting full frames or partial frames), combined with frame decryption information synchronized with encrypted frames, was well-known, routine, or conventional at the time of the '673 invention.

94. Prior video data structures did not include frame decryption information synchronized in the data with encrypted frames in a video stream that includes both encrypted and unencrypted frames. Thus, "[t]he bounded encryption approach of the invention requires substantially less peak processing power (see, e.g., frames 8, 15, and 20) during the decryption process than would otherwise be required using standard encryption techniques." *Id.* at 10:18-34. "FIG. 10 provides a graphical representation of the processing power required for decryption of a digital video stream encrypted in accordance with the present invention relative to the power required for decryption of a conventionally-encrypted video stream." *Id.* at 10:18-22.



Id. at FIG. 10. The top line in the figure represents the processing power needed to decrypt and decode a fully encrypted stream, the bottom line represents the power needed to decode an unencrypted stream, and the middle line represents the power needed to decrypt and decode the new file structure of the invention—reducing the resources needed from a fully encrypted approach while providing more security than the unencrypted approach. Processes and systems for encoding, encrypting, decrypting, and decoding the new structure, therefore, also were not well-known, routine, or conventional at the time of the '673 patent.

95. The new structure of encrypted video data of the '673 invention, and the processes and systems for encoding, encrypting, decrypting, and decoding the new structure, provide technical benefits that improve the functionality and capabilities of computer systems performing these operations. *See, e.g., id.* at 9:34-10:17, FIG. 9 (describing improvement to decoding process allowing the decoder to efficiently identify, decrypt, and decode the encrypted frames),10:18-34, FIG. 10 (describing reduction in peak processing power required to decrypt, decode, and play back video on a playback device by using the invention). By encrypting only a

portion of the video stream, the new structure of encrypted video data reduces the computing resources required both for encoding and encrypting the data and for decrypting and decoding the data, as depicted, for example, in FIG. 10. Encrypting only a portion of each encrypted frame can further reduce the necessary computing resources. By synchronizing frame decryption information with the encrypted frames in the video data, the new structure of encrypted video data improves the performance of the computer system executing decryption and decoding operations, making decryption less computationally intensive and reducing errors that could be caused by a lack of synchronization.

Prosecution History of the '673 Invention

96. The claims of the '673 patent issued, among other reasons, because "the admitted prior arts taken independently or in combination, do not disclose, teach, or suggest creating 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; generating 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; and assembling 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; wherein said frame decryption information is synchronized with said set of encrypted frames into a synchronized frame decryption stream," or similar limitations. '673 File History,[35] Notice of Allowability, July 13, 2007, at 2-3 (underlining in original).

97. During prosecution, the patent examiner did not reject any claims of the '673 patent under 35 U.S.C. § 101.

[35] Cited excerpts of the '673 file history attached as Exhibit 9.

Claims Reciting the Technical Solutions of the '673 Invention[36]

98. The '673 claims recite methods and systems setting forth how to improve the creation, format, and playback of protected video streams using partial encryption and frame decryption information synchronized with the encrypted frames. Claim 1 of the '673 patent recites how to perform an improved method for producing a new and improved structure of encrypted video data:

> 1. A method for producing a protected stream of compressed video content, said method comprising:
> receiving an input stream of compressed video content containing a sequence of frames;
> generating a frame encryption key and storing the encryption key in a key table;
> creating 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;
> generating frame decryption information necessary to decrypt said set of encrypted frames including an

[36] In the August 8, 2019 Order (Dkt. 59), the Court found that "Defendants identify certain claims of the five asserted, challenged patents as representative . . . Plaintiff does not respond to this identification in its oppositions or otherwise argue that other claims in these patents have one or more limitations with distinctive significance compared to the limitations Defendants' identified claims. The Court deems Plaintiff to have waived any argument to the contrary, both for purposes of these motions and any future motions to dismiss." Dkt. 59 at 12-15. DivX does not agree that it has waived any argument that the claims discussed in prior briefing are representative of all claims in the asserted patents for purposes of assessment of the patentability of the asserted patents' claims. In this first amended complaint, DivX has pleaded facts addressing each claim of the asserted patents demonstrating that each claim recites a technological improvement and/or non-routine and unconventional inventive concept.

> encryption key pointer identifying a decryption key to be used in the decryption of each encrypted frame; and
>
> assembling 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;
>
> wherein said frame decryption information is synchronized with said set of encrypted frames into a synchronized frame decryption stream.

'673 patent, 11:39-60.

99. The claim limitations of claim 1 explain how to achieve the benefits of providing a protected video stream while reducing the processing power required for decryption of that video stream. In particular, the limitations of claim 1 produce a specific structure of a protected video stream that preserves security while reducing the processing power required for decryption and making it easier to decrypt encrypted frames because the frame decryption information is synchronized with the encrypted frames in the stream, which also reduces decryption errors where some frames are at least partially encrypted and at least some frames are not encrypted. 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.*

100. Claims 2-13 of the '673 patent depend from claim 1, and each of claims 2-13 further describe how to perform the invention's improved method for producing a new and improved structure of encrypted video data that maintains security while reducing the processing power required for decryption of that video stream. The ordered combination of elements in each of claims 2-13, in conjunction with the elements of the claims from which they depend, therefore recite

unconventional new and improved computer processes and video stream structures that were not well-understood at the time of the '673 invention.

- Claim 2 depends from claim 1 and further describes the structure of the new video stream produced by the improved method, reciting "said synchronized frame decryption stream includes references to frame encryption keys in the key table." *Id.* at 11:61-63.
- Claim 3 depends from claim 1 and further describes the structure of the new video stream produced by the improved method, reciting "said synchronized frame decryption stream includes encryption status information corresponding to each frame of said protected stream." *Id.* at 11:64-67.
- Claim 4 depends from claim 1 and further describes the structure of the new video stream produced by the improved method, reciting "said synchronized frame decryption stream includes a reference to a decryption key in the key table." *Id.* at 12:1-3.
- Claim 5 depends from claim 1 and further describes the structure of the new video stream produced by the improved method, including partial encryption of individual frames, reciting "said synchronized frame decryption stream includes intra-frame encryption offset information corresponding to each encrypted frame of said protected stream." *Id.* at 12:4-7.
- Claim 6 depends from claim 5 and further describes how the improved method produces the new video stream, including partial encryption of individual frames, reciting "parsing said input stream in order to determine frame boundaries and frame types associated with frames of said sequence of frames." *Id.* at 12:8-11.
- Claim 7 depends from claim 6 and further describes how the improved method produces the new video stream, reciting "maintaining counts

corresponding to each of said frame types, said counts and said boundaries being used to determine said intra-frame encryption offset information." *Id.* at 12:12-15.

- Claim 8 depends from claim 7 and further describes how the improved method produces the new video stream, reciting "determining sizes of said frames of said sequence of frames, said sizes also being used in determining said intra-frame offset information." *Id.* at 12:16-19.
- Claim 9 depends from claim 6 and further describes how the improved method produces the new video stream, reciting "maintaining counts corresponding to each of said frame types, said counts being used to determine when to use a new frame encryption key in said encrypting of said selected frames." *Id.* at 12:20-23.
- Claim 10 depends from claim 1 and further describes the structure of the new video stream produced by the improved method, including partial encryption of individual frames, reciting "said synchronized frame decryption stream includes information identifying a data field size to be decrypted with respect to each encrypted frame of said protected stream." *Id.* at 12:24-27.
- Claim 11 depends from claim 1 and further describes how the improved method produces the new video stream, reciting "encrypting a first consecutive number of said selected frames using a first frame encryption key and encrypting a second consecutive number of said selected frames using a second frame encryption key." *Id.* at 12:28-32.
- Claim 12 depends from claim 1 and further describes how the improved method produces the new video stream, including partial encryption of individual frames, reciting "determining a number of bytes to be encrypted within each of said selected frames based upon a

level of available processing power and a desired degradation of visual quality." *Id.* at 12:33-36.

- Claim 13 depends from claim 1 and further describes how the improved method produces the new video stream, reciting "receiving an input stream of video content containing a sequence of frames; and generating the input stream of compressed video content by applying processing techniques in accordance with an applicable encoding standard to produce a plurality of video information streams; wherein the encrypting of selected frames includes encrypting a portion of a predetermined video information stream." *Id.* at 12:37-46.

101. Claim 14 of the '673 patent recites how to perform an improved method for decrypting a new and improved structure of encrypted video data:

> 14. A method for decrypting a protected stream of compressed video content comprising:
> receiving an input stream of compressed video content containing encrypted frames and unencrypted frames;
> 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;
> obtaining an applicable frame decryption key from the received frame decryption information; and
> 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.

Id. at 12:47-64.

102. The claim limitations of claim 14 explain how to improve the decryption process by interpreting a particular structure of a protected video stream to efficiently decrypt the video stream, reducing the processing power required by the playback device while maintaining security of the video content and making it easier to decrypt selected portions of encrypted frames because the frame decryption information is synchronized with the encrypted frames in the stream. This synchronization also reduces decryption errors where some frames are partially encrypted and at least some frames are not encrypted. Claim 14 of the '673 patent, therefore, recites a novel solution of decrypting a protected video stream using decryption information synchronized 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.*

103. Claims 15-20 of the '673 patent depend from claim 14, and each of claims 15-20 further describe how to perform the invention's improved method for decrypting a new and improved structure of encrypted video data that maintains security while reducing the processing power required for decryption of that video stream. The ordered combination of elements in each of claims 15-20, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer processes and video stream structures that were not well-understood at the time of the '673 invention.

- Claim 15 depends from claim 14 and further describes the structure of the new video stream for decryption by the improved method, reciting "said input stream and said synchronized frame decryption stream collectively comprise a protected video stream, said synchronized

frame decryption stream being synchronized with said encrypted frames within said input stream." *Id.* at 12:65-13:2.

- Claim 16 depends from claim 14 and further describes the structure of the new video stream for decryption by the improved method, reciting "said synchronized frame decryption stream includes encryption status information corresponding to each of said encrypted frames." *Id.* at 13:3-5.
- Claim 17 depends from claim 14 and further describes the structure of the new video stream for decryption by the improved method, reciting "said synchronized frame decryption stream includes a reference to a frame decryption key for each of said encrypted frames." *Id.* at 13:6-8.
- Claim 18 depends from claim 14 and further describes the structure of the new video stream for decryption by the improved method, including partial encryption of individual frames, reciting "said synchronized frame decryption stream includes intra-frame encryption offset information corresponding to each of said encrypted frames." *Id.* at 13:9-12.
- Claim 19 depends from claim 14 and further describes the structure of the new video stream for decryption by the improved method, including partial encryption of individual frames, reciting "said synchronized frame decryption stream includes size information identifying a data field size to be decrypted with respect to each of said encrypted frames." *Id.* at 13:13-16.
- Claim 20 depends from claim 14 and further describes how the improved method decrypts the new video stream, reciting "decrypting a first consecutive number of said encrypted frames using a first frame decryption key and decrypting a second consecutive number of said

encrypted frames using a second frame decryption key." *Id.* at 13:17-21.

104. Claim 21 of the '673 patent recites how an improved encrypting digital video encoder produces a new and improved structure of encrypted video data:

> 21. An encrypting digital video encoder comprising:
> a video processing unit for generating a plurality of input data streams in response to a sequence of uncompressed video frames;
> an entropy compression unit for creating, based upon said plurality of input data streams, compressed video content containing a sequence of compressed frames; and
> a video encryption module configured to generate a table of encryption keys;
> wherein the video encryption module is also configured to create a set of encrypted frames by encrypting at least selected portions of selected frames of said sequence of compressed frames using said frame encryption keys in accordance with a frame encryption function;
> wherein the video encryption module is also configured to transform said sequence of compressed frames into a protected video stream containing at least the set of encrypted frames, the unencrypted frames and a synchronized frame decryption stream necessary to decrypt said set of encrypted frames;
> wherein said synchronized frame decryption stream includes encryption key pointers identifying encryption a decryption key to be used in the decryption of each encrypted frame.

Id. at 13:22-45.

105. The claim limitations of claim 21 explain how to achieve the benefits of providing a protected video stream while reducing the processing power required for decryption of that video stream. In particular, the limitations of claim 21 produce a specific structure of a protected video stream produced by an encrypting digital video encoder. The protected video stream produced by the encoder claimed in claim 21 preserves security while reducing the processing power required for decryption and makes it easier to decrypt encrypted frames because the frame decryption information necessary to decrypt the encrypted frames is synchronized with the video frames and includes an encryption key pointer to identify the necessary decryption key, which also reduces decryption errors where some frames are at least partially encrypted and at least some frames are not encrypted. Claim 21 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.*

106. Claims 22-28 of the '673 patent depend from claim 21, and each of claims 22-28 further describe how the invention's improved video encoder computing system is configured to encode a new and improved structure of encrypted video data that maintains security while reducing the processing power required for decryption of that video stream. The ordered combination of elements in each of claims 22-28, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer configurations and video stream structures that were not well-understood at the time of the '673 invention.

- Claim 22 depends from claim 21 and further describes the structure of the new video stream encoded using the improved encoder, reciting "said protected video stream is comprised of an encrypted video

stream including said set of encrypted frames and unencrypted ones of said compressed frames, said synchronized frame decryption stream being synchronized with said encrypted video stream." *Id.* at 13:46-51.

- Claim 23 depends from claim 22 and further describes the structure of the new video stream encoded using the improved encoder, reciting "said synchronized frame decryption stream includes encryption status information corresponding to each frame of said encrypted video stream." *Id.* at 13:52-55.
- Claim 24 depends from claim 22 and further describes the structure of the new video stream encoded using the improved encoder, reciting "said synchronized frame decryption stream also includes, intra-frame encryption offset information, and data field size decryption information corresponding to each frame of said encrypted video stream." *Id.* at 13:56-60.
- Claim 25 depends from claim 21 and further describes how the improved encoder encodes the new video stream, including partial encryption of individual frames, reciting "said video encryption module is operative to parse said sequence of frames in order to determine frame boundaries and frame types associated with individual frames of said sequence of frames." *Id.* at 13:61-14:2.
- Claim 26 depends from claim 25 and further describes how the improved encoder encodes the new video stream, reciting "said video encryption module is operative to maintain counts corresponding to each of said frame types, said counts and said boundaries being used to determine intra-frame encryption offset information." *Id.* at 14:3-7.
- Claim 27 depends from claim 25 and further describes how the improved encoder encodes the new video stream, reciting "said video encryption module is operative to maintain counts corresponding to

each of said frame types, said counts being used to determine when to create new encryption keys used in generating ones of said encrypted frames." *Id.* at 14:8-12.

- Claim 28 depends from claim 21 and further describes how the improved encoder encodes the new video stream, reciting "the entropy compression unit is configured to encrypt a predetermined one of said video information streams." *Id.* at 14:13-16.

107. Claim 29 of the '673 patent recites how an improved decrypting digital video encoder is configured to decrypt and decode a new and improved structure of encrypted video data:

> 29. A decrypting digital video decoder comprising:
> a video decryption module configured to receive a protected input stream of compressed video content containing at least a set of encrypted frames and synchronized frame decryption stream, said synchronized frame decryption stream being necessary for decrypting said set of encrypted frames so as to form a set of decrypted frames;
> wherein the video decryption module is further configured to obtain an applicable frame decryption key from the received frame decryption stream;
> wherein the video decryption module is further configured to generate the set of decrypted frames by decrypting selected portions of the encrypted frames in accordance with said frame decryption stream, which identifies the specific portions of the frames to be decrypted and the applicable frame decryption key;

> an entropy decompression unit for creating, based at least in part upon said set of decrypted frames, a plurality of video data streams; and
>
> a video processing unit for generating an output stream of uncompressed video content in response to said plurality of video data streams;
>
> wherein said synchronized frame decryption stream includes encryption key pointers identifying an applicable decryption key to be used in the decryption of each encrypted frame.

Id. at 14:18-45.

108. These claim limitations of claim 29 explain how to improve digital video decoders and the decryption process by interpreting a particular structure of a protected video stream to efficiently decrypt the video stream, reducing the processing power required by the playback device while maintaining security of the video content and making it easier to decrypt encrypted frames because the frame decryption information, which includes encryption key pointers, is synchronized with the encrypted frames in the stream, which also reduces decryption errors where some frames are partially encrypted and at least some frames are not encrypted. Claim 29 of the '673 patent, therefore, recites a novel solution of configuring a digital video decoder to decrypt a protected video stream using decryption information synchronized 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.*

109. Claims 30-32 of the '673 patent depend from claim 29, and each of claims 30-32 further describe how the invention's improved video decoder computing system is configured to decode a new and improved structure of

encrypted video data that maintains security while reducing the processing power required for decryption of that video stream. The ordered combination of elements in each of claims 30-32, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer configurations and video stream structures that were not well-understood at the time of the '673 invention.

- Claim 30 depends from claim 29 and further describes the structure of the new video stream decoded using the improved decoder, reciting "said protected input stream is comprised of an encrypted video stream including said set of encrypted frames and unencrypted frames, said synchronized frame decryption stream being synchronized with said encrypted video stream." *Id.* at 14:46-50.
- Claim 31 depends from claim 30 and further describes the structure of the new video stream decoded using the improved decoder, reciting "said synchronized frame decryption stream includes encryption status information corresponding to each frame of said encrypted video stream." *Id.* at 14:51-54.
- Claim 32 depends from claim 30 and further describes the structure of the new video stream decoded using the improved decoder, including partial encryption of individual frames, reciting "said synchronized frame decryption stream also includes intra-frame encryption offset information, and data field size decryption information corresponding to each frame of said encrypted video stream." *Id.* at 14:55-59.

III. The '651 Patent

110. 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, 2004.

Summary of the '651 Invention

111. The '651 claims are directed to improvements to digital video compression and, in particular, to improved methods of deblocking reconstructed digital video frames. '651 patent, 1:15-16. A deblocking filter smooths the boundary lines between blocks of pixels in a frame of digital video that appear when compressed video is reconstructed in anticipation of playback. Deblocking smooths pixilation. The new deblocking methods efficiently reduce image flaws (such as pixilation) that can occur during video playback because the video data displayed on the screen has been compressed (encoded), which results in some data loss, and then decompressed (decoded). 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 and more accurate video encoding, decoding, and reconstruction for playback, creating resource savings that make 4K streaming attainable and providing a higher-quality streaming video experience.

Technical Problems Addressed by the '651 Invention

112. The '651 patent addresses a technical problem. "Digital video sequences are composed of frames of pixels, where the characteristics of the pixels are represented using digital information." *Id.* at 1:17-19. Compression reduces the amount of data required to represent a video sequence. *Id.* at 1:19-21. Known compression techniques use "characteristics that commonly occur within video sequences to achieve significant reductions in the amount of digital data required to encode a video sequence." *Id.* at 1:21-24. Known encoding techniques—for example, the Motion Picture Expert Group's MPEG-4 standard—divides frames into blocks of pixels and uses information regarding the pixels within a block to encode that block. *Id.* at 1:25-29. 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.*, *id.* at 1:27-34. For example, block-based encoding degrades the quality of the reconstructed image because "[t]reating adjacent blocks separately . . . can result in artifacts at block boundaries when an encoded video frame is reconstructed." *Id.* at 1:29-31. Viewers often observe "artifacts" as pixelated video:

Norkin, *et al.*, *HEVC Deblocking Filter*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, at 1752 (Dec. 2012), *available at* https://www.researchgate.net/publication/260665382_HEVC_deblocking_filter (image with "deblocking turned off").

113. To overcome this problem, the computing system can use a "deblocking filter" when reconstructing compressed digital video to produce better image quality. *See, e.g.*, *id.* at 1:29-34. Known encoding and decoding techniques apply deblocking filters to pixels surrounding block boundaries, to smooth the appearance of the reconstructed video frame and to remove the artifacts that compression and predictive coding leave behind. *Id.* at 1:31-34. But those deblocking filters are inefficient and require large amounts of processing power and time to implement. The MPEG-4 standard, for example, "involves applying the

MPEG-4 deblocking filter to each row of pixels at vertical block boundaries and each column of pixels at horizontal block boundaries." *Id.* at 1:65-67. Each filter application requires a separate decision regarding whether and how to filter at each row and each column. *Id.* at 1:67-2:15. The deblocking filter addresses a single row or column—effectively eight pixels—at a time. *Id.* at 2:22-44 (determining "[t]he filter that is applied to the chrominance[37] or luminance[38] values of the four pixels on either side of a block boundary (i.e. v_1-v_8)").

114. In addition, 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, was and continues to be critical. *See, e.g., id.* at 10:27-33 ("When a reconstructed image includes a boundary that lies within a detailed region of the image, the quality of the image can be improved by applying a level of filtering appropriate to the level of detail. In one embodiment, the amount of smoothing applied by a deblocking filter along a block boundary is inversely proportional to the level of detail of the image in the region being filtered."). 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.

Technical Solutions and Benefits Provided by the '651 Invention

115. The '651 patent claims specific ways to solve this problem with improved methods of inspecting block boundaries within a reconstructed video frame 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. The '651 claims are directed to improvements to the functionality of computer systems that perform

---

[37] Color information.

[38] Achromatic or black-and-white, referring to intensity or brightness in color.

digital encoding and decoding. The '651 claims are directed to a new, multidimensional deblocking filter and methods providing how to apply that filter (claim 1 and dependents).

116. The new deblocking filter method of the '651 patent assesses the level of detail at the block boundary "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." *Id.* at 13:12-18 (claim 1); *see also id.* at 8:38-51, FIG. 3. This multidimensional deblocking filter was not well-understood, routine, or conventional at the time of the '651 invention. Prior deblocking filters were incapable of assessing the level of detail of a *region* of a video frame, which is often referred to as multidimensional video deblocking. Prior deblocking filters were not multidimensional and computationally efficient.

117. The new deblocking filter of the '651 patent provides technical benefits that improve the functionality and capabilities of computer systems that deblock reconstructed video frames in anticipation of playback. The '651 claims are directed to a deblocking filter that analyzes the level of detail in a multidimensional region of pixels surrounding a block boundary, rather than analyzing a single row or single column in one pass. *Id.* at 7:65-8:5. The filter is more efficient and more accurate than those known in the art and used in the streaming video context, as it does not proceed row by row only and column by column only, assessing each in isolation one after another.

Prosecution History of the '651 Invention

118. The claims of the '651 patent issued at least because they recite a multidimensional deblocking filter: "determining 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 the block boundary," or similar limitations, which were not found in the prior art. '651 File History,[39] Applicant Reply to Office Action of February 15, 2011, Aug. 15, 2011, at 12; *see also* Notice of Allowability, Nov. 15, 2011 (allowed without examiner comment). Unlike the '651 patent's multidimensional deblocking filter, the prior art recited only "one dimensional" deblocking filters. '651 File History, Applicant Reply to Office Action of February 15, 2011, Aug. 15, 2011, at 13.

119. During prosecution, the patent examiner did not reject any claims of the '651 patent under 35 U.S.C. § 101.

Claims Reciting the Technical Solutions of the '651 Invention

120. Claim 1 of the '651 patent recites how to perform an improved method of deblocking a reconstructed video frame:

> A method of deblocking a reconstructed video frame, comprising:
>
> identifying a boundary between two blocks of the reconstructed video frame;
>
> determining 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;

[39] Cited excerpts of the '651 file history attached as Exhibit 10.

selecting a filter to apply to predetermined pixels on either side of the block boundary based upon the determined level of detail.

'651 patent, 13:8-22.

121. 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. It recites a method of applying a multidimensional deblocking filter to analyze and filter a reconstructed video frame more efficiently, over a region, than previous row-by-row and column-by-column filters.

122. Claims 2-20 of the '651 patent depend from claim 1, and each of claims 2-20 further describes the improved, multidimensional deblocking filter that allows for greater decoding efficiency and an improved, smoother playback experience. The ordered combination of elements in each of claims 2-20, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer processes and deblocking filters that were not well-understood at the time of the '651 invention.

- Claim 2 depends from claim 1 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting "the determination of the level of detail of the reconstructed video frame in a region in which the block boundary is located further comprises taking the sum of the absolute difference of at least some of the pixels within a set of pixels surrounding the block boundary." *Id.* at 13:23-27.
- Claim 3 depends from claim 2 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting, "the block boundary is a horizontal block boundary; the set of pixels is a

block of pixels that is divided by the horizontal block boundary; the sum of the absolute difference is taken for each vertically adjacent pair of pixels in each column of the block of pixels, except the pair of pixels that are separated by the block boundary." *Id.* at 13:28-35.

- Claim 4 depends from claim 2 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame may be applied, reciting, "the set of pixels is an 8x8 block that is evenly divided by the horizontal block boundary." *Id.* at 13:36-37.
- Claim 5 depends from claim 4 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i+1,j} - v_{i,j}|$$

where i=1 to 7 and i≠4, j=1 to 8[, and] where: $v_{i,j}$ is the chrominance of a pixel in row i and column j of the 8x8 block of pixels." *Id.* at 13:38-50.

- Claim 6 depends from claim 4 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i+1,j} - v_{i,j}|$$

where i=1 to 7 and i≠4, j=1 to 8[, and] where: $v_{i,j}$ is the luminance of a pixel in row i and column j of the 8x8 block of pixels." *Id.* at 13:51-63.

- Claim 7 depends from claim 3 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame may be applied, reciting "the set of pixels is a 4x8 block that is evenly divided by the horizontal block boundary." *Id.* at 13:64-65.
- Claim 8 depends from claim 7 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i+1,j} - v_{i,j}|$$

where i=1 to 7 and i≠4, j=1 to 4[, and] where: $v_{i,j}$ is the chrominance of a pixel in row i and column j of the 4x8 block of pixels." *Id.* at 13:66-14:10.

- Claim 9 depends from claim 7 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i+1,j} - v_{i,j}|$$

where i=1 to 7 and i=4, j=1 to 4[, and] where: $v_{i,j}$ is the luminance of a pixel in row i and column j of the 4x8 block of pixels." *Id.* at 14:11-23.

- Claim 10 depends from claim 2 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting "the block boundary is a vertical block boundary; the set of pixels is a block of pixels that is divided by the vertical block boundary; the sum of the absolute difference is taken for each adjacent pair of pixels in each row of the block of pixels, except the pair of pixels that are separated by the block boundary." *Id.* at 14:24-32.
- Claim 11 depends from claim 10 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is applied, reciting "the set of pixels is an 8x8 block that is evenly divided by the vertical block boundary." *Id.* at 14:33-35.
- Claim 12 depends from claim 11 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i,j+1} - v_{i,j}|$$

where i=1 to 8, j=1 to 7 and j≠4[, and] where: $v_{i,j}$ is the chrominance of a pixel in row i and column j of the 8x8 block of pixels." *Id.* at 14:36-49.
- Claim 13 depends from claim 11 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the

reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i,j+1} - v_{i,j}|$$

where i=1 to 8, j=1 to 7 and j≠4[, and] where: $v_{i,j}$ is the luminance of a pixel in row i and column j of the 8x8 block of pixels." *Id.* at 14:50-62.

- Claim 14 depends from claim 10 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is applied, reciting "the set of pixels is an 8x4 block that is evenly divided by the vertical block boundary." *Id.* at 14:63-65.
- Claim 15 depends from claim 14 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i,j+1} - v_{i,j}|$$

where i=1 to 4, j=1 to 7 and j≠4[, and] where: $v_{i,j}$ is the chrominance of a pixel in row i and column j of the 8x4 block of pixels." *Id.* at 14:66-15:9.

- Claim 16 depends from claim 14 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the determination of the level of detail involves calculating the following sum:

$$\sum_i \sum_j |v_{i,j+1} - v_{i,j}|$$

where i=1 to 4, j=1 to 7 and j≠4[, and] where: $v_{i,j}$ is the luminance of a pixel in row i and column j of the 8×4 block of pixels." *Id.* at 15:10-21.

- Claim 17 depends from claim 1 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting "selecting a filter to apply to predetermined pixels on either side of the block boundary based upon the determination of the level of detail comprises comparing the determined level of detail to a threshold." *Id.* at 15:22-26.
- Claim 18 depends from claim 17 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting "the threshold varies depending upon the quantizer used in the encoding of the blocks at the block boundary." *Id.* at 16:1-3.
- Claim 19 depends from claim 17 and further describes the improved, multidimensional deblocking filter and how the improved method for deblocking a reconstructed video frame is performed, reciting "selecting a filter to apply to predetermined pixels on either side of the block boundary based upon the determination of the level of detail further comprises: when the level of detail exceeds the threshold, selecting a filter to apply to predetermined pixels; wherein selection of a filter comprises determining the detail of the image in the region of the pixels being filtered." *Id.* at 16:4-12.
- Claim 20 depends from claim 19 and further describes the improved, multidimensional deblocking filter and how the improved method for

deblocking a reconstructed video frame is performed, reciting a particular equation used to determine the level of detail of the reconstructed video frame: "the block boundary is a horizontal block boundary; and determining the detail of the image in the region of the pixels being filtered comprises calculating the following expression with respect to a column of pixels divided by the horizontal block boundary:

$$(|v_{-2} - v_{-1}| * 3 + |v_{-1} - v_1| * 2 + |v_1 - v_2| * 3)//8$$

where: v_i is the chrominance of the pixel i pixels from the horizontal block boundary." *Id.* at 16:13-23.

IV. The '792 Patent

123. 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.

Summary of the '792 Invention

124. The '792 claims are directed to a new, improved multimedia file structure to facilitate sending digital video over networks to playback devices. The new file structure improves the playback device's ability to navigate and play back the file's digital video content. '792 patent, Abstract, 1:20-21, 1:38-40, 1:48-53. The new multimedia file of the '792 patent includes a specific dual-index structure, including an abridged index, that allows the playback device to more quickly access index information and, as a result, navigate and more efficiently request the video content during streaming. The new structure enables playback features that 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 index that improves the user playback experience by enabling chunk-

based ABS, "trick play," and "fast start" functionality. "Trick play" can include digital video implementations of features such as scene skipping, rewind, and fast forward; digital implementations of these differ significantly from traditional analog implementations of rewind and fast forward for video stored on tape, requiring technical solutions including specific file structures and processing operations to mimic the videotape operations that users expect. "Fast start" describes technical features allowing a digital video to begin playing nearly immediately upon the user making a selection.

Technical Problems Addressed by the '792 Invention

125. The '792 patent's new multimedia file addresses a technical problem. Originally, multimedia, like video, transmitted over the internet had a single index for all of 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-intensive 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 could enable desirable playback features while reducing the computing resources, and associated delays, required to obtain and process the index.

126. The technical problem addressed by the '792 invention specifically relates to the structure of video files. Multimedia files containing video must be structured in a specific way so that they can be decoded, navigated, and played back by a variety of computing devices, including "a lap-top computer . . . digital set-top boxes, desk-top computers, game machines, [and] CE devices." *Id.* at 5:6-23. For example, multimedia files containing video can include header information, metadata, video frames ("the 'movi' list chunk"), and an index, arranged in a

particular order. *Id.* at 5:32-6:44, FIG. 2.0 (illustrating structure of multimedia file). The "index chunk" of the file "can be implemented using data structures that reference the location within the file of each of the 'data' chunks in the 'movi' list chunk," providing a full index. *Id.* at 22:18-28. This full index "is created by reading the location within the 'movi' list chunk of each 'data' chunk." *Id.* at 46:4-13.

Technical Solutions and Benefits Provided by the '792 Invention

127. The '792 patent claims specific ways to solve the problems associated with having to request and navigate the multimedia file's full index, specifically, by providing an improved multimedia file with an *abridged* index. The '792 claims are directed to improvements to the functionality of computers that decode and play back digital video content stored in multimedia container files. The '792 claims are directed to decoders that can decode this new multimedia file (claim 1 and dependents and claim 15 and dependents) and encoders configured to encode the new multimedia file (claim 9 and dependents).

128. The new multimedia file structure of the '792 invention includes a specific dual-index structure—an abridged index and a complete index. *See, e.g.*, *id.* at 15:9-21, 16:26-36 (describing packaging the new multimedia file to include an abridged index, for example, "before the 'movi' list chunk" including the video frame data, and explaining that the dual-index structure "can enable rapid location of a specific video frame" and "can enable a device to start playing and performing other functions, such as fast forward, rewind and scene skipping, prior to the downloading of the [full index]"), 15:9-21 (explaining that the second index, or abridged index, is different than the complete index because "the 'index' chunk does not include information concerning every 'data' chunk in the 'movi' list chunk," and "[t]ypically, the 'index' chunk includes information concerning a subset of the 'data' chunks"). This structure was not well-understood, routine, or conventional at the time of the '792 invention. Prior multimedia file structures did

not include the dual-index structure that enables chunk-based ABS and easier, more efficient playback.

129. The new multimedia file structure of the '792 invention provides technical benefits that improve the functionality of playback devices decoding and playing back the video content contained therein. *See, e.g.*, *id.* at 48:21-49:42 (describing that the abridged index "can be used to skip frames either in a regular fashion (such as during fast forwarding or rewinding) or in an irregular fashion (such as when skipping between scenes or chapters)"). By packaging a multimedia file with an abridged index in addition to a full index, the '792 patent's new multimedia file structure solves the technical problems and resource-intensive computing issues associated with complex video files. *See, e.g.*, *id.* at 15:10-16:36, 48:21-49:42. These technical solutions enable desired video playback features like starting video immediately and the ability to fast forward, rewind, and skip scenes. *See, e.g.*, *id.* at 16:26-29, 48:21-37.

Prosecution History of the '792 Invention

130. The '792 invention improved upon prior art multimedia files by "provid[ing] two separate indexes that enable[] trick play functionality upon processing and playback of the multimedia file." '792 File History,[40] Amendment and Remarks, June 29, 2010, at 8.

131. In the new multimedia file structure of the '792 patent, with "two separate indexes," "the presence of the second index can increase the speed with which . . . playback devices, such as many consumer electronics devices, can commence playback with trickplay [sic] functionality. In the absence of the earlier [i.e., abridged] index, [a] playback device will typically parse through an entire video track to locate the first index before it can commence playback with trick play functionality (typically fast forward, rewind, and scene skipping). The time

[40] Cited excerpts of the '792 file history attached as Exhibit 11.

taken to parse through the video track can introduce a significant delay in the commencement of playback." *Id.* at 8-9.

132. Multimedia files before the '792 invention did not provide "separate full or subset indexes" as recited in the '792 claims. *Id.* at 9. A multimedia file that "includes two indexes in which one index includes location information regarding each video frame and the second index includes location information a subset of video frames and that proceeds the video frames and the first index . . . is particularly useful, e.g., for trick play applications." '792 File History, Amendment and Remarks, Dec. 21, 2009, at 9 (underlining in original).

133. The new multimedia file structure of the '792 patent, including "redundant information such as two separate indexes is counter intuitive to compression designs common in [the] technology space of multimedia container formats, where the objective is typically to reduce the size of a file as much as possible to accommodate transfer over a network, such as the Internet." '792 File History, Amendment and Remarks, June 29, 2010, at 9.

134. Claim 1 of the '792 patent and the claims that depend from claim 1 issued, among other reasons, because they recite "the unique distinct feature 'a separate second index that includes information indicative of the location within the file of a subset of the encoded video frames, the separate second index located prior to the series of encoded video frames and the first index, the first and second indexes enabling trick play functionality,'" which was not found in the prior art. '792 File History, Notice of Allowability, Mar. 1, 2013, at 2 (underlining in original).

135. Claim 9 of the '792 patent (prosecuted as application claim 5) and the claims that depend from claim 9 issued, among other reasons, because they recite "the unique distinct feature 'wherein the processor is configured to generate an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames and to encode a multimedia file including the abridged

index, the at least one sequence of encoded video frames, and a furl [sic] index so that the abridged index is located within the multimedia file is provided prior to the series of encoded video frames, the first and second indexes enabling trick play functionality,'" which was not found in the prior art. *Id.* at 3 (underlining in original).

136. Claim 15 of the '792 patent (prosecuted as application claim 9) and the claims that depend from claim 15 issued, among other reasons, because they recite "the unique distinct feature 'wherein the processor is configured 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 pray [sic] functionality.'" *Id.* (underlining in original).

137. The prior art identified during prosecution of the '792 patent did not disclose "'location information' for 'a subset of video frames,'" and, therefore, did not teach "a separate second index that includes information indicative of the location within the file of a subset of the encoded video frames" as recited in claim 1 of the '792 patent, and the claims that depend from claim 1, or similar limitations in the other claims of the '792 patent. '792 File History, Response to Office Action, May 16, 2012, at 8-9. The prior art did not teach a first index that includes information indicative of the location within the file and characteristics of each encoded video frame; and a separate second index that includes information indicative of the location within the file of a subset of the encoded video frames, the separate second index located prior to the series of encoded video, as recited in claim 1 of the '792 patent, and the claims that depend from claim 1, or similar limitations in the other claims of the '792 patent. *See* '792 File History, Response to Office Action, Sept. 19, 2011, at 8-11.

138. The prior art identified during prosecution of the '792 patent taught away from "a multimedia file including the 'first index' and the 'second separate

index' recited in claim 1" and the claims that depend from claim 1, and similar limitations in the other claims of the '792 patent. '792 File History, Amendment and Remarks, Dec. 17, 2010, at 13-14. The prior art also taught away from an "abridged index . . . located within the multimedia file prior to the series of encoded video frames" as recited in claim 9 of the '792 patent and the claims that depend from claim 9. *Id.* at 15 (underlining in original).

139. The prior art identified during prosecution of the '792 patent did not disclose that "the multimedia file includes . . . an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames," as recited in claim 15 of the '792 patent and the claims that depend from claim 15. *Id.* at 16 (underlining in original). Instead, the prior art taught away from "including 'an abridged index' within a 'multimedia file' so that a decoder receiving the multimedia file can 'locate a particular encoded video frame within the multimedia file using the abridged index.'" *Id.* at 16-17.

140. The prior art identified during prosecution of the '792 patent did not disclose "an index that includes information indicative of locations within the entire multimedia file and characteristics of each encoded video frame." '792 File History, Amendment and Remarks, Dec. 21, 2009, at 9-11 (underlining in original).

141. The prior art identified during prosecution of the '792 patent did not disclose "a separate second index or that the second index includes information indicative of the location within the file of a subset of the encoded video frames." '792 File History, Amendment and Remarks, Apr. 27, 2009, at 8 (underlining in original). The prior art also did not disclose "an abridged index that references a subset of the encoded video frames and a processor that generates such an abridged index." *Id.* at 11-12, 14 (underlining in original).

142. During prosecution, the patent examiner rejected a subset of the pending claims of the '792 patent under 35 U.S.C. § 101. '792 File History, Office Action, Jan. 26, 2009, at 2-4. The applicants overcame the rejection by amending

the rejected claims to recite improved decoders for decoding the improved multimedia file format of the invention. '792 File History, Amendment and Remarks, Apr. 27, 2009, at 2-3, 5-6, 7. The patent examiner did not raise a rejection under § 101 after the amendment. *See* '792 File History, Office Action, July 21, 2009 (no § 101 rejection).

Claims Reciting the Technical Solutions of the '792 Invention

143. Claim 1 of the '792 patent recites how an improved decoder is configured to decode a new and improved multimedia file:

> 1. A decoder for decoding a multimedia file comprising at least one video track and at least one audio track, the decoder comprising:
> a processor; and
> memory having a multimedia file including:
> a series of encoded video frames;
> a first index that includes information indicative of the location within the file and characteristics of each encoded video frame; and
> a separate second index that includes information indicative of the location within the file of a subset of the encoded video frames, the separate second index located prior to the series of encoded video frames and the first index, the first and second indexes enabling trick play functionality.

'792 patent, 51:31-45.

144. Claim 1 of the '792 patent, therefore, recites a decoder for decoding a new multimedia file with an unconventional structure according to the invention, enabling trick play functionality. *Id.* Claim 1 discloses the new multimedia file structure, with an abridged index. The presence of the abridged index allows the

claimed decoder to more easily seek within the video content. It can request the much smaller index and navigate it more easily, enabling trick play functionality and rapid fast start of playback. Claim 1 recites a novel solution for more efficiently processing certain improved multimedia files to enable desirable playback features in a manner that was not well-understood, routine, or conventional at the time of the '792 invention.

145. Claims 2-8 of the '792 patent depend from claim 1, and each of claims 2-8 further describes how the new and improved multimedia file of the invention is structured for decoding and enabling better performance of the decoder. The ordered combination of elements in each of claims 2-8, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer multimedia files that were not well-understood at the time of the '792 invention.

- Claim 2 depends from claim 1 and further describes the structure of the new multimedia file, reciting that "the second index includes at least one tag that references an encoded video frame in the subset of encoded video frames," and "each tag comprises: the location within the file of the referenced encoded video frame; [and] the frame number of the encoded video frame in the sequence of encoded video frames." *Id.* at 51:46-54.
- Claim 3 depends from claim 2 and further describes the structure of the new multimedia file, reciting that the multimedia file contains "at least one audio track," "each tag further comprises a reference to a portion of at least one of the audio tracks," and "the portion . . . referenced accompanies the encoded video frame referenced by the tag." *Id.* at 51:55-61.
- Claim 4 depends from claim 2 and further describes the structure of the new multimedia file, reciting that "each tag further comprises a

reference to information located within the first index," and "the information referenced in the first index is indicative of the location within the file and characteristics of the encoded video frame referenced by the tag." *Id.* at 51:62-67.

- Claim 5 depends from claim 1 and further describes the structure of the new multimedia file, reciting that "the second index includes a plurality of tags," and "each tag references encoded video frames that are evenly spaced throughout the encoded video frames." *Id.* at 52:1-4.
- Claim 6 depends from claim 1 and further describes the structure of the new multimedia file, reciting that "the second index includes a plurality of tags," and "each tag references encoded video frames that are spaced at least ten seconds apart." *Id.* at 52:5-7.
- Claim 7 depends from claim 1 and further describes the structure of the new multimedia file, reciting that "the second index includes a plurality of tags," and "each tag includes chunk offset information, index offset information, video frame identifiers and audio track identifiers." *Id.* at 52:8-11.
- Claim 8 depends from claim 1 and further specifies the "trick play functionality" enabled by the new multimedia file: "at least one of fast forward, rewind and scene skipping." *Id.* at 52:12-14.

146. Claim 9 of the '792 patent recites how an improved encoder is configured to encode a new and improved multimedia file:

> 9. An encoder for encoding a multimedia file comprising at least one video track and at least one audio track, the encoder comprising:
>
> a processor;
>
> 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; wherein the processor is configured to generate an abridged index that references a subset of the encoded video frames in the sequence of encoded video frames and to encode a multimedia file including the abridged index, the at least one sequence of encoded video frames, and a full index so that 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.

Id. at 52:16-32.

147. Claim 9 of the '792 patent recites an encoder for encoding a new 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.* The presence of the abridged index allows a playback device to more easily seek within the content. It can request the much smaller index and navigate it more easily, enabling trick play functionality. And encoding the abridged index before the sequence of encoded video frames makes it even easier for a playback device to locate the abridged index to navigate the file's content. Claim 9 recites a novel solution for more efficiently processing certain improved multimedia files to enable desirable playback features in a manner that was not well-understood, routine, or conventional at the time of the '792 invention.

148. Claims 10-14 of the '792 patent depend from claim 9, and each of claims 10-14 further describes how the new and improved multimedia file of the invention is structured during the encoding process. The ordered combination of elements in each of claims 10-14, in conjunction with the elements of the claims

from which they depend, therefore recite unconventional computer encoding operations for new and improved computer multimedia files that were not well-understood at the time of the '792 invention.

- Claim 10 depends from claim 9 and further describes the structure of the new multimedia file and how the encoder encodes that structure, reciting that "the processor is configured to generate a complete index that references all of the encoded video frames in the sequence of encoded video frames," and "each reference to an encoded video frame in the abridged index includes a reference to the reference to that frame in the complete index." *Id.* at 52:33-40.
- Claim 11 depends from claim 9 and further describes the structure of the new multimedia file, reciting that "each reference to an encoded video frame in the abridged index includes the sequence number of the encoded video frame." *Id.* at 52:41-43.
- Claim 12 depends from claim 11 and further describes the structure of the new multimedia file and how the encoder encodes that structure, reciting that "the processor is configured to include in each reference to an encoded video frame a reference to a location within at least one sound track." *Id.* at 52:44-46.
- Claim 13 depends from claim 9 and further describes the structure of the new multimedia file and how the encoder encodes that structure, reciting that "the processor is configured to insert key frames when one of the processor detects a scene change and a threshold interval of video frames is exceeded without the processor detecting a scene change." *Id.* at 52:47-50.
- Claim 14 depends from claim 9 and further specifies the "trick play functionality" enabled by the new multimedia file: "at least one of fast forward, rewind and scene skipping." *Id.* at 52:51-53.

149. Claim 15 of the '792 patent recites how an improved decoder is configured to decode a new and improved multimedia file:

> 15. A decoder for decoding multimedia comprising at least one video track and at least one audio track, the decoder comprising:
> a processor configured to decode multimedia;
> wherein the multimedia includes:
> a sequence of encoded video frames;
> a complete index referencing each encoded video frame in the sequence of encoded video frames;
> an abridged index referencing a subset of the encoded video frames in the sequence of encoded video frames;
> wherein the processor is configured 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.

Id. at 52:54-53:3.

150. Claim 15 of the '792 patent, therefore, recites a decoder for decoding a new multimedia file with an unconventional structure according to the invention, enabling trick play functionality. *Id.* Claim 15 discloses the new multimedia file structure, with an abridged index. The presence of the abridged index allows the claimed decoder to more easily seek within the content. It can request the much smaller index and navigate it more easily, enabling trick play functionality and rapid fast start of playback. Claim 15 recites a novel solution for more efficiently processing certain improved multimedia files to enable desirable playback features

in a manner that was not well-understood, routine, or conventional at the time of the '792 invention.

151. Claims 16-23 of the '792 patent depend from claim 15, and each of claims 16-23 further describes how the new and improved multimedia file of the invention is structured for decoding and how the decoder decodes that structure, enabling better performance of the decoder. The ordered combination of elements in each of claims 16-23, in conjunction with the elements of the claims from which they depend, therefore recite unconventional computer decoding operations for new and improved computer multimedia files that were not well-understood at the time of the '792 invention.

- Claim 16 depends from claim 15 and further describes how the decoder decodes the structure of the new multimedia file, reciting "the processor is configured to locate reference information in the complete index using the abridged index." *Id.* at 53:4-6.
- Claim 17 depends from claim 15 and further describes the structure of the new multimedia file, reciting "the multimedia file includes at least one audio track accompanying the sequence of encoded video frames," and "each reference to an encoded video frame in the abridged index includes a reference to a portion of at least one of the video tracks." *Id.* at 53:7-12.
- Claim 18 depends from claim 15 and further describes how the decoder decodes the structure of the new multimedia file, reciting "the processor is configured to identify a desired encoded video frame; determine the encoded video frame that is closest to the desired video frame in the abridged index; and display an encoded video frame." *Id.* at 53:13-17.
- Claim 19 depends from claim 18 and further describes the structure of the new multimedia file and how the decoder decodes the structure of

the new multimedia file, reciting "each reference in the abridged index to an encoded video frame also includes a reference to the portion of the complete index that refers to that encoded video frame; and wherein the processor configured to display an encoded video frame, further comprises: the processor uses the reference to the encoded video frame in the abridged index that is closest to the desired encoded video frame to locate that encoded frame within the complete index; the processor searches in the complete index for the desired encoded video frame; and the processor displays the desired encoded video frame." *Id.* at 52:18-54:9.

- Claim 20 depends from claim 19 and further describes the structure of the new multimedia file, reciting that "the closest frame is the closest preceding frame in the sequence to the desired frame." *Id.* at 54:10-11.
- Claim 21 depends from claim 18 and further describes how the decoder decodes the structure of the new multimedia file, reciting that "the processor configured to an encoded video frame further comprises the processor displaying the encoded video frame that is determined to be closest to the desired video frame." *Id.* at 54:12-15.
- Claim 22 depends from claim 15 and further describes how the decoder decodes the structure of the new multimedia file, reciting that "the processor is configured to locate and playback the sequence of encoded video frame without receiving the complete index." *Id.* at 54:16-18.
- Claim 23 depends from claim 15 and further specifies the "trick play functionality" enabled by the new multimedia file: "at least one of fast forward, rewind and scene skipping." *Id.* at 54:19-21.

V. The '920 Patent

152. 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.

Summary of the '920 Invention

153. The '920 claims are directed to improvements to security and access control for digital video content distributed to playback devices. '920 patent, Abstract, 1:19-44, 1:48-54, 6:14-36, 6:50-61, 7:27-40, 10:44-12:14, 13:48-16:4. The '920 invention applies a new encryption and decryption scheme for digital video content. It incorporates multiple layers of encryption and, specifically, an active user encryption key that is stored on the playback device and is required for decryption, to enhance security of the video content and control over which devices and users can play back encrypted content.

154. 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. The content security provided by the '920 inventions also allows Netflix to obtain and offer its users a library of high-quality video content. Moreover, upon information and believe, that content security is important to producers of content, including studios, and Netflix's assurances of security to these content makers is important to Netflix's ability to obtain the rights to stream such content.

Technical Problems Addressed by the '920 Invention

155. 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.*, *id.* at 1:25-29. This can be accomplished by issuing "keys" to authorized

users to unlock the content. *See, e.g.*, *id.* at 1:29-32. 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 playback device manufacturers.

156. In addition, video streaming service providers, and content providers from whom the service providers obtain video content, also face technical challenges in restricting playback rights to particular users who are authorized to use the service and in controlling the keys issued to those authorized users so that they are not "leaked" to unauthorized users. *See, e.g.*, *id.* at 11:44-67, 13:48-65 (describing improvements to key revocation and rotation).

Technical Solutions and Benefits Provided by the '920 Invention

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

158. The '920 claims are directed to improvements to the functionality of computer systems that perform digital video decryption, decoding, and playback. The '920 claims are directed to a new digital video encryption format, how that new format is decrypted, decoded, and played back (claim 1 and dependents), and how a playback device is configured to decrypt, decode, and play back the new format (claim 10 and dependents).

159. The new video encryption format described in the '920 invention includes multiple levels of encryption using separate keys; decryption and playback require the use of an active user key stored on the playback device. Prior video

encryption formats did not incorporate multiple keys, including an active user key that must be stored on the playback device for decryption and play back. This new video encryption format, and the methods and devices used to decrypt and play back video encrypted in this new format, therefore were not well-known, routine, and conventional at the time of the '920 invention.

160. The new video encryption format of the '920 invention and the methods and systems used to decrypt and play back video encrypted in this new format provide technical benefits that improve the functionality and capabilities of computer systems performing these operations. By requiring decryption of video data using multiple keys, and specifically requiring decryption using an active user key stored on the playback device, the new video encryption format increases the security of the video data, reduces the likelihood of unauthorized access and use of that data, and enables content providers and video streaming service providers to better control access to the content by revoking or retiring keys. *See, e.g.*, *id.* at 1:32-34 (describing improved security using multiple keys), 6:14-28 (describing use of user encryption keys unique to a device or user), 10:44-11:27, FIG. 5 (describing improved security of content and improved control over user access rights using "user encryption keys," which can be tied to "information about the user requesting the content"), 11:44-67, FIG. 6 (explaining that encryption using "user encryption key(s)" enables key revocation or retirement), 13:48-65, FIG. 8 (describing decryption of content encrypted in the new video encryption format).

Prosecution History of the '920 Invention

161. The claims of the '920 patent issued, among other reasons, because they recite "obtaining 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," or similar limitations, which were not found in the prior art. '920 File History,[41] Office Action Response, Feb. 3, 2015, at 7-8.

162. During prosecution, the patent examiner did not reject any claims of the '920 patent under 35 U.S.C. § 101. The '920 patent issued on November 10, 2015, after the U.S. Supreme Court's decision in *Alice Corp. Pty Ltd. v. CLS Bank Int'l*, 573 U.S. 208 (2014).

Claims Reciting the Technical Solutions of the '920 Invention

163. The '920 claims recite methods of decrypting and decoding and devices configured to decrypt, decode, and play back encrypted content that improve the security of the content and also improve control over user access rights by video streaming service providers. Claim 1 recites how to perform an improved method for decoding the new digital video encryption format of the invention:

> 1. 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 method comprising:
>
> obtaining encrypted content using a playback device, where the content includes frames of video and at least a

[41] Cited excerpts of the '920 file history attached as Exhibit 12.

> portion of a plurality of frames of video are encrypted using at least one frame encryption key;
>
> obtaining 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;
>
> decrypting one of the one or more copies of the content encryption key using the playback device and the active user encryption key; and
>
> playing back frames of the encrypted content using the playback device, where playing back frames of the encrypted content further comprises:
>
> identifying any portions of a frame that are encrypted;
>
> identifying the frame encryption key used to encrypt the identified portions of the frame;
>
> decrypting the identified frame encryption key using the decrypted content encryption key;
>
> decrypting the encrypted portions of the frame using the decrypted identified frame encryption key; and
>
> decoding the unencrypted frame of video.

'920 patent, 16:49-17:14. The limitations of claim 1 enable the benefits of the '920 invention of enhanced content security and enhanced access control that are improvements over prior video encryption formats. Claim 1, therefore, recites a novel solution for improving the security of digital content and user access control in a manner that was not well-understood, routine, or conventional at the time of the '920 patent.

164. Claims 2-9 of the '920 patent depend from claim 1, and each of claims 2-9 further describes how to perform an improved method for decrypting and decoding the new digital video encryption format of the invention that enhances content security by binding active encryption keys to a user, allowing secure streaming. The ordered combination of elements in each of claims 2-9, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer processes and video stream structures that were not well-understood at the time of the '920 invention.

- Claim 2 depends from claim 1 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "wherein the encrypted copies of the content encryption key are entries in a table." *Id.* at 17:16-17.
- Claim 3 depends from claim 1 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "wherein the encrypted content is sent in response to a request from the playback device." *Id.* at 17:18-19.
- Claim 4 depends from claim 1 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "wherein the active user encryption key is encrypted by a base encryption key, where the base encryption key is inherent to the class of devices to which the playback device belongs." *Id.* at 17:20-23.
- Claim 5 depends from claim 1 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "wherein digital rights specified with respect to the content by a content provider are encrypted using at least one base encryption key, where the base encryption key is inherent to the class of devices to which the playback device belongs." *Id.* at 17:24-28.

- Claim 6 depends from claim 1 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "identifying, using the playback device, an active base encryption key for the particular class of device that is attempting to access the content; and accessing, using the playback device, information concerning the type of playback parameters supported by a playback certification included with the content." *Id.* at 17:29-35.
- Claim 7 depends from claim 6 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "where the playback certification includes multiple base encryption keys." *Id.* at 17:36-37.
- Claim 8 depends from claim 7 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "where each of the base keys is identifiable using a unique identifier." *Id.* at 17:38-39.
- Claim 9 depends from claim 7 and further describes how the new DRM architecture enhances content security and allows secure streaming, reciting "where each of the base encryption keys is used to encrypt the same information." *Id.* at 40-41.

165. Claim 10 of the '920 patent recites how an improved playback device is configured to decrypt, decode, and play back content encrypted using the new digital video encryption format of the invention:

> 10. A playback device configured to playback encrypted content,
>
> 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 encrypted using one or more user encryption keys including the active user encryption key, the playback device comprising:

memory comprising a playback application; and

a processor;

wherein the processor is configured by the playback application to:

obtain encrypted content, 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;

obtain 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;

decrypt one of the one or more copies of the content encryption key using the active user encryption key; and

play back frames of the encrypted content, where playing back frames of the encrypted content further comprises:

identifying any portions of a frame that are encrypted;

identifying the frame encryption key used to encrypt the identified portions of the frame;

decrypting the identified frame encryption key using the decrypted content encryption key;

decrypting the encrypted portions of the frame using the decrypted identified frame encryption key; and

decoding the unencrypted frame of video.

Id. at 17:42-18:23. The limitations of claim 10 enable the benefits of the '920 invention of enhanced content security and enhanced access control that are improvements over prior video encryption formats. Claim 10, therefore, recites a novel solution for improving the security of digital content and user access control in a manner that was not well-understood, routine, or conventional at the time of the '920 invention.

166. Claims 11-18 of the '920 patent depend from claim 10, and each of claims 11-18 further describes how an improved playback device is configured to decrypt, decode, and play back content encrypted using the new digital video encryption format of the invention, enhancing content security and access control and allowing secure streaming. The ordered combination of elements in each of claims 11-18, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer processes and video stream structures that were not well-understood at the time of the '920 invention.

- Claim 11 depends from claim 10 and further describes the structure of the new DRM architecture that enhances content security and allows secure streaming, reciting "wherein the encrypted copies of the content encryption key are entries in a table." *Id.* at 18:25-27.
- Claim 12 depends from claim 10 and further describes the structure of the new DRM architecture that enhances content security and allows secure streaming, reciting "wherein the encrypted content is sent in response to a request from the playback device." *Id.* at 18:28-30.
- Claim 13 depends from claim 10 and further describes the structure of the new DRM architecture that enhances content security and allows

secure streaming, reciting "wherein the active user encryption key is encrypted by a base encryption key, where the base encryption key is inherent to the class of devices to which the playback device belongs." *Id.* at 18:31-34.

- Claim 14 depends from claim 10 and further describes the structure of the new DRM architecture that enhances content security and allows secure streaming, reciting "wherein digital rights specified with respect to the content by a content provider are encrypted using at least one base encryption key, where the base encryption key is inherent to the class of devices to which the playback device belongs." *Id.* at 18:35-39.
- Claim 15 depends from claim 10 and further describes the structure of the new DRM architecture that enhances content security and allows secure streaming, reciting "identifying, using the playback device, an active base encryption key for the particular class of device that is attempting to access the content; and accessing, using the playback device, information concerning the type of playback parameters supported by a playback certification included with the content." *Id.* at 18:40-47.
- Claim 16 depends from claim 15 and further describes the structure of the new DRM architecture that enhances content security and allows secure streaming, reciting "where the playback certification includes multiple base encryption keys." *Id.* at 18:48-49.
- Claim 17 depends from claim 16 and further describes the structure of the new DRM architecture that enhances content security and allows secure streaming, reciting "where each of the base keys is identifiable using a unique identifier." *Id.* at 18:50-51.

- Claim 18 depends from claim 16 and further describes the structure of the DRM architecture that enhances content security and allows secure streaming, reciting "where each of the base encryption keys is used to encrypt the same information." *Id.* at 18:52-53.

VI. The '720 Patent

167. 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.

Summary of the '720 Invention

168. The '720 claims are directed to improvements to the functionality of computer systems providing adaptive bitrate streaming (ABS) of digital video from server computers to client computers (playback devices), including, for example, personal computers, CE players, smartphones, DVD players, Blu-ray players, televisions, video game consoles, and tablets. '720 patent, 9:1-8 (describing playback devices). The '720 claims describe how to automatically generate and provide new, improved top level index files to playback devices, to use for performing ABS. The improved top level index file that the claims provide are tailored to the capabilities of each playback device requesting a video to play back, which improves the playback device's ability to efficiently request the correct, compatible streams from the playback server system.

169. The inventions recited in the '720 claims enable Netflix to offer ABS services that perform smoothly and without stalls when switching among video streams of different resolutions during playback on a playback device. Specifically, the '720 claims are 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 video streams, improving ABS.

Technical Problems Addressed by the '720 Invention

170. The '720 patent addresses a technical problem related to ABS. In ABS, a playback device detects streaming conditions (such as changes in network bandwidth) in real time and adjusts the resolution of the streamed video accordingly so that the viewer does not experience interruptions due to changes in conditions. *Id.* at 1:26-45. Specifically, in ABS, the playback device uses a digital "top level index file" to request different video streams from the server, encoded at different bitrates. *Id.* at 6:39-43. A "top level index file" is a type of computer data structure used specifically for video streaming. *Id.* at 6:39-45.

171. "In adaptive bitrate streaming systems, the top level index file typically references the alternative streams that the playback device can switch between." *Id.* "To perform adaptive bitrate streaming, the playback devices . . . select content from different alternative streams described in the top level index file." *Id.* at 7:29-31. "The playback device can select one or more streams for conventional streaming or can switch between alternative streams to perform adaptive bitrate streaming." *Id.* at 7:39-42; *see also id.* at 9:20-48 (describing use of top level index file during ABS operation), 10:18-22 (same).

172. Many different types of consumer devices can play back video delivered over the internet, including computers, mobile phones, Blu-ray players, and televisions. *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. ABS increases the complexity of digital video delivery by, among other reasons, enabling the playback device to switch among different quality streams based on changes in network conditions. *See, e.g., id.* at 1:30-45, 12:20-31. Each playback device needs a separate top level index file containing information regarding each piece of video content that the device will request during ABS. *See, e.g., id.* at 12:20-40. That is, each playback device has unique computing characteristics and capabilities and, therefore, needs a device-specific

index file that enables it to request the video streams suitable for playback on that device. *Id.* at 12:20-13:24 (describing filtering assets to generate top level index files tailored to device capabilities, including aspect ratio, resolution of the playback device's display, and the maximum data rate of the playback device's network connection).

173. Before the '720 invention, servers could not provide device-specific index files for a variety of devices without compiling and maintaining a library of separate index files for each device supported by the streaming system. This would have imposed a burden on server-side computing resources, including processing power and memory, that scaled with the number of supported devices. The computing resources needed to compile and maintain a separate index file for each combination of content and device would have made such a system infeasible. Further, using the same index file for devices with different characteristics produces poor playback, including video stalls, on many devices. Accordingly, a need existed for an efficient system to automatically generate top level index files for different playback devices for ABS based on device characteristics, to improve the performance of the computing devices playing back video.

Technical Solutions and Benefits Provided by the '720 Invention

174. The '720 patent claims specific ways to solve these technical problems with methods and systems for automatically generating an improved top level index file for a particular playback device based on that playback device's unique computing characteristics for use in ABS. The '720 claims are directed to a new, improved method for providing a top level index file to a playback device by generating tailored files in response to a request for content (claim 1 and dependents) and a new, improved playback server system specifically configured to automatically generate an improved, tailored top level index file in response to a request from a playback device (claim 13 and dependents).

175. The new methods and systems for automatically generating an improved top level index file for a particular playback device of the '720 patent include receiving specific information from a playback device requesting a video, retrieving and filtering streams associated with the requested video based on the playback device's capabilities, generating the improved top level index file describing video streams compatible with the playback device, and sending that improved index file to the playback device for use in ABS. *See, e.g.*, *id.* at 2:24-28 (describing filtering the streams associated with requested content using criteria specific to the playback device, to generate a top level index file), 6:39-43 (stating that the "top level index is a file that describes the location and content of container files containing streams of media . . . that can be utilized by the playback device to stream and playback content"), 6:50-55 (describing filtering the streams based on playback device capabilities, information associated with the user account, or other rules defined by the content owner). This method for generating an improved, tailored top level index file was not well-known, routine, or conventional at the time of the '720 invention. Prior playback server systems did not generate tailored top level index files for use by a particular playback device in ABS, in response to the playback device request.

176. The methods and systems for automatically generating an improved, tailored top level index file of the '720 patent provide technical benefits that improve the functionality and capabilities of computer systems performing ABS. *See, e.g.*, *id.* at Abstract, 2:17-28, 9:63-10:17 (describing automatically generating the top level index file in response to particularized device characteristics provides the playback server), 12:20-13:24 (describing filtering assets for specific playback devices). The server does not need to store a static top level index file for every unique playback device, and each playback device receives an index to video streams tailored to that specific device's computing characteristics. That is, the '720 patent's methods for generation of top level index files results in more efficient

ABS specific to the technical capabilities of a particular playback device, improving the performance of both the ABS server computer and the playback device. The '720 patent's generation and delivery of tailored top level index files based on device characteristics improves the performance of the playback devices using those files. *Id.* at 12:20-13:24.

Prosecution History of the '720 Invention

177. The '720 invention improves upon the "many . . . ways one could practice the art of distributing content to playback devices." '720 File History,[42] Amendment and Remarks, Sept. 17, 2015, at 12. The '720 claims "encompass a transformation, in that a playback server system receives a request from a playback device that identifies a piece of content, retrieves a list of assets and filters the lists of assets based on a device capability of the playback device, generates a top level index file describing the filtered list of assets and sends the top level index file to the playback device, which allows the playback device to determine which assets to request for playback on the device, and these assets would be compatible with the capabilities of the playback device." *Id.*

178. The prior art identified during prosecution of the '720 patent did not disclose a list of assets "dependent on the media playback capabilities of the intended destination device," as required by the '720 claims. *Id.* at 14.

179. The prior art identified during prosecution of the '720 patent also taught away from "***sending the top level index file to the playback device*** using the playback server." *Id.* at 15 (bold and italics in original).

180. The claims of the '720 patent issued at least because they recite "assessing by a media server the capabilities of a playback device and providing an index file with a list of assets based on these capabilities" and "providing a playback device with a list of assets as opposed to a media repository streaming a

[42] Cited excerpts of the '720 file history attached as Exhibit 13.

multi-part media item file in a manner 'dependent on the media playback capabilities of the intended destination device.'" *Id.* at 14-15. The '720 claims are specifically directed to "a playback server . . . that automatically generates a top level index file with a list of assets for a playback device based on the capabilities of the device." *Id.* at 15.

181. During prosecution, the patent examiner rejected the pending claims of the '720 patent under 35 U.S.C. § 101. '720 File History, Office Action, June 3, 2015, at 13-14. The applicant overcame the rejection through further examiner consideration and amendment of the rejected independent claims to recite "wherein the top level index file is used by the playback device to determine which assets to request for playback on the device." '720 File History, Amendment and Remarks, Sept. 17, 2015, at 11-12; '720 File History, Examiner Initiated Interview Summary, Oct. 19, 2015; '720 File History, Notice of Allowance, Oct. 19, 2015, at 2, 4 ("The examiner's amendment above [at 2] to the claims have been considered, and have been found to be persuasive, therefore the [§ 101] rejections are withdrawn."). The patent examiner did not raise a rejection under § 101 after the amendment. *See* '720 File History, Notice of Allowance, Oct. 19, 2015, at 4. The '720 patent issued on February 23, 2016, after the U.S. Supreme Court's decision in *Alice Corp. Pty Ltd. v. CLS Bank Int'l*, 573 U.S. 208 (2014).

Claims Reciting the Technical Solutions of the '720 Invention

182. Claim 1 of the '720 patent recites a specific way to automatically generate a new top level index file for ABS tailored to a specific playback device. '720 patent, 20:15-35. The steps of '720 claim 1 recite how to improve the performance of an ABS computing system by generating a specific top level index file in response to particularized device characteristics and providing that specific index file to the device for use in selecting streams during ABS playback. Claim 1 of the '720 patent recites how to perform an improved method for generating a top level index file according to the invention:

> 1. A method of generating a top level index file, comprising:
>
> receiving 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;
>
> retrieving, 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;
>
> filtering 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;
>
> generating a top level index file describing each asset in the filtered list of assets using the playback server system; and
>
> sending 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.

Id. The language of claim 1 indicates that the "retrieving," "filtering," and "generating a top level index file" steps occur *in response to* the step of "receiving a request from a playback device" because those later steps refer back to the content of the request (e.g., "retrieving . . . a list of assets associated with *the* identified piece of content" identified in the request; "retrieving . . . at least one device capability based upon *the* product identifier" included in the request; "filtering *the* list of assets using *the* at least one device capability"; and "generating a top level

index file describing each asset in *the* filtered list of assets") (all emphases added). The '720 specification consistently describes the series of logically ordered steps recited in '720 claim 1 as "automatically generating top level index files," including visual portrayals of "automatically generating" in FIG. 4 and FIG. 9. *See, e.g.*, *id.* at Title ("Systems and Methods for Automatically Generating Top Level Index Files"), Abstract, 1:15-19 (Field of the Invention), 2:15-28 (Summary of the Invention), 6:12-18 (describing FIG. 4 and FIG. 5), 6:28-32 (describing FIG. 9), 7:15-42, 7:55-8:3 (describing "automatically generat[ing] top level index files" using "product IDs"), 10:61-12:19 (describing "Automatic Generation of Top Level Indexes" and depiction in FIG. 4), 12:20-13:34 (describing "Filtering Assets for Inclusion in Top Level Index Files," depiction in FIG. 5, and creation of "a top level index file . . . in real time in response to a request from a specific playback device"), 19:32-62 (describing FIG. 9 in the context of processes "for automatically generating a top level index file in response to a request to access content from a playback device").

183. These steps of claim 1 enable the benefits of reducing computing resources consumed at the playback server while improving performance of video playback using device-specific index files. *Id.* at 2:24-28, 6:39-43, 6:50-55, 20:15-35. 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 ABS specific to the technical capabilities of a particular playback device. Claim 1 recites a novel method that provides a solution for improving the performance of ABS using new top level index files in a manner that was not well-understood, routine, or conventional at the time of the '720 patent.

184. Claims 2-12 of the '720 patent depend from claim 1, and each of claims 2-12 further describes how to perform the invention's improved method for producing a new top level index file that improves the performance of ABS on the

playback computing device. The ordered combination of elements in each of claims 2-12, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer processes and top level index file structures that were not well-understood at the time of the '720 invention.

- Claim 2 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "filtering the list of assets based upon at least one of a geographic location of the playback device, a language associated with the playback device, one or more user preferences, and one or more requirements of a content owner." *Id.* at 20:36-40.
- Claim 3 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the at least one device capability is at least one of a: display aspect ratio, anticipated maximum network connection data rate, device outputs, supported formats, device buffer size, device resolution, device region, and device language." *Id.* at 20:41-45.
- Claim 4 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the playback server system maintains a database of assets associated with specific pieces of content." *Id.* at 20:46-48.
- Claim 5 depends from claim 1 and further describes the structure of the improved top level index file automatically generated using the improved method, reciting "the top level index file describes at least a bitrate of each asset in the filtered list of assets and identifies locations of the assets in the filtered list of assets." *Id.* at 20:49-52.
- Claim 6 depends from claim 5 and further describes the structure of the improved top level index file automatically generated using the

improved method, reciting "the top level index file is a SMIL file." *Id.* at 20:53-54.

- Claim 7 depends from claim 6 and further describes how the improved method automatically generates the improved top level index file, reciting "generating an XML string including a SWITCH element to describe alternative streams for use in adaptive bitrate streaming." *Id.* at 20:55-57.
- Claim 8 depends from claim 6 and further describes how the improved method automatically generates the improved top level index file, reciting "generating an XML string including an EXCL element to describe alternative streams for use in conventional streaming." *Id.* at 20:58-60.
- Claim 9 depends from claim 6 and further describes how the improved method automatically generates the improved top level index file, reciting "generating an XML string including a URI for each asset, wherein the URI references a container file and the XML string for each assets includes an element that defines the size of a header section of the container file." *Id.* at 20:61-65.
- Claim 10 depends from claim 6 and further describes how the improved method automatically generates the improved top level index file, reciting "the XML string includes an element that identifies the encoding of the asset." *Id.* at 20:66-67.
- Claim 11 depends from claim 6 and further describes how the improved method automatically generates the improved top level index file, reciting "the XML string of a video asset includes at least one element selected from the group consisting of: an element that describes the maximum bitrate of the video; an element that describes

the width and height of the video; and an element that describes the video buffer verifier size of the video." *Id.* at 21:1-9.

- Claim 12 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "each asset is a different alternative stream associated with the piece of content and each alternative stream encodes the piece of content at a different maximum bitrate." *Id.* at 21:10-13.

185. Claim 13 of the '720 patent recites a "playback server system" implementing a specific way to automatically generate an improved top level index file for ABS tailored to a specific playback device. *Id.* at 21:14-22:6. The elements of '720 claim 13 recite how to improve the performance of an ABS computing system by generating a specific top level index file in response to particular device characteristics and providing that specific index file to the device for use in selecting streams during ABS playback. Claim 13 of the '720 patent recites how an improved playback server system is configured to generate a top level index file according to the invention:

> 13. A playback server system, comprising:
> a database that stores descriptions of assets associated with specific pieces of content;
> a database that stores a plurality of product identifiers and associated device capabilities;
> a processor configured using a playback management application;
> wherein the playback management application configures the processor to:

> receive a request from a playback device, where the request (i) identifies a piece of content and (ii) includes a product identifier;
> retrieve (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;
> filter the list of assets using the at least one device capability;
> generate a top level index file describing each asset in the filtered list of assets; and
> send the top level index file to the playback device, wherein the top level index file is used by the playback device to determine which assets to request for playback on the device.

Id. The language of claim 13 indicates that the operations to "retrieve," "filter," and "generate a top level index file" occur *in response to* the operation to "receive a request from a playback device" because those later operations refer back to the content of the request (e.g., "retrieve . . . a list of assets associated with *the* identified piece of content" identified in the request; "retrieve . . . at least one device capability based upon *the* product identifier" included in the request; "filter *the* list of assets using *the* at least one device capability"; and "generate a top level index file describing each asset in *the* filtered list of assets" (all emphases added)). The '720 specification consistently describes the series of operations recited in '720 claim 13 as "automatically generating top level index files," including visual portrayals of "automatically generating" in FIG. 4 and FIG. 9. *See, e.g., id.* at Title ("Systems and Methods for Automatically Generating top Level Index Files"), Abstract, 1:15-19 (Field of the Invention), 2:15-28 (Summary of the Invention),

6:12-18 (describing FIG. 4 and FIG. 5), 6:28-32 (describing FIG. 9), 7:15-42, 7:55-8:3 (describing "automatically generat[ing] top level index files" using "product IDs"), 10:61-12:19 (describing "Automatic Generation of Top Level Indexes" and depiction in FIG. 4), 12:20-13:34 (describing "Filtering Assets for Inclusion in Top Level Index Files," depiction in FIG. 5, and creation of "a top level index file . . . in real time in response to a request from a specific playback device"), 19:32-62 (describing FIG. 9 in the context of processes "for automatically generating a top level index file in response to a request to access content from a playback device").

186. The steps of claim 13 enable the benefits of reducing computing resources consumed at the playback server while improving performance of video playback using device-specific index files. *Id.* at 2:24-28, 6:39-43, 6:50-55, 21:14-22:6. 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 ABS specific to the technical capabilities of a particular playback device. Claim 13 recites a novel system that provides a solution for improving the performance of ABS using new top level index files in a manner that was not well-understood, routine, or conventional at the time of the '720 patent.

187. Claims 14-18 of the '720 patent depend from claim 13, and each of claims 14-18 further describes how the invention's improved playback server system is configured to generate a new top level index file that improves the performance of ABS on the playback computing device. The ordered combination of elements in each of claims 14-18, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer systems and top level index file structures that were not well-understood at the time of the '720 invention.

- Claim 14 depends from claim 13 and further describes how the improved playback server system is configured to automatically

generate the improved top level index file, reciting "the playback management application further configures the processor to filter the list of assets based upon at least one of a geographic location of the playback device, a language associated with the playback device, one or more user preferences, and one or more requirements of a content owner." *Id.* at 22:7-12.

- Claim 15 depends from claim 13 and further describes how the improved playback server system is configured to automatically generate the improved top level index file, reciting "the at least one device capability is least one of a: display aspect ratio, anticipated maximum network connection data rate, device outputs, supported formats, device buffer size, device resolution, device region, and device language." *Id.* at 22:13-17.
- Claim 16 depends from claim 13 and further describes the structure of the improved top level index file automatically generated using the improved playback server system, reciting "the top level index file describes at least a bitrate of each asset in the filtered list of assets and identifies locations of the assets in the filtered list of assets." *Id.* at 22:18-21.
- Claim 17 depends from claim 16 and further describes the structure of the improved top level index file automatically generated using the improved playback server system, reciting "the top level index file is a SMIL file that is an XML file that includes a list of URIs describing each of the different streams associated with the piece of content and container files that contain the streams." *Id.* at 22:22-26.
- Claim 18 depends from claim 13 and further describes how the improved playback server system is configured to automatically generate the improved top level index file, reciting "each asset is a

different alternative stream associated with the piece of content and each alternative stream encodes the piece of content at a different maximum bitrate." *Id.* at 22:27-30.

VII. The '515 Patent

188. 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.

Summary of the '515 Invention

189. The '515 claims are directed to improvements to the functionality of computer systems used to provide adaptive bitrate streaming (ABS) of digital video from server computers to client computers (playback devices), including, for example, personal computers, CE players, smartphones, DVD players, Blu-ray players, televisions, video game consoles, and tablets. '515 patent, 9:16-23 (describing playback devices). The '515 inventions claim a method for providing (or, on the playback device side, requesting, receiving, and using) improved top level index files used to perform ABS. Those new top level index files are tailored to the capabilities of each playback device requesting a video to play back, including device type and device software version, which improves the playback device's ability to efficiently request the correct, compatible streams from the playback server system.

190. The inventions recited in the '515 claims enable Netflix to offer ABS services that perform smoothly and without stalls when switching among video streams of different resolutions during playback on a user's device. Specifically, the '515 patent is directed to a playback server system that automatically generates—and a playback device configured to request, receive, and use—an improved top

level index file tailored to a particular playback device that the playback device uses to request a streaming file, improving ABS.

Technical Problems Addressed by the '515 Invention

191. The '515 patent shares a specification with the '720 patent and thus addresses the corresponding technical problem related to ABS for a diverse device ecosystem with many different kinds of devices and corresponding technical capabilities. *See, e.g.*, *id.* at 1:30-45, 8:2-9, 9:17-23, 11:65-12:16, 12:40-60.

192. In ABS, a playback device detects streaming conditions (such as changes in network bandwidth) in real time and adjusts the resolution of the streamed video accordingly so that the viewer does not experience interruptions due to changes in conditions. *Id.* at 1:30-45. Specifically, in ABS, the playback device uses a digital "top level index file" to request different video streams from the server. *Id.* at 6:50-54. A "top level index file" is a type of computer data structure used specifically for video streaming. *Id.* at 6:50-57. "A top level index file is a file that describes the location and content of container files containing streams of media (e.g. audio, video, metadata, and subtitles) that can be utilized by the playback device to stream and playback content." *Id.*

193. "In adaptive bitrate streaming systems, the top level index file typically references the alternative streams that the playback device can switch between." *Id.* "To perform adaptive bitrate streaming, the playback devices . . . select content from different alternative streams described in the top level index file." *Id.* at 7:43-56. "The playback device can select one or more streams for conventional streaming or can switch between alternative streams to perform adaptive bitrate streaming." *Id.*; *see also id.* at 9:35-63 (describing use of top level index file during ABS operation), 10:34-38 (same).

194. Many different types of consumer devices can play back video delivered over the internet, including computers, mobile phones, Blu-ray players, and televisions. *See, e.g.*, *id.* at 9:16-23. All of these devices have different

characteristics and technical capabilities for video playback. *See, e.g.*, *id.* at 8:2-9, 11:65-12:18, 12:39-51. ABS 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, 12:39-51. Each playback device needs a separate top level index file containing information regarding each piece of video content that a user will watch using ABS. *See*, *e.g.*, *id.* at 12:39-51. That is, each playback device has unique computing characteristics and, therefore, needs a device-specific index file that enables it to request the video streams suitable for playback on that device. *Id.* at 12:39-13:55 (describing filtering assets to generate top level index files tailored to device capabilities, including aspect ratio, resolution of the playback device's display, and the maximum data rate of the playback device's network connection).

195. Before the '515 invention, servers could not provide device-specific index files for a variety of devices without compiling and maintaining a library of separate index files for each device supported by the streaming system. This would have imposed a burden on server-side computing resources, including processing power and memory, that scaled with the number of supported devices. The computing resources needed to compile and maintain a separate index file for each combination of content and device would have made such a system infeasible. Further, using the same index file for devices with different characteristics produces poor playback, including video stalls, on many devices. Accordingly, a need existed for an efficient system to automatically generate and use index files for different playback devices for ABS based on device characteristics, to improve the performance of the computing devices playing back video.

Technical Solutions and Benefits Provided by the '515 Invention

196. The '515 patent claims a solution to these problems with methods and systems for automatically generating and using a top level index file for a particular playback device and particular video content for use in ABS based on the device's

specific attributes, including the type of device and software version. *See, e.g., id.* at 11:40-46, 20:43-67. The '515 claims are directed to improvements to the functionality of computer systems that perform ABS. The '515 claims are directed to a new, improved method for providing a top level index file to a playback device by generating tailored files in response to a request for content (claim 1 and dependents) and a new, improved playback device specifically configured to request and use an improved, tailored top level index file (claim 16 and dependents).

197. The new methods and systems for automatically generating or requesting and using an improved top level index file of the '515 patent include receiving specific information from a playback device requesting a video, filtering streams associated with the requested video based on the playback device's capabilities, generating the improved top level index file describing only those video streams compatible with the playback device, and sending that improved index file to the playback device for use in ABS. *See, e.g., id.* at 2:28-33 (describing filtering the streams associated with requested content using criteria specific to the playback device, to generate a top level index file), 6:50-54 (The "top level index is a file that describes the location and content of container files containing streams of media . . . that can be utilized by the playback device to stream and playback content."), 6:62-67 (describing filtering the streams based on playback device capabilities, information associated with the user account, or other rules defined by the content owner), 20:43-67 (claim 1), 22:4-27 (claim 16); *see also id.* at Abstract, 1:18-22 (Field of the Invention), 2:19-33 (Summary of the Invention), 6:22-28 (describing FIG. 4 and FIG. 5), 6:38-43 (describing FIG. 9), 7:28-56, 8:2-17 (describing "automatically generat[ing] top level index files" using "product IDs"), 11:12-12:38 (describing "Automatic Generation of Top Level Indexes" and depiction in FIG. 4), 12:39-13:55 (describing "Filtering Assets for Inclusion in Top Level Index Files," depiction in FIG. 5, and creation of "a top level index file . . . in

real time in response to a request from a specific playback device"), 19:60-20:23 (describing FIG. 9 in the context of processes "for automatically generating a top level index file in response to a request to access content from a playback device"). These methods for generating an improved top level index file and using that improved file were not well-known, routine, or conventional at the time of the '515 invention. Prior playback server systems did not generate tailored top level index files for use by a particular playback device in ABS. And playback devices did not receive and use those tailored files for ABS.

198. The methods and systems for automatically generating or using an improved, tailored top level index file of the '515 patent provide technical benefits that improve the functionality and capabilities of computer systems—including server-side and device-side computers—performing ABS. *See, e.g.*, *id.* at Abstract, 2:21-33, 10:11-34 (describing automatically generating the top level index file in response to particularized device characteristics provides the playback server), 12:39-13:55 (describing filtering assets for specific playback devices). The server does not need to store a static top level index file for every unique playback device, and each playback device receives and uses an index containing video streams tailored to that specific device's computing characteristics. The '515 patent's method for generation of top level index files results in more efficient ABS specific to the technical capabilities of a particular playback device, improving the performance of both the ABS server computer and the playback device.

Prosecution History of the '515 Invention

199. The '515 invention improves upon the "many . . . ways one could practice the art of distributing content to playback devices." '515 File History,[43] Amendment and Remarks, Feb. 17, 2017, at 11. The '515 claims "encompass a transformation, in that a playback server receives a request from a playback device

[43] Cited excerpts of the '515 file history attached as Exhibit 14.

that identifies a piece of content, retrieves a list of assets and filters the lists of assets based on a device capability of the playback device, generates a top level index file describing the filtered list of assets and sends the top level index file to the playback device, which allows the playback device to determine which assets to request for playback on the device, and these assets would be compatible with the capabilities of the playback device." *Id.*

200. The prior art identified during prosecution of the '515 patent did not disclose "a product identifier that identifies a device configuration" or that "the top level index file identifies locations and bitrates of a plurality of alternative streams capable of being used to perform adaptive streaming of the content," as claim 1 of the '515 patent requires (prosecuted as claim 21). *Id.* at 14.

201. The prior art identified during prosecution of the '515 patent also taught away from "sending the top level index file from the playback server to the playback device." *Id.* at 15, 17.

202. The claims of the '515 patent issued at least because they recite "providing a playback device with a list of assets as opposed to a media repository streaming a multi-part media item file in a manner 'dependent on the media playback capabilities of the intended destination device.'" *Id.* at 17.

203. During prosecution, the patent examiner initially rejected the pending claims of the '515 patent under 35 U.S.C. § 101. '515 File History, Office Action, May 18, 2016, at 17-19. The applicant addressed the rejection by offering a new claim set. '515 File History, Amendment and Remarks, Sept. 19, 2016, at 2-5. The patent examiner then rejected those pending claims under § 101. '515 File History, Final Rejection, Nov. 17, 2016, at 9-11. The applicant then overcame that rejection through further examiner consideration and applicant amendment. '515 File History, Amendment and Remarks, Feb. 17, 2017, at 9-11. Specifically, the amended claims were directed toward "receiving, processing, generating, and sending data using a playback server or a playback device, that are not merely

generic computers performing generic computer functions that are well understood, routine, and conventional activities previously known in the industry." *Id.* at 10; *see also id.* at 2-6 (amending claims to explicitly provide that the claimed method is performed "using the playback server" or "using the playback device"). The examiner withdrew his § 101 rejections following these amendments. '515 File History, Office Action, June 28, 2017, at 5 ("[A]pplicant's amendments have been considered, applicant's claims now contain significantly more, and therefore, the [§ 101] rejections are withdrawn."). The patent examiner did not raise a rejection under § 101 after the amendment. *See* '515 File History, Notice of Allowance, Feb. 9, 2018. The '515 patent issued on June 12, 2018, after the U.S. Supreme Court's decision in *Alice Corp. Pty Ltd. v. CLS Bank Int'l*, 573 U.S. 208 (2014).

Claims Reciting the Technical Solutions of the '515 Invention

204. Claim 1 of the '515 patent recites a specific way to automatically generate a top level index file for ABS tailored to a specific playback device. '515 patent, 20:43-67. The steps of '515 claim 1 recite how to improve the performance of an ABS computing system by automatically generating a specific top level index file in response to particularized device information and providing that specific index file to the device for use in selecting streams during ABS playback. Claim 1 of the '515 patent recites how to perform an improved method for authorizing playback of content, including automatically generating a top level index file according to the invention:

> 1. A method for authorizing playback of content, comprising:
>
> receiving a request for content from a playback device at a playback server, where the request includes a product identifier that identifies a device configuration;
>
> identifying, 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;
>
> retrieving, 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;
>
> filtering, using the playback server, the list of assets based on the plurality of device capabilities;
>
> generating, 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 streaming of the content; and
>
> sending the top level index file from the playback server to the playback device.

Id. The language of claim 1 indicates that the "identifying," "retrieving," "filtering," and "generating . . . a top level index file" steps occur *in response to* the step of "receiving a request for content from a playback device" because those later steps refer back to the content of the request (e.g., "identifying . . . based on *the* product identifier, a plurality of device capabilities," where the request contained the product identifier; "retrieving . . . a list of assets associated with *the* identified piece of content" in the request; "filtering . . . *the* list of assets" retrieved using the identified piece of content "based on *the* plurality of device capabilities"; and "generating . . . a top level index file describing each asset in *the* filtered list of assets" (all emphases added)). The '515 specification consistently describes the

series of steps recited in '515 claim 1 as "automatically generating top level index files," including visual portrayals of "automatically generating" in FIG. 4 and FIG. 9. *See, e.g.*, *id.* at Title ("Systems and Methods for Automatically Generating Top Level Index Files"), Abstract, 1:20-22 (Field of the Invention), 2:21-33 (Summary of the Invention), 6:22-28 (describing FIG. 4 and FIG. 5), 6:38-43 (describing FIG. 9), 7:28-56, 8:2-17 (describing "automatically generat[ing] top level index files" using "product IDs"), 11:12-12:39 (describing "Automatic Generation of Top Level Indexes" and depiction in FIG. 4), 12:40-13:55 (describing "Filtering Assets for Inclusion in Top Level Index Files," depiction in FIG. 5, and creation of "a top level index file . . . in real time in response to a request from a specific playback device"), 19:60-20:23 (describing FIG. 9 in the context of processes "for automatically generating a top level index file in response to a request to access content from a playback device").

205. The steps of claim 1 enable the benefits of reducing computing resources consumed at the playback server while improving performance of video playback using device-specific index files. *Id.* at 2:28-33, 6:50-54, 6:62-67, 20:43-67. The playback server system automatically 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 ABS specific to the technical capabilities of a particular playback device. Claim 1 recites a novel method that provides a solution for improving the performance of ABS using specific top level index files in a manner that was not well-understood, routine, or conventional at the time of the '515 invention.

206. Claims 2-15 and 20 of the '515 patent depend from claim 1, and each of claims 2-15 and 20 further describes how to perform the invention's improved method for automatically generating a new top level index file that improves the performance of ABS on the playback computing device. The ordered combination of elements in each of claims 2-15 and 20, in conjunction with the elements of the

claims from which they depend, therefore recite unconventional new and improved computer processes and top level index file structures that were not well-understood at the time of the '515 invention.

- Claim 2 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the request for content from the playback device further comprises information regarding a web browser on the playback device being used to request the content." *Id.* at 21:1-4.
- Claim 3 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the request for content from the playback device further comprises information regarding an operating system of the playback device." *Id.* at 21:5-7.
- Claim 4 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the request is made via a Hypertext Transfer Protocol (HTTP) on a network." *Id.* at 21:8-9.
- Claim 5 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the request including information includes (i) information describing a type of the playback device and (ii) information regarding the network bandwidth." *Id.* at 21:10-13.
- Claim 6 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "filtering the list of assets based upon a version of a web browser on the playback device requesting content." *Id.* at 21:14-16.
- Claim 7 depends from claim 1 and further describes the structure of the improved top level index file generated using the improved

method, reciting "the top level index file is a SMIL file." *Id.* at 21:17-18.

- Claim 8 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the request from the playback device also includes information describing the capabilities of the playback device; and the playback server filters the list of assets based upon the capabilities of the playback device." *Id.* at 21:19-24.
- Claim 9 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "retrieving at least one device capability based upon the product identifier using the playback server." *Id.* at 21:25-27.
- Claim 10 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the device capability is at least one device capability selected from the group consisting of: display aspect ratio, anticipated maximum network connection data rate, device outputs, supported formats, device buffer size, device resolution, device region, and device language." *Id.* at 21:28-33.
- Claim 11 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the request from the playback device also includes information indicative of a geographic location of the playback device; and the method further comprising filtering the list of assets based on whether an asset is permitted to be played back in the geographic location indicated by the request." *Id.* at 21:34-41.
- Claim 12 depends from claim 1 and further describes how the improved method automatically generates the improved top level

index file, reciting "filtering the list of assets based upon at least one language." *Id.* at 21:42-43.

- Claim 13 depends from claim 1 and further describes how the playback server system is able to perform the improved method of automatically generating the improved top level index file, reciting "the playback server maintains a database of assets associated with specific pieces of content." *Id.* at 21:44-46.
- Claim 14 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file and its structure, reciting "generating the top level index file describing each asset in the filtered list of assets comprises generating an XML string including a URI for each asset." *Id.* at 21:47-50.
- Claim 15 depends from claim 14 and further describes how the improved method automatically generates the improved top level index file and its structure, reciting "generating an XML string including a SWITCH element to describe alternative streams for use in adaptive bitrate streaming." *Id.* at 22:1-3.
- Claim 20 depends from claim 1 and further describes how the improved method automatically generates the improved top level index file, reciting "the playback device is capable of playing back streams with an identified resolution, and the list of assets is filtered to exclude streams with the identified resolution from the filtered list of assets." *Id.* at 22:41-44.

207. Claim 16 of the '515 patent recites a "playback device" configured to request, receive, and use the new, device-tailored top level index files for ABS. *Id.* at 22:4-27. The elements of '515 claim 16 recite how to improve the performance of an ABS computing system by having the playback device request a top level index file for a particular piece of content and specific device traits (e.g., "a

software version indicating a version number for an adaptive streaming software component implemented on the device") and receive a tailored top level index file in response to that request. Claim 16 of the '515 patent recites how an improved playback device is configured to request and receive a new, improved top level index file according to the invention and to play back video using that top level index file:

> 16. A playback device, comprising:
> memory containing information used to identify capabilities of the playback device; and
> a processor configured by a client application;
> wherein the client 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;
> 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;
> select, using the playback device, an initial stream from the plurality of different alternative streams;
> retrieve, using the playback device, at least a portion of the initial stream from the locations identified in the top level index file; and

play back, using the playback device, the portion of the initial stream.

Id. The invention of claim 16 reduces computing resources consumed at the playback device. The playback device is configured to provide particular information to the playback server system and, in turn, receives a device-specific top level index file for use in ABS. The top level index file identifies alternative streams "accessible to the playback device." *Id.* More specifically, the top level index file is tailored to the device-provided "software version indicating a version number for an adaptive streaming software component implemented on the device," and the file provides the "locations and bitrates" of those streams. *Id.* The playback device need not parse through a larger top level index file that contains streams that the device cannot or does not want to play back. And that file specifically provides the locations and bitrates of relevant streams that the playback device uses to retrieve and play back portions of those streams. Thus, claim 16 recites a novel top level index file that provides a solution for improving the performance of ABS in a manner that was not well-understood, routine, or conventional at the time of the '515 patent.

208. Claims 17-19 and 21 of the '515 patent depend from claim 16, and each of claims 17-19 and 21 further describes how the invention's playback device is configured to request, receive, and use an improved top level index file that enhances the performance of ABS on the playback device. The ordered combination of elements in each of claims 17-19 and 21, in conjunction with the elements of the claims from which they depend, therefore recite unconventional, new-and-improved computer systems and top level index file structures that were not well-understood at the time of the '515 invention.

- Claim 17 depends from claim 16 and further describes the structure of the improved top level index file provided to the improved playback device, reciting "the top level index file describes each stream using an

XML string including a URI identifying the location of the stream." *Id.* at 22:28-30.

- Claim 18 depends from claim 17 and further describes the structure of the improved top level index file provided to the improved playback device, reciting "the URI references a container file and the XML string for each stream includes an element that defines the size of a header section of the container file." *Id.* at 22:31-34.
- Claim 19 depends from claim 16 and further describes how the improved playback device receives the improved top level index file, reciting "at least one of the capabilities of the playback device is selected from the group consisting of: display aspect ratio, anticipated maximum network connection data rate, device outputs, supported formats, device buffer size, display resolution, device region, and device language." *Id.* at 22:35-40.
- Claim 21 depends from claim 16 and further describes the structure of the improved top level index file that the improved playback device receives and uses, reciting "the playback device is capable of playing back streams with an identified resolution, and the top level index file excludes descriptions of streams with the identified resolution." *Id.* at 22:45-48.

VIII. The '486 Patent

209. 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.

Summary of the '486 Invention

210. The '486 claims are directed to improvements to the structure of encrypted video files in playback devices and methods for decrypting and decoding those files to improve the security of digital video content during playback. '486 patent, 1:26-59. The '486 invention provides a new type of encryption for digital video files, and improved methods for decrypting and playing back those encrypted files, that improve the security of the digital video data by reducing the likelihood that an unauthorized user can access the data. Specifically, the '486 patent is directed to a content security architecture that deciphers frame keys within a secure video decoder, efficiently enhancing content security. "[B]y 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." *Id.* at 5:37-40. The inventions recited in the '486 patent enable Netflix to improve the security of its video streaming system, allowing it to obtain content from content providers and to trust in the security of its own, home-grown content.

Technical Problems Addressed by the '486 Invention

211. In digital multimedia distribution systems, "the multimedia file is authorized and decrypted in a demultiplexer and then transmitted downstream unencrypted to the decoder via an inter-communication data channel. This however can present a security problem due to the high value of the unencrypted but still encoded bitstream that can be captured during transmission. This bitstream is considered high-value since the encoded data can be easily multiplexed[44] back into a container for unprotected and unauthorized views and/or distribution with no loss in the quality of the data." *Id.* at 6:55-65.

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

[44] Multiplexed typically refers to repackaging into a multimedia file.

content is transmitted over connections that are not secure and can be intercepted. *See, e.g.*, *id.* at 1:53-59 (explaining that "when communication or the transporting of information becomes unsecured or untrustworthy, such gaps need to be accounted for and filled"). 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. *Id.* at 1:51-53, 1:57-59.

Technical Solutions and Benefits Provided by the '486 Invention

213. 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 invention packages decryption information with digital video in a "container file" and allows processing of that file such that decryption can occur on the video decoder. *Id.* at 5:66-6:32, FIG. 1, FIG. 2. The '486 claims are therefore directed to improvements to the functionality of computer systems that perform digital video decryption, decoding, and playback. The '486 claims are directed to a playback device with a new structure of container file containing encrypted digital video; how a playback device is configured to decrypt, decode, and play back the new file structure (claim 1 and dependents); and the method of decrypting, decoding, and playing back that new file structure (claim 15 and dependents). Prior video container file formats did not contain this specific structure of partially encrypted frames and cryptographic information necessary for decryption and decoding. This new file structure, and the playback devices and methods used to decrypt and play back video structured in this new way, therefore were not well-known, routine, or conventional at the time of the '486 invention.

214. The new structure of a container file containing encrypted digital video of the '486 invention and the playback devices and methods used to decrypt and play back video structured in this new way provide technical benefits that improve

the functionality and capabilities of computer systems performing these operations. By providing partially encrypted video frames, coupled with specific cryptographic information describing the encrypted portion of each partially encrypted frame, and requiring deciphering of frame keys using the cryptographic material, the new container file format improves the security of the video data and reduces the processing resources required to decrypt and play back the video. The '486 inventions "do not secure the transmission but rather secure the data being transmitted via the unsecured connection." *See, e.g., id.* at 5:22-40. The inventions accomplish this using enciphered 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.

Prosecution History of the '486 Invention

215. The prior art identified during prosecution of the '486 patent did not disclose "video data with a plurality of partially encrypted frames, wherein each partially encrypted frame contains encrypted portions and unencrypted portions of data; and 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, and a block reference that identifies the encrypted portion of the partially encrypted frame," as recited in claims 1 and 10 of the '486 patent (later amended and issued as claims 1 and 15), and the claims that depend from those claims. '486 File History,[45] Notice of Allowance, Nov. 21, 2018, at 8-9.

[45] Cited excerpts of the '486 file history attached as Exhibit 15.

216. During prosecution, the patent examiner did not reject any claims of the '486 patent under 35 U.S.C. § 101. The '486 patent issued on November 10, 2015, after the U.S. Supreme Court's decision in *Alice Corp. Pty Ltd. v. CLS Bank Int'l*, 573 U.S. 208 (2014).

Claims Reciting the Technical Solutions of the '486 Invention

217. The claims of the '486 patent recite these computing improvements that provide content security benefits for video transmission and decoding. Claim 1 of the '486 patent recites how an improved playback device decrypts and decodes the invention's new container file structure containing encrypted digital video:

> 1. A playback device for playing back encrypted video,
> the playback device comprising:
> a set of one or more processors; and
> a non-volatile storage containing a playback application
> for causing the set of one or more processors to perform
> the steps of:
> receiving a container file with video data at a parser;
> extracting portions of the container file using the parser,
> wherein the container file comprises:
> video data with a plurality of partially encrypted frames,
> wherein each partially encrypted frame contains
> encrypted portions and unencrypted portions of data; and
> 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, and
> a block reference that identifies the encrypted portion of

> the partially encrypted frame,
> 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;
> identifying the encrypted portion of each partially encrypted frame using the block reference for each partially encrypted frame;
> 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;
> decrypting the encrypted portion of each partially encrypted frame based upon the frame key for each partially encrypted frame using the video decoder; and
> decoding each decrypted frame for rendering on a display device using the video decoder.

'486 patent, 10:55-11:26.

218. Claim 1 recites how to improve content security during video decoding by using a novel container file format including encrypted video that is decrypted at the decoder by "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.* 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.

219. Claims 2-14 of the '486 patent depend from claim 1, and each of claims 2-14 further describes how the invention's improved playback device is configured to decrypt and play back the new container file structure containing encrypted digital video that improves security of the video content during decoding and playback. The ordered combination of elements in each of claims 2-14, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer configurations and video container file structures that were not well-understood at the time of the '486 invention.

- Claim 2 depends from claim 1 and further describes how the improved playback device is configured to process the improved video container file, reciting "each partially encrypted frame is provided by the parser to a video decoder over an unsecured channel." *Id.* at 11:27-29.
- Claim 3 depends from claim 1 and further describes the structure of the new video container file for decryption by the improved playback device, reciting "each block reference comprises offset and length information." *Id.* at 11:30-31.
- Claim 4 depends from claim 1 and further describes how the improved playback device is configured to process the improved video container file, reciting "the playback application is further for causing the set of processors to communicate with a digital rights management component to decipher a frame key for each partially encrypted frame from the cryptographic material for each partially encrypted frame." *Id.* at 11:32-37.
- Claims 5 and 6 depend from claim 1 and further describe the structure of the new video container file for decryption by the improved playback device, reciting "the frame key is encrypted to restrict playback to a particular user." *Id.* at 11:38-42.

- Claim 7 depends from claim 1 and further describes how the improved playback device is configured to process the improved video container file, reciting "the playback application is further for causing the set of one or more processors to stream the container file." *Id.* at 11:43-45.
- Claim 8 depends from claim 1 and further describes how the improved playback device is configured to process the improved video container file, reciting "the playback application is further 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 by building a cryptographic payload comprising: cryptographic material for a partially encrypted frame, and a block reference for the partially encrypted frame." *Id.* at 11:46-56.
- Claim 9 depends from claim 8 and further describes how the improved playback device is configured to process the improved video container file, reciting "the cryptographic payload is delimited by an identifier." *Id.* at 11:57-58.
- Claim 10 depends from claim 9 and further describes how the improved playback device is configured to process the improved video container file, reciting "the decoder uses the identifier to extract cryptographic material for the partially encrypted frame and the block reference for the partially encrypted frame from the cryptographic payload." *Id.* at 11:59-62.
- Claim 11 depends from claim 1 and further describes how the improved playback device is configured to process the improved video container file, reciting "the playback application is further for causing the set of one or more processors to perform the step of inserting the

cryptographic payload at the front of each partially encrypted frame of video that is demultiplexed by the parser." *Id.* at 11:63-67.

- Claim 12 depends from claim 1 and further describes how the improved playback device is configured to process the improved video container file, reciting "inserting the cryptographic payload at the front of each partially encrypted frame of video using the parser." *Id.* at 12:1-3.
- Claim 13 depends from claim 1 and further describes the structure of the new video container file for decryption by the improved playback device, reciting "an encrypted portion of a partially encrypted frame comprises a frame header." *Id.* at 12:4-5.
- Claim 14 depends from claim 1 and further describes the structure of the new video container file for decryption by the improved playback device, reciting "an unencrypted portion of a partially encrypted frame comprises a frame header." *Id.* at 12:6-8.

220. Claim 15 of the '486 patent recites how to perform an improved method of playing back video encrypted in the new container file structure of the invention:

> 15. A method for playing back encrypted video, the method comprising:
> receiving a container file with video data at a parser;
> extracting portions of the container file using the parser, wherein the container file comprises:
> video data with a plurality of partially encrypted frames, wherein each partially encrypted frame contains encrypted portions and unencrypted portions of data; and
> 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, and

a block reference that identifies the encrypted portion of the partially encrypted frame,

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;

identifying the encrypted portion of each partially encrypted frame using the block reference for each partially encrypted frame;

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;

decrypting the encrypted portion of each partially encrypted frame based upon the frame key for each partially encrypted frame using the video decoder; and

decoding each decrypted frame for rendering on a display device using the video decoder.

Id. at 12:9-42.

221. Claim 15 recites how to improve content security during video decoding by using the new container file structure, reciting "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.* That is, the keys necessary to decrypt the video are protected until they are

deciphered on the device. The invention recited in claim 15 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 invention.

222. Claims 16-25 of the '486 patent depend from claim 15, and each of claims 16-25 further describes how to perform the invention's improved method for playing back the new container file structure containing encrypted digital video that improves security of the video content during decoding and playback on a video decoder. The ordered combination of elements in each of claims 16-25, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved computer processes and video container file structures that were not well-understood at the time of the '486 invention.

- Claim 16 depends from claim 15 and further describes the structure of the new video container file for decryption and playback by the improved method, reciting "each partially encrypted frame is provided by the parser to a video decoder over an unsecured channel." *Id.* at 12:43-45.
- Claim 17 depends from claim 15 and further describes the structure of the new video container file for decryption and playback by the improved method, reciting "each block reference comprises offset and length information." *Id.* at 12:46-47.
- Claim 18 depends from claim 15 and further describes how the improved method decrypts and plays back the new video container file, reciting "communicating with a digital rights management component to decipher a frame key for each partially encrypted frame from the cryptographic material for each partially encrypted frame." *Id.* at 12:48-52.

- Claims 19 and 20 depend from claim 15 and further describe the structure of the new video container file for decryption and playback by the improved method, reciting "the frame key is encrypted to restrict playback to a particular user." *Id.* at 12:53-56.
- Claim 21 depends from claim 15 and further describes how the improved method decrypts and plays back the new video container file, reciting "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 further comprises building a cryptographic payload comprising: cryptographic material for a partially encrypted frame, and a block reference for the partially encrypted frame." *Id.* at 12:57-65.
- Claim 22 depends from claim 21 and further describes how the improved method decrypts and plays back the new video container file, reciting "the cryptographic payload is delimited by an identifier." *Id.* at 12:66-67.
- Claim 23 depends from claim 22 and further describes how the improved method decrypts and plays back the new video container file, reciting "extracting cryptographic material for the partially encrypted frame and the block reference for the partially encrypted frame from the cryptographic payload based upon the identifier using the video decoder." *Id.* at 13:1-5.
- Claim 24 depends from claim 15 and further describes the structure of the new video container file for decryption and playback by the improved method, reciting "an encrypted portion of a partially encrypted frame comprises a frame header." *Id.* at 13:6-7.
- Claim 25 depends from claim 15 and further describes the structure of the new video container file for decryption and playback by the

improved method, reciting "an unencrypted portion of a partially encrypted frame comprises a frame header." *Id.* at 13:8-10.

IX. The '588 Patent

223. 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.

Summary of the '588 Invention

224. The '588 claims are directed to "perform[ing] adaptive bitrate streaming using alternative streams of protected content." '588 patent, 2:66-3:1. The '588 invention reduces the complexity of the cryptographic information needed to ensure content security for multiple, alternative video streams so that a user performing ABS experiences fewer stalls, delays, or errors caused by processing of the cryptographic information on the playback device. The '588 patent is directed to a new encryption architecture for digital video streams that uses common frame encryption keys to encode alternate video streams, reducing playback stalls and improving performance during ABS. The inventions recited in the '588 patent enable Netflix to offer its users an improved experience for ABS while maintaining the content security that it and other content providers require to make video content available over the internet.

Technical Problems Addressed by the '588 Invention

225. The '588 patent addresses a technical problem: providing content security while reducing the computational burdens of processing cryptographic information for alternative video streams during ABS. "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., id.*

at 1:45-58. In ABS, "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 ABS 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.

Technical Solutions and Benefits Provided by the '588 Invention

226. The '588 patent claims a solution to this problem with playback device implementations and methods that reduce the computer memory and other resources consumed by cryptographic information during ABS. The '588 claims are directed to improvements to the functionality of computer systems that perform digital video decryption and playback during ABS. More specifically, the '588 claims are directed to a new index file structure and a new structure of encrypted data for ABS, how a playback device is configured to request, decrypt, and play back video data using the new structures (claim 1 and dependents), and how to request, decrypt, and play back video data using the new structures (claim 12 and dependents).

227. The new index file structure and a new structure of encrypted data of the '588 invention incorporates alternative video streams including partially encrypted video frames that are encrypted using a set of common keys, a top level index identifying those streams, and a container index containing byte ranges for

portions of a stream. 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. Prior ABS video encryption formats and index files did not encrypt alternative streams using a set of common keys. The new index file structure and new structure of encrypted data of the '588 patent, and the devices and methods used to process the new index file structure and encrypted data structure, therefore were not well-known, routine, and conventional at the time of the '588 invention.

228. The new index file structure and new encrypted data structure of the '588 invention, and the devices and methods used to process the new index file structure and encrypted data structure, provide technical benefits that improve the functionality and capabilities of computer systems performing these operations. Encrypting alternative video streams using a set of common keys, and identifying those encrypted streams using a top level index file, allows playback devices to switch between alternative video streams during ABS and to decrypt those streams without having to perform the computationally intensive processes of obtaining and processing additional cryptographic information, while maintaining the security of the video content. *Id.* at 8:55-61, 10:22-31. The new files of the '588 invention, and new methods for processing those files, therefore reduce the computing resources needed to provide ABS while providing content security. The '588 invention, therefore, allows an ABS 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. *Id.*

Prosecution History of the '588 Invention

229. Claims 1 and 12 of the '588 patent and the claims that depend from claims 1 and 12 issued, among other reasons, because "[n]one of the prior art of record, either taken by itself or in any combination, would have anticipated or made

obvious the invention of the present application at or before the time it was filed." '588 File History,[46] Notice of Allowability, Nov. 19, 2018, at 11.

230. During prosecution, the patent examiner rejected pending claims of the '588 patent under 35 U.S.C. § 101. '588 File History, Notice of Allowance, Dec. 18, 2018, at 3. After an interview, the examiner proposed an examiner's amendment to the claims that removed the rejection under § 101. *Id.* The '588 patent issued on March 5, 2019, with incorporation of the examiner's amendments, after the U.S. Supreme Court's decision in *Alice Corp. Pty Ltd. v. CLS Bank Int'l*, 573 U.S. 208 (2014).

Claims Reciting the Technical Solutions of the '588 Invention

231. The '588 claims recite methods and systems setting forth how to improve ABS using partial-frame encryption with common encryption keys to improve performance of the playback device displaying the video streams. Claim 1 of the '588 patent recites how an improved playback device is configured to play back video using the new index file structure and new encrypted data structure of the invention:

> 1. A playback device for playing protected content from a plurality of alternative streams, comprising:
> a set of one or more processors; and
> a non-volatile storage containing an application for causing the set of one or more processors to perform the steps 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

[46] Cited excerpts of the '588 file history attached as Exhibit 16.

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;

obtaining a copy of the set of common keys;

detecting streaming conditions for the playback device;

selecting a stream from the plurality of alternative streams of protected video based on the detected streaming conditions;

receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file;

requesting portions of the selected stream of protected video based on the provided byte ranges;

locating encryption information that identifies encrypted portions of frames of video within the requested portions of the selected stream of protected video;

decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys; and

playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video.

'588 patent, 27:30-63.

232. Claim 1 employs a new kind of file (partially encrypted video frames encrypted using a set of common keys) that enables a playback device to do things it could not do before. Specifically, the new file type and the use of byte ranges provide the playback device with newly available computing resources during ABS. *Id.* at Abstract, 2:66-3:30, 8:37-61, 9:65-10:31. The claim element "partially

encrypted video frames that are encrypted using a set of common keys" allows the playback device to store and process common cryptographic information instead of multiple sets of cryptographic information for each video stream. *Id.* Claim 1, therefore, solves the problem of inefficient and low-performance video playback caused by the use of different cryptographic information for each video stream in an ABS service in a manner that was not well-understood, routine, or conventional at the time of the '588 patent. *Id.* at 10:22-31.

233. Claims 2-11 and 23 of the '588 patent depend from claim 1, and each of claims 2-11 and 23 further describe how an improved playback device is configured to play back video using the new index file structure and new encrypted data structure, reducing playback stalls and improving performance during ABS. The ordered combination of elements in each of claims 2-11 and 23, in conjunction with the elements of the claims from which they depend, therefore recite new and improved computer processes and video stream structures that were not well-understood at the time of the '588 invention.

- Claim 2 depends from claim 1 and further describes how the improved playback device is configured to play back video using the new index file structure and new encrypted data structure, reciting "wherein the step of requesting portions of the selected stream further comprises the step of obtaining a container file containing protected video from at least one of the plurality of alternative streams, where the container file also includes encryption information that identifies portions of frames of video that are encrypted and a reference to at least one key from the set of common keys to utilize in accessing the encrypted portions of the frames of video." *Id.* at 27:64-28:5.
- Claim 3 depends from claim 1 and further describes the structure of the new encrypted data structure, reciting "wherein the located

encryption information comprises a reference to the start of an encrypted block of data." *Id.* at 28:6-8.

- Claim 4 depends from claim 3 and further describes the structure of the new encrypted data structure, reciting "wherein the located encryption information further comprises the size of the encrypted block of data." *Id.* at 28:9-11.
- Claim 5 depends from claim 3 and further describes the structure of the new encrypted data structure, reciting "wherein the located encryption information further comprises cryptographic information that can be utilized to access the encrypted portion of the frame." *Id.* at 28:12-15.
- Claim 6 depends from claim 5 and further describes the structure of the new encrypted data structure, reciting "wherein the cryptographic information is a reference to at least one key from the set of common keys." *Id.* at 28:16-18.
- Claim 7 depends from claim 1 and further describes how the improved playback device is configured to play back video using the new index file structure and new encrypted data structure, reciting "wherein the application is further for causing the set of processors to perform the steps of: detecting a change in the streaming conditions; identifying a second alternative stream of protected video based on the detected change; receiving a container index that provides byte ranges for portions of the second alternative stream of protected video within a second associated container file; requesting portions of the second alternative stream of protected video based on the provided byte ranges; decrypting each encrypted portion of the frames of video from the requested portions of the second alternative stream of protected video using the set of common keys; and playing back the decrypted

frames of video from the requested portions of the second alternative stream of protected video." *Id.* at 28:19-36.

- Claim 8 depends from claim 1 and further describes how the improved playback device is configured to play back video using the new index file structure and new encrypted data structure, reciting "wherein the application is further for causing the set of processors to perform the steps of: transmitting a request for content to a set of one or more content distribution servers; and receiving the content from the set of one or more content distribution servers." *Id.* at 28:37-43.
- Claim 9 depends from claim 8 and further describes how the improved playback device is configured to play back video using the new index file structure and new encrypted data structure, reciting "wherein the application is further for causing the set of processors to perform the steps of: transmitting a request for content to a set of one or more content distribution servers; and receiving the content from the set of one or more content distribution servers." *Id.* at 28:44-51.
- Claim 10 depends from claim 1 and further describes how the improved playback device is configured to play back video using the new index file structure and new encrypted data structure, reciting "wherein the application is further for causing the set of processors to perform the step of obtaining the container index from at least one file selected from the group consisting of: the associated container file containing the selected stream of protected content; and a separate file to the associated container file containing the selected stream of protected content." *Id.* at 28:52-58.
- Claim 11 depends from claim 1 and further describes the structure of the new encrypted data structure, reciting "wherein the common set of keys comprises a plurality of keys." *Id.* at 28:59-60.

234. Claim 12 of the '588 patent recites how to perform an improved method for playing, on a playback device, video using the new index file structure and new encrypted data structure of the invention:

> 12. A method for playing protected content from a plurality of alternative streams on a playback device, the method comprising:
>
> 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;
>
> obtaining a copy of the set of common keys;
>
> detecting streaming conditions for the playback device;
>
> selecting a stream from the plurality of alternative streams of protected video based on the detected streaming conditions;
>
> receiving a container index that provides byte ranges for portions of the selected stream of protected video within an associated container file;
>
> requesting portions of the selected stream of protected video based on the provided byte ranges;
>
> locating encryption information that identifies encrypted portions of frames of video within the requested portions of the selected stream of protected video;

> decrypting each encrypted portion of the frames of video identified within the located encryption information using the set of common keys; and
>
> playing back the decrypted frames of video obtained from the requested portions of the selected stream of protected video using a decoder.

Id. at 28:61-29:23.

235. The method in claim 12 employs a new kind of file (partially encrypted video frames encrypted using a set of common keys) that enables a playback device to do things that it could not do before. Specifically, the new file type and the use of byte ranges provide the playback device with newly available computing resources during ABS. *Id.* at Abstract, 2:66-3:30, 8:37-61, 9:65-10:31. The claim element "partially encrypted video frames that are encrypted using a set of common keys" allows the playback device to store and process common cryptographic information instead of multiple sets of cryptographic information for each video stream. *Id.* Claim 12 of the '588 patent, therefore, recites a novel solution for inefficient and low-performance video playback caused by the use of different cryptographic information for each video stream in an ABS service in a manner that was not well-understood, routine, or conventional at the time of the '588 patent. *Id.* at 10:22-31.

236. Claims 13-22 and 24 of the '588 patent depend from claim 12, and each of claims 13-22 and 24 further describe how to perform the invention's improved method for playing, on a playback device, video using the new index file structure and new encrypted data structure of the invention, reducing playback stalls and improving performance during ABS. The ordered combination of elements in each of claims 13-22 and 24, in conjunction with the elements of the claims from which they depend, therefore recite unconventional new and improved

computer processes and video stream structures that were not well-understood at the time of the '588 invention.

- Claim 13 depends from claim 12 and further describes how to perform the improved method for playing video using the new index file structure and new encrypted data structure, reciting "wherein requesting portions of the selected stream further comprises obtaining a container file containing protected video from at least one of the plurality of alternative streams, where the container file also includes encryption information that identifies portions of frames of video that are encrypted and a reference to at least one key from the set of common keys to utilize in accessing the encrypted portions of the frames of video." *Id.* at 29:24-31.
- Claim 14 depends from claim 12 and further describes the structure of the new encrypted data structure, reciting "wherein the located encryption information comprises a reference to the start of an encrypted block of data." *Id.* at 29:32-34.
- Claim 15 depends from claim 14 and further describes the structure of the new encrypted data structure, reciting "wherein the located encryption information further comprises the size of the encrypted block of data." *Id.* at 29:35-37.
- Claim 16 depends from claim 14 and further describes the structure of the new encrypted data structure, reciting "wherein the located encryption information further comprises cryptographic information that can be utilized to access the encrypted portion of the frame." *Id.* at 29:38-41.
- Claim 17 depends from claim 16 and further describes the structure of the new encrypted data structure, reciting "wherein the cryptographic

information is a reference to at least one key from the set of common keys." *Id.* at 29:42-44.

- Claim 18 depends from claim 12 and further describes how to perform the improved method for playing video using the new index file structure and new encrypted data structure, reciting "detecting a change in the streaming conditions; identifying a second alternative stream of protected video based on the detected change; receiving a container index that provides byte ranges for portions of the second alternative stream of protected video within a second associated container file; requesting portions of the second alternative stream of protected video based on the provided byte ranges; decrypting each encrypted portion of the frames of video from the requested portions of the second alternative stream of protected video using the set of common keys; and playing back the decrypted frames of video from the requested portions of the second alternative stream of protected video." *Id.* at 29:45-30:9.
- Claim 19 depends from claim 12 and further describes how to perform the improved method for playing video using the new index file structure and new encrypted data structure, reciting "transmitting a request for content to a set of one or more content distribution servers; and receiving the content from the set of one or more content distribution servers." *Id.* at 30:10-14.
- Claim 20 depends from claim 19 and further describes how to perform the improved method for playing video using the new index file structure and new encrypted data structure, reciting "further comprising obtaining the container index from at least one file selected from the group consisting of: the associated container file containing the selected stream of protected content; and a separate file to the

associated container file containing the selected stream of protected content." *Id.* at 30:15-21.

- Claim 21 depends from claim 12 and further describes how to perform the improved method for playing video using the new index file structure and new encrypted data structure, reciting "further comprising obtaining the container index from at least one file selected from the group consisting of: the associated container file containing the selected stream of protected content; and a separate file to the associated container file containing the selected stream of protected content." *Id.* at 30:22-27.
- Claim 22 depends from claim 12 and further describes the structure of the new encrypted data structure, reciting "wherein the common set of keys comprises a plurality of keys." *Id.* at 30:28-29.
- Claim 24 depends from claim 12 and further describes the structure of the new encrypted data structure, reciting "wherein: the container index is part of a hierarchical index; and the method further comprises: obtaining a lower layer index that identifies the location of frames within a specific requested portion of the selected stream of protected video; and identify partially encrypted video frames from within the specific requested portion of the selected stream using the lower layer index." *Id.* at 30:41-50.

NETFLIX'S INTERNAL TESTING

237. 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.

238. 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.

239. 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."[47]

240. 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: [48]

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

[48] *Id.*

241. As of August 10, 2016, Netflix employees estimated that the Netflix ecosystem ran approximately 20,000 test cases per day.[49]

242. 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.

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

244. 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.[50] Upon information and belief, Netflix's agreements with studios require that Netflix agree to provide secure DRM to protect content.[51]

---

[49] *Id.*

[50] 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.

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

245. 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.[52]

246. Indeed, Netflix identifies any compromise to its ability to obtain content as one a material risk to Netflix's business.[53] 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

247. 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.

248. 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

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

---

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

[53] *Id.*

250. 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.

251. 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

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

253. 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."[54]

254. 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

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

available through television set-top boxes.[55] Netflix also has entered into agreements with other consumer electronics device manufacturers to make Netflix's service available on those consumer devices.[56] Those products include streaming media players, smart TVs, game consoles, Blu-ray players, smartphones and tablets, and personal computers.[57] Netflix recommends, directly to consumers, certain consumer electronics devices preloaded with Netflix.[58]

255. 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."[59] 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.

256. 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":[60]

---

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

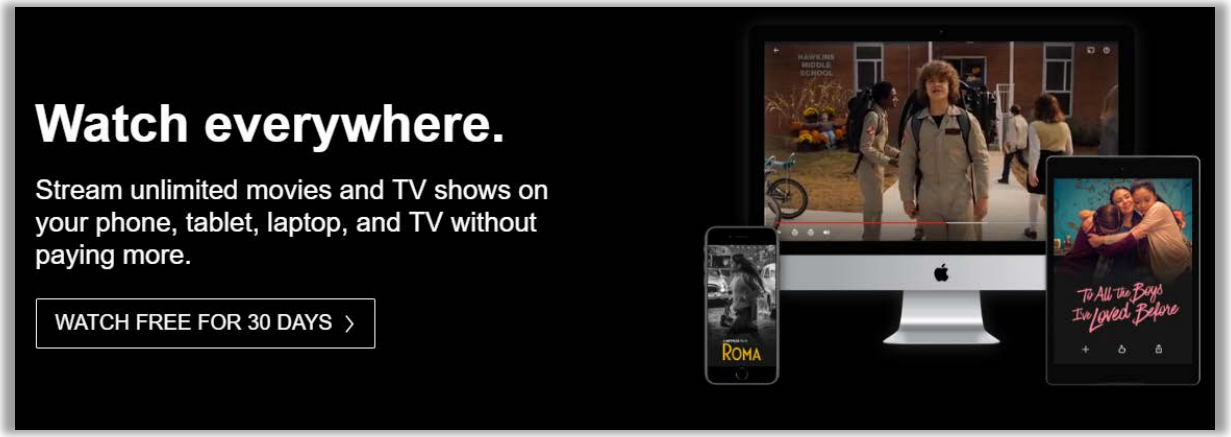
[56] https://devices.netflix.com/en/.

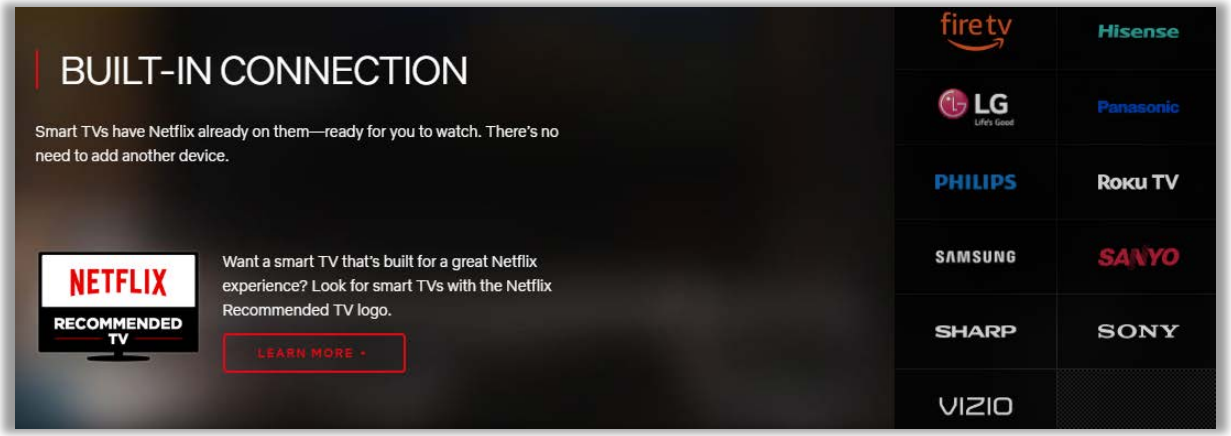
[57] *Id.*

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

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

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





257. 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.

258. 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.

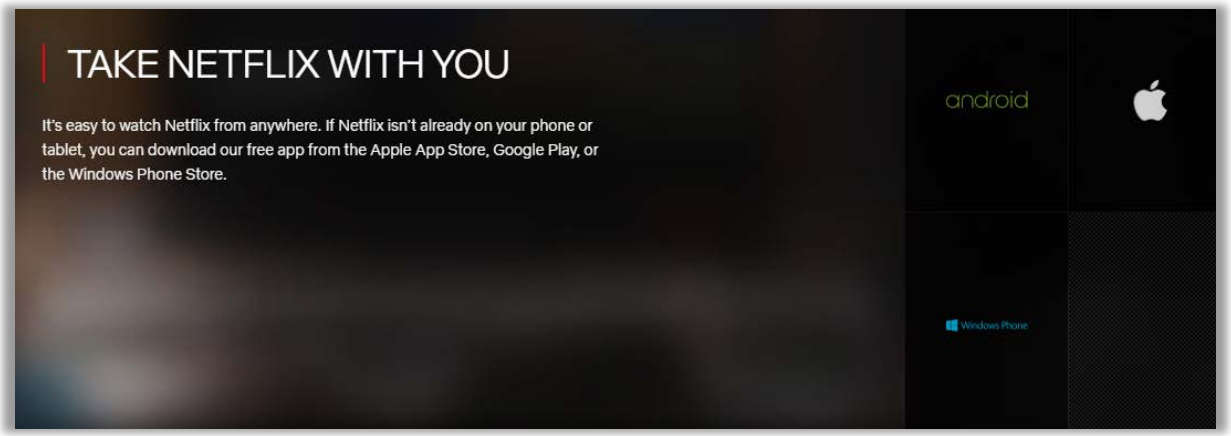
III. Netflix's Specific Intent to Cause Third-Party Actions Infringing DivX's Patents

259. 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.

260. 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.

261. 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:[61]

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



262. 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.[62]

263. 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.

264. 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

265. The allegations of paragraphs 1-264 of this Complaint are incorporated by reference as though fully set forth herein.

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

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

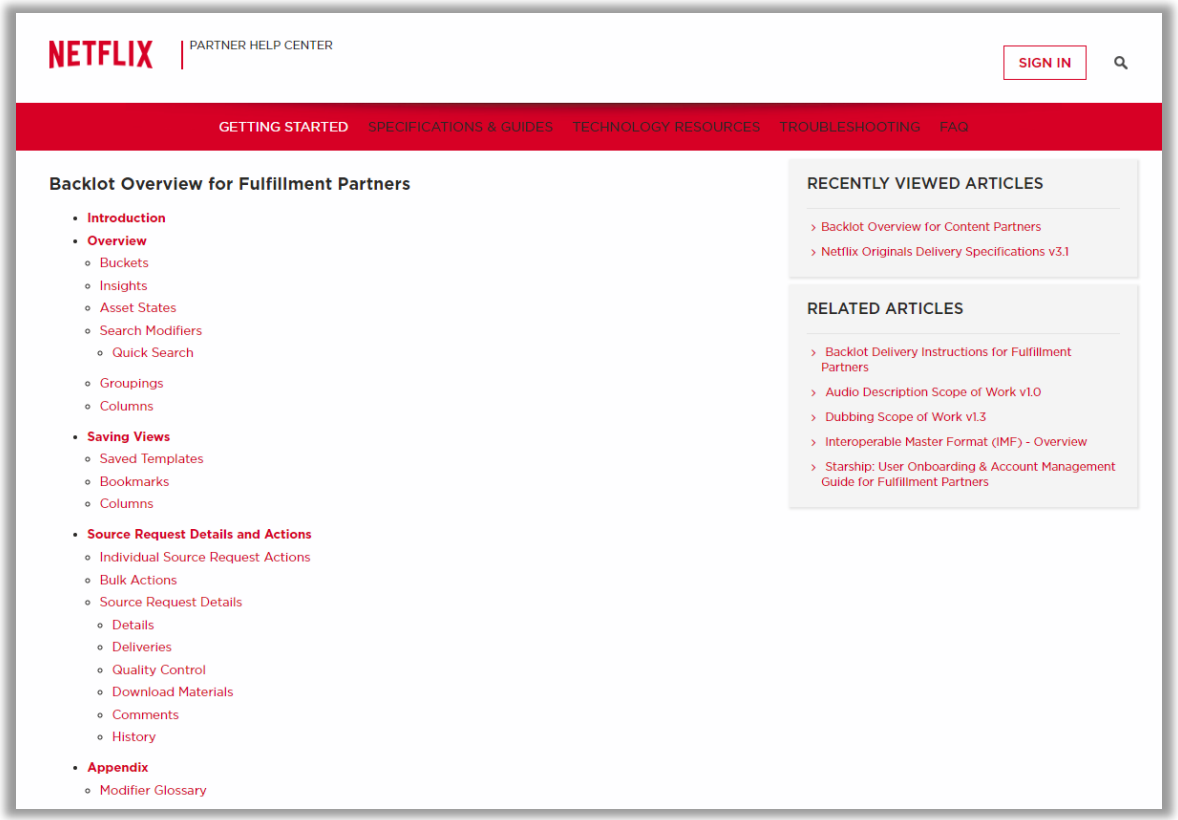
267. Upon information and belief, Netflix directly infringes the '673 patent by making, using, offering to sell, selling, and/or importing into the United States its Netflix service, which provides a method and system for securing compressed digital video (collectively, the "Accused '673 Infringing Products").

268. 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 269-276 below.

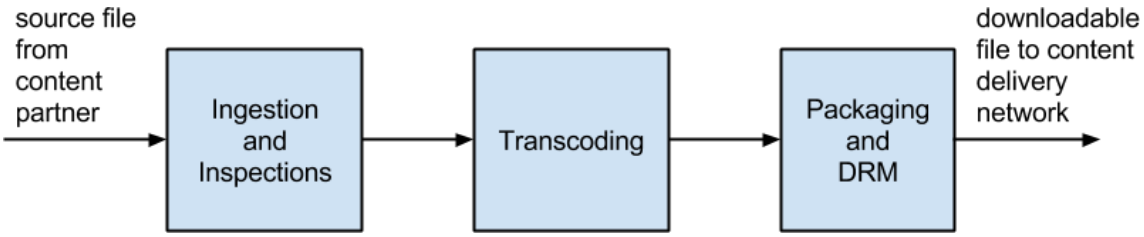
269. Netflix provides a "method for producing a protected stream of compressed video content," namely, Netflix encoding, encrypting, and packaging videos in the H.265 (HEVC) format for streaming.

270. 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.[63]

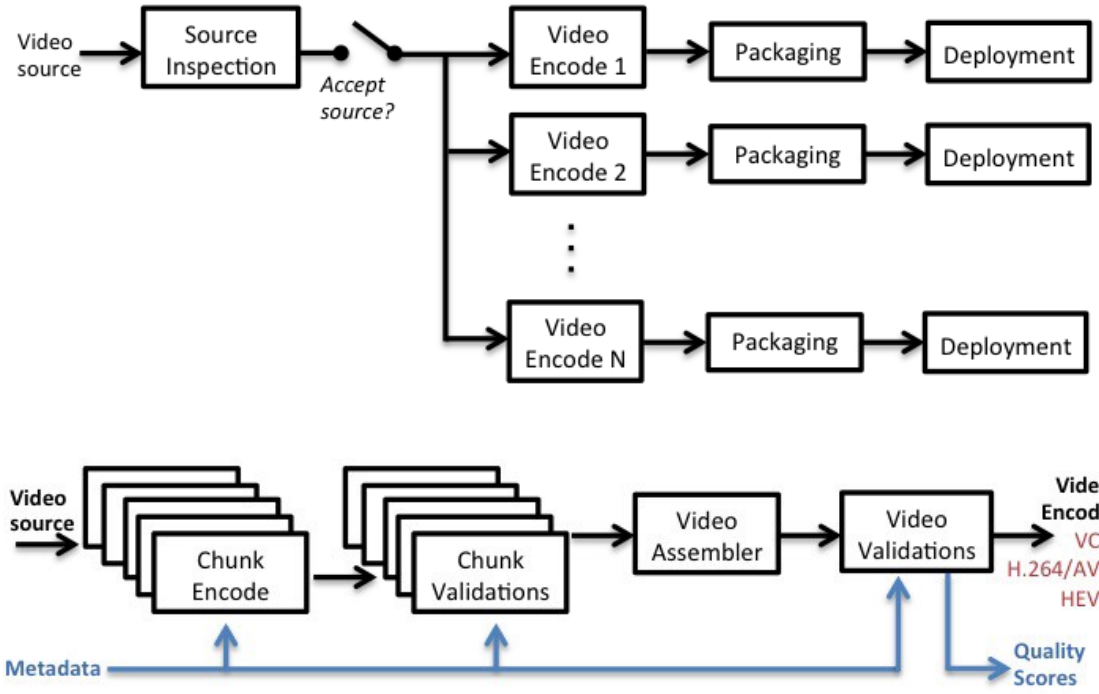
[63] https://partnerhelp.netflixstudios.com/hc/en-us/articles/115004872247-Backlot-Overview-for-Fulfillment-Partners#Intro.



271. Netflix encodes videos via, for example, an H.265 codec, which becomes an input stream of compressed video content containing a sequence of frames.[64]



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



272. 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.

273. 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.

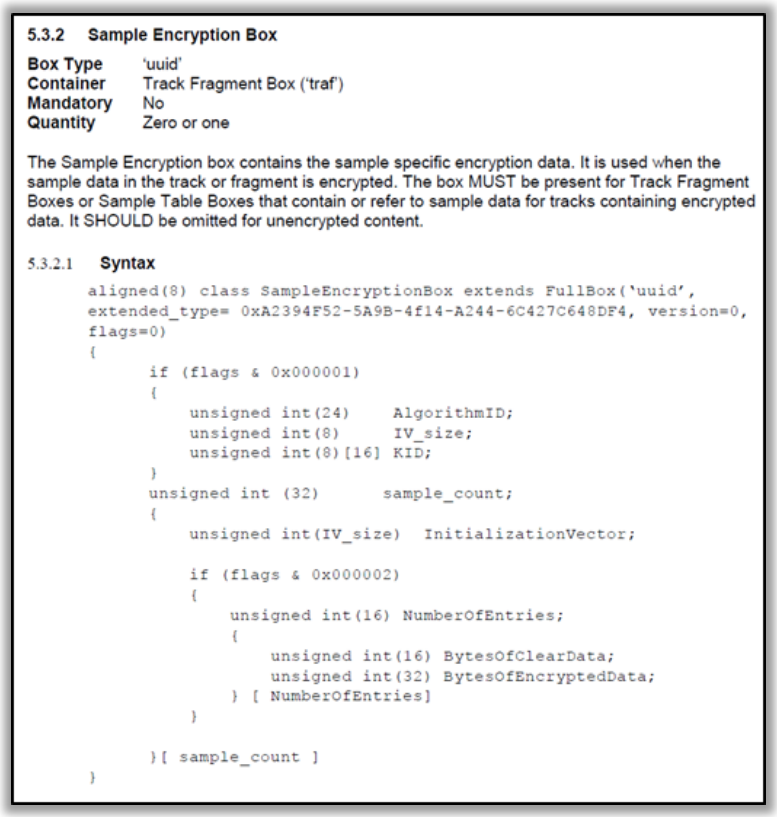
274. Netflix "generat[es] frame decryption information necessary to decrypt said set of encrypted frames including an encryption key pointer identifying a decryption key to be used in the decryption of each encrypted frame" by generating frame decryption information that includes (1) a pointer to a decryption 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.

275. 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.

276. 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.[65]

[65] 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 ]
}
```

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

278. 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.

279. 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).

280. 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 281-288, in violation of 35 U.S.C. § 271(b).

281. 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.

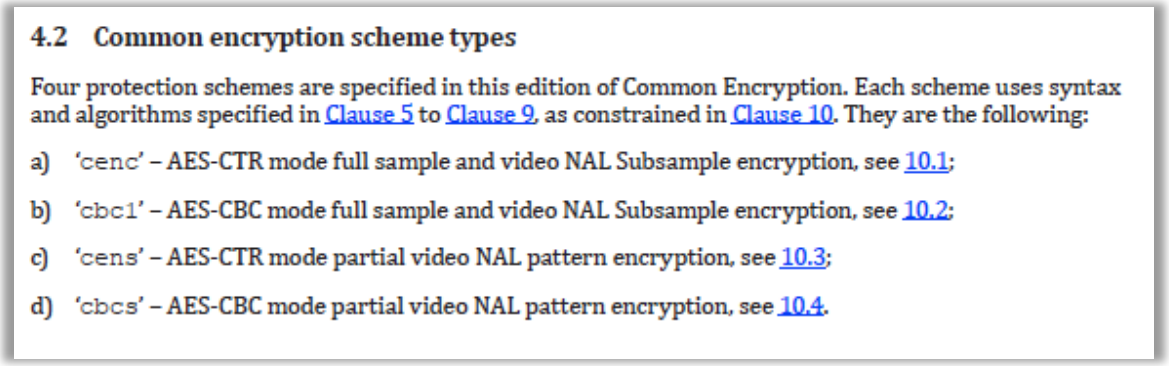
282. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, meets the limitations of claim 14.

283. 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.

284. 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.[66] 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.

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

285. 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.

286. 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.[67]

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

287. 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.

288. 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.

289. 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

290. The allegations of paragraphs 1-289 of this Complaint are incorporated by reference as though fully set forth herein.

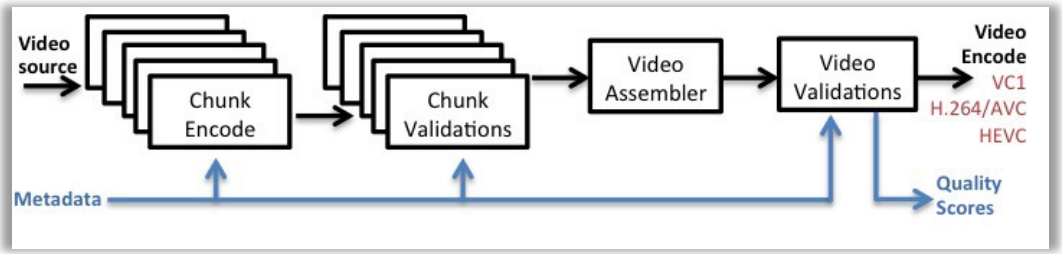
291. Pursuant to 35 U.S.C. § 282, the '651 patent is presumed valid.

292. 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").

293. 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 294-297 below.

294. 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.[68]



[68] 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.

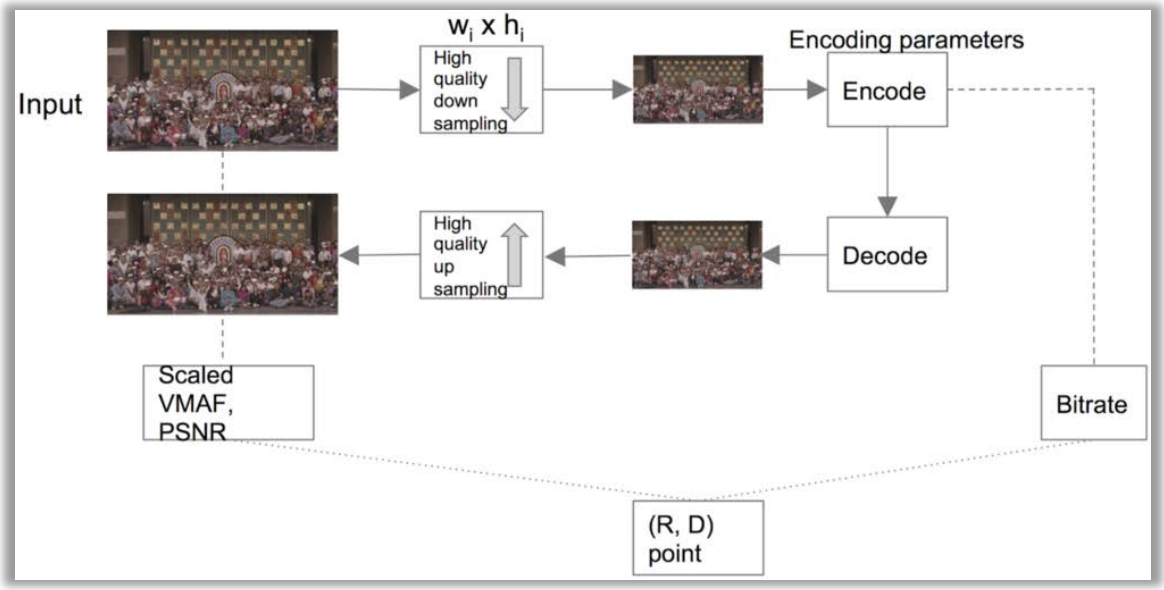
Parameter	Value
Video codec	AVC-High/x264, VP9/libvpx, HEVC/x265
Encoding mode	Shot-based
Encoding recipe	AQ-mode=0, fixed QP/CRF, CPU speed varying by resolution
Quality metric - aggregation strategy	Linear VMAF (LVMAF) @ 1920x1080 display resolution
Baseline	Fixed-QP encode (middle QP value for each 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.[69]

	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

[69] 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.[70] 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).[71]



[70] 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.").

[71] 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.[72] 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:[73]

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

[73] 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").

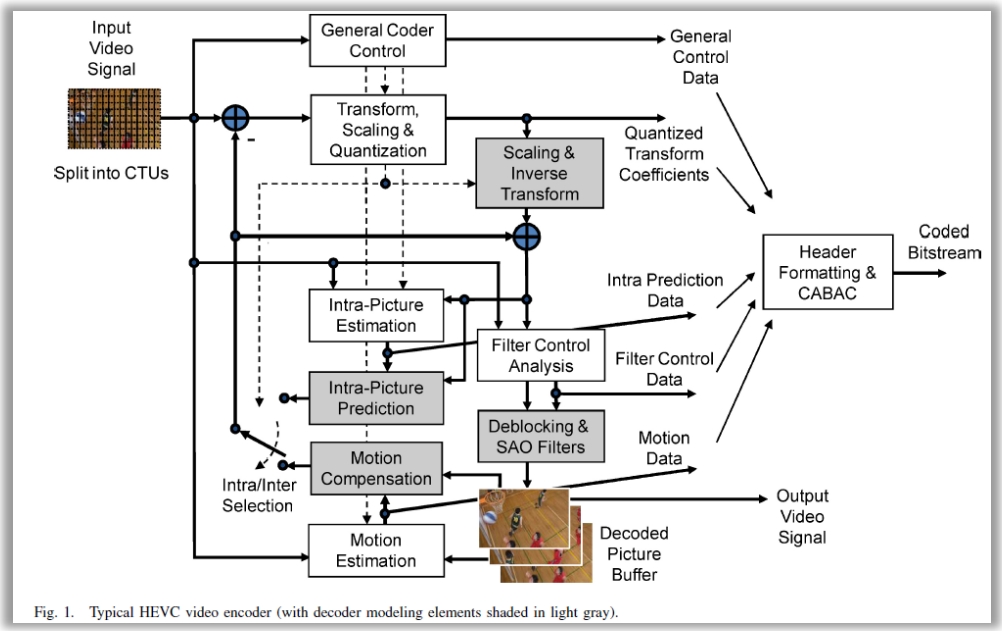


Fig. 1. Typical HEVC video encoder (with decoder modeling elements shaded in light gray).

More specifically, the encoding profile "main" within the H.265 (HEVC) Standard requires a deblocking filter.[74] 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.

295. 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.[75]

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

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

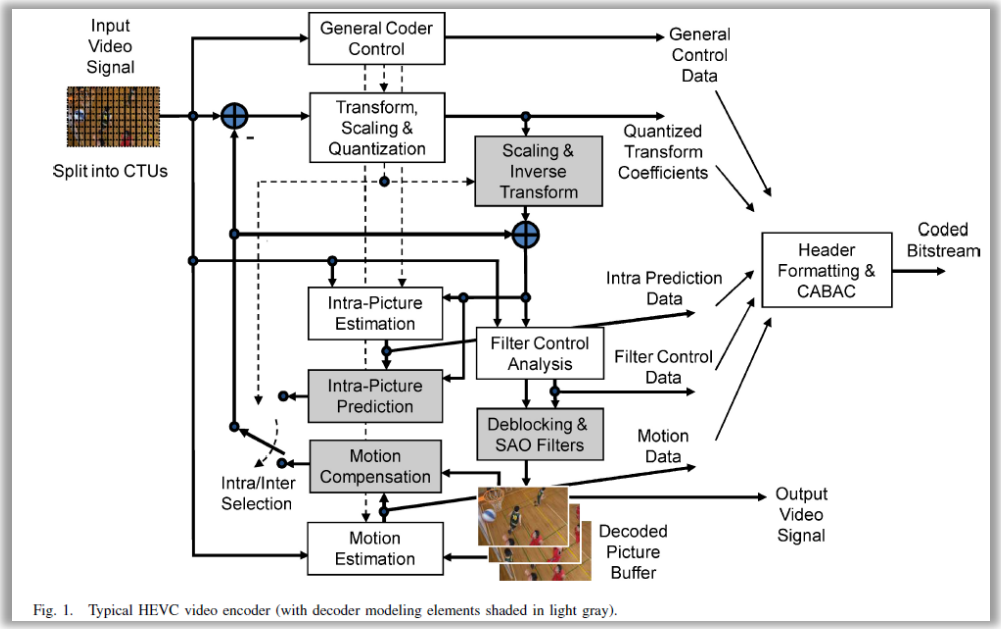


Fig. 1. Typical HEVC video encoder (with decoder modeling elements shaded in light gray).

The deblocking filter modifies a reconstructed video frame according to the deblocking filter process, including filtering the boundaries of the video frame.[76] 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.[77]

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

[77] *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").

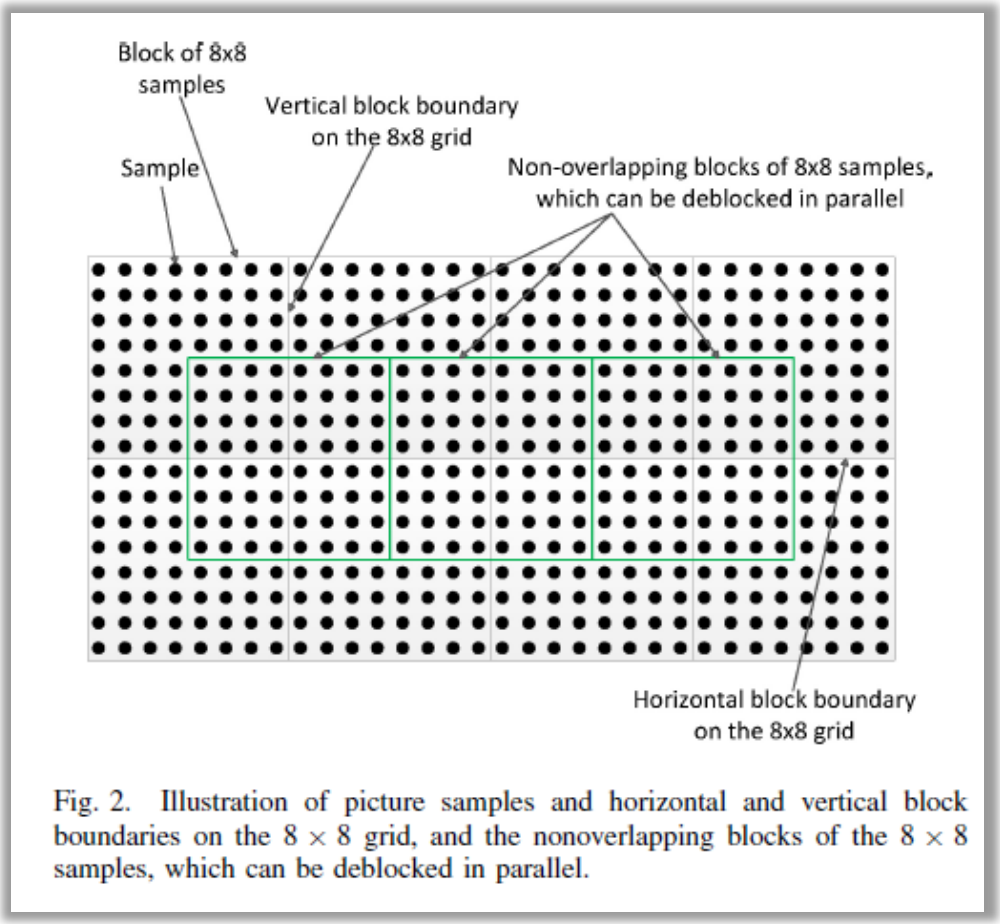


Fig. 2. Illustration of picture samples and horizontal and vertical block boundaries on the 8 × 8 grid, and the nonoverlapping blocks of the 8 × 8 samples, which can be deblocked in parallel.

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.[78]

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

296. 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.[79] 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.[80] 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.[81]

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

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

[81] *Id.* at 1748.

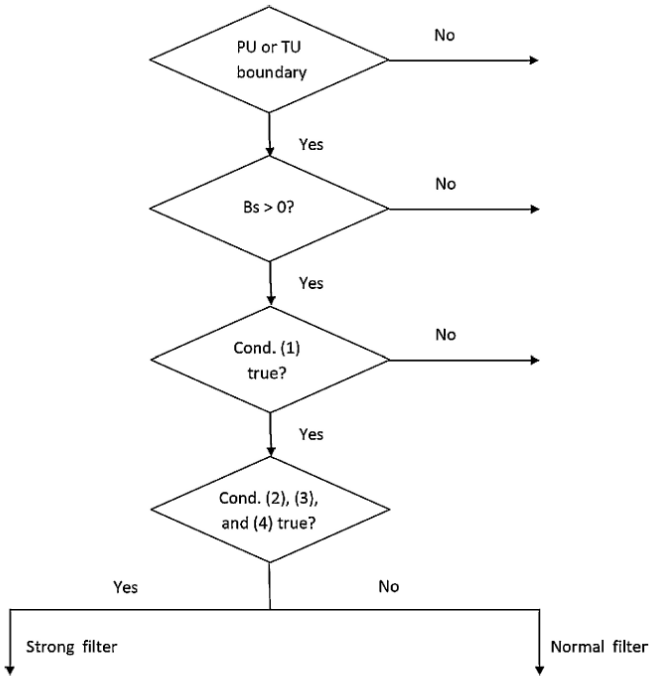


Fig. 4. Decisions for each segment of block boundary of four samples in length lying on 8 × 8 block boundary. PU: prediction unit. TU: transform unit.

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

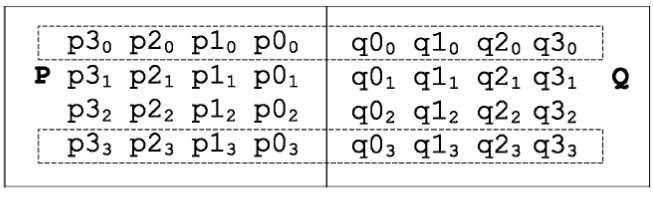


Fig. 3. Four-pixel long vertical block boundary formed by the adjacent blocks P and Q. Deblocking decisions are based on lines marked with the dashed line (lines 0 and 3).

An illustration of the multiple rows (row 0 & 3) and multiple columns (cols 0 & 3) involved in such determination is shown below.[82] Column 0 pixels are immediately adjacent to at least two sides of the block boundary. Column 3 pixels are not.

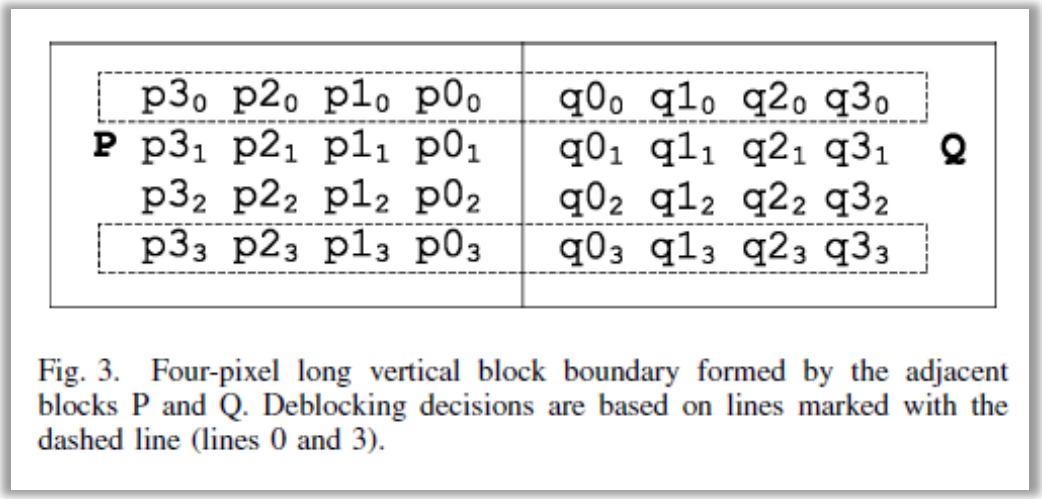


Fig. 3. Four-pixel long vertical block boundary formed by the adjacent blocks P and Q. Deblocking decisions are based on lines marked with the dashed line (lines 0 and 3).

297. 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.[83]

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

299. 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.

[82] *Id.*

[83] *Id.* at 1749.

300. 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).

301. 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 302-304, in violation of 35 U.S.C. § 271(b).

302. 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.

303. 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.

304. 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 292-298.

305. 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

306. The allegations of paragraphs 1-305 of this Complaint are incorporated by reference as though fully set forth herein.

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

308. 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").

309. 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 310-315 below.

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

311. Netflix's encoder comprises "a processor." Netflix encodes movies and other titles using computers with processors, as the Netflix Tech Blog confirms:[84]

> **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.

312. 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,"

[84] 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.[85]

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.

[85] 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.

313. 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.[86]

[86] *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.

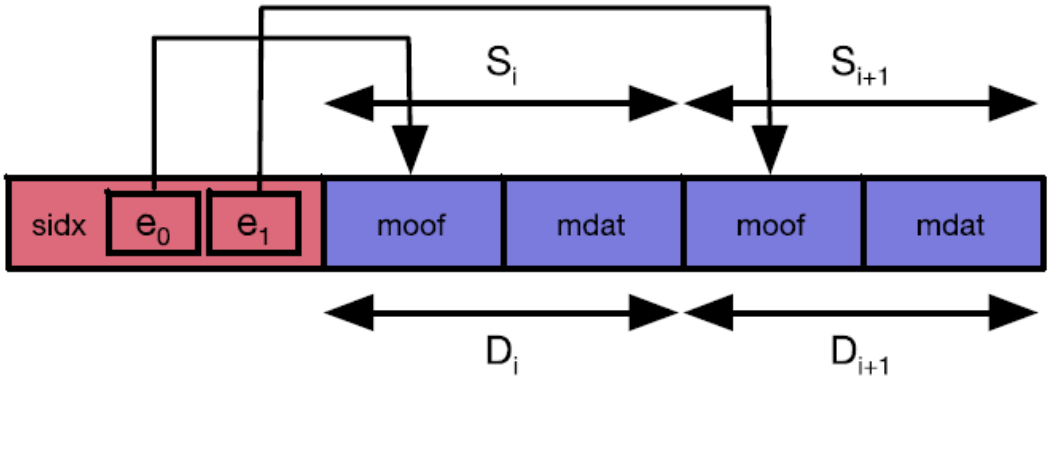


Figure K. 1: Simple Segment Index

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.[87] The ssix

[87] *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.

314. 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).

315. 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.[88]

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

317. 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.

318. 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).

319. 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 320-327, in violation of 35 U.S.C. § 271(b).

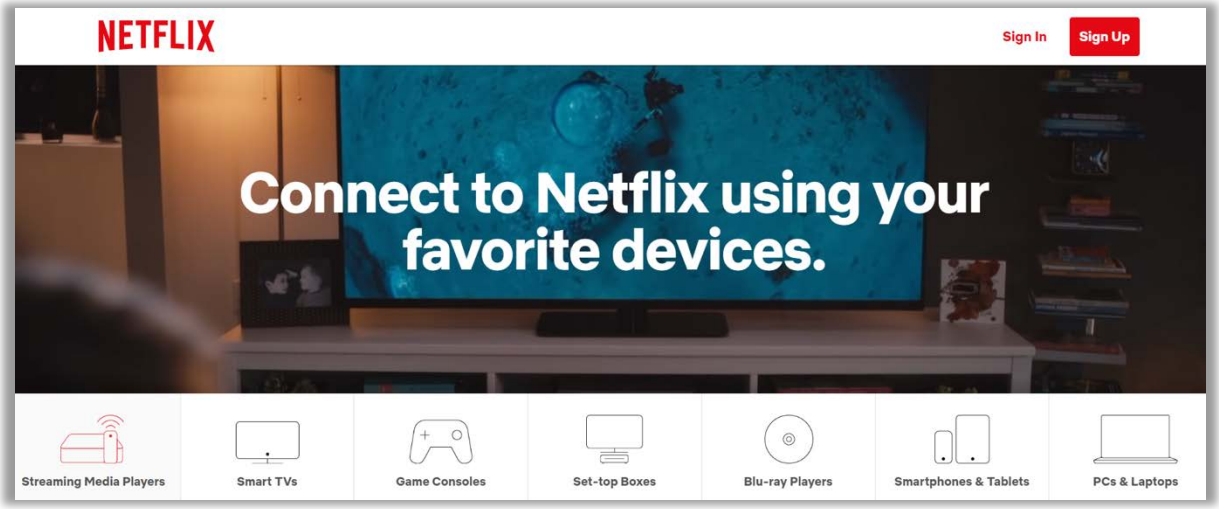
320. At least as of the date of this Complaint, Netflix knows that the '792 patent enables playback features that video streaming users expect, enjoy, and use to navigate digital video easily, and they improve the user experience by reducing delays in loading and playing a video when it is selected by 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.

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

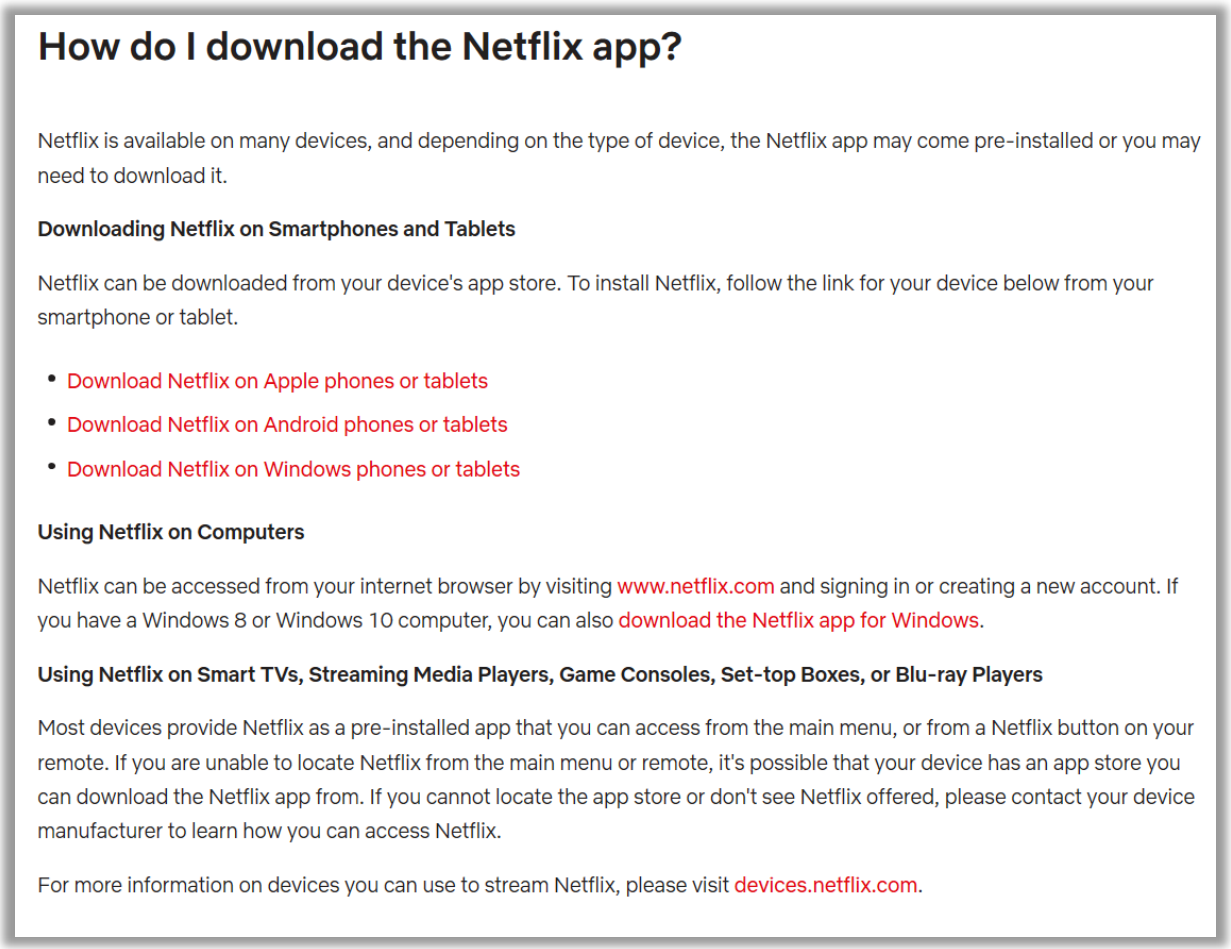
321. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service for CE playback devices that, when used as intended, meets the limitations of claim 15.

322. 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.

323. 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.[89]



[89] 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.

324. 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.[90] Upon information and belief, Netflix video streams contain mdat boxes.

325. 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

[90] 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.[91] 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.

326. 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.

327. 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.

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

328. 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

329. The allegations of paragraphs 1-328 of this Complaint are incorporated by reference as though fully set forth herein.

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

331. 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").

332. 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 333-342 below.

333. 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.[92] Due to

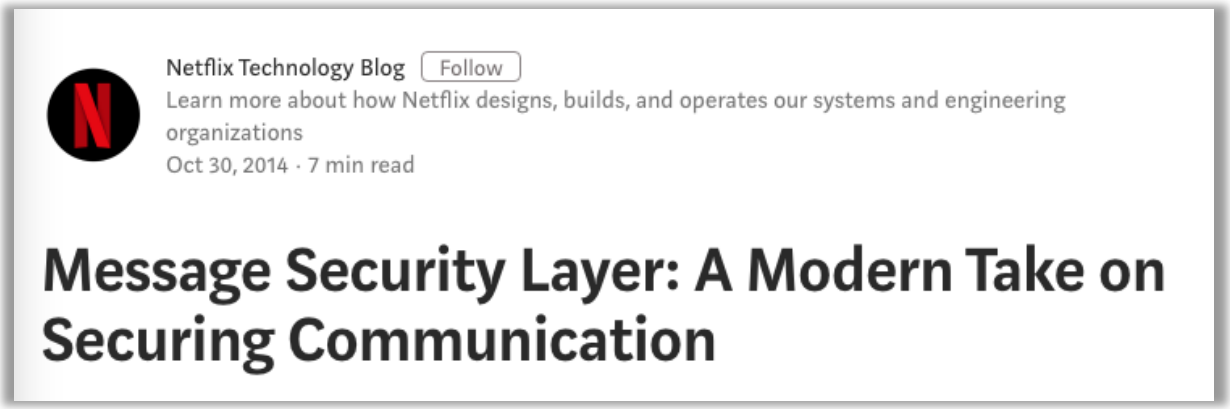
[92] *See* ISO/IEC 23009-1 (2014) Information technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and

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:[93]

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).

[93] 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:[94]

[94] 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:[95]

[95] *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."[96]

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:[97]

[96] *Id.*

[97] 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:[98]

[98] *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:[99]

[99] 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.[100]

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"

[100] *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.[101] 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

[101] *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.

334. 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:[102]

> 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

[102] 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.

335. 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.

336. 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.

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

338. 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.

339. 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.

340. 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.

341. 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.

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

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

344. 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.

345. 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).

346. 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 347-349, in violation of 35 U.S.C. § 271(b).

347. 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.

348. 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.

349. 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 333-342.

350. 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

351. The allegations of paragraphs 1-350 of this Complaint are incorporated by reference as though fully set forth herein.

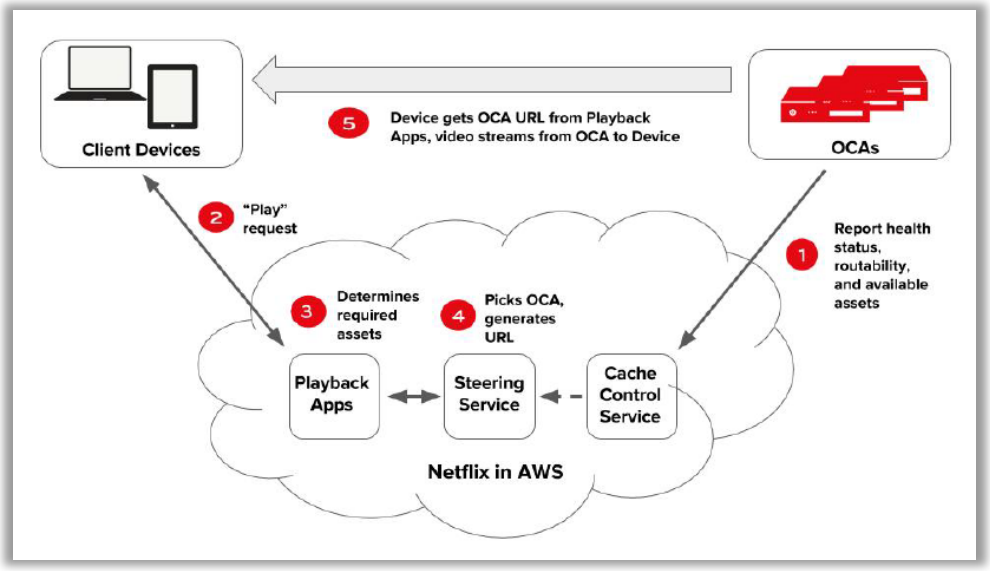
352. Pursuant to 35 U.S.C. § 282, the '720 patent is presumed valid.

353. 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").

354. 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 355-360 below.

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

356. 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."[103]

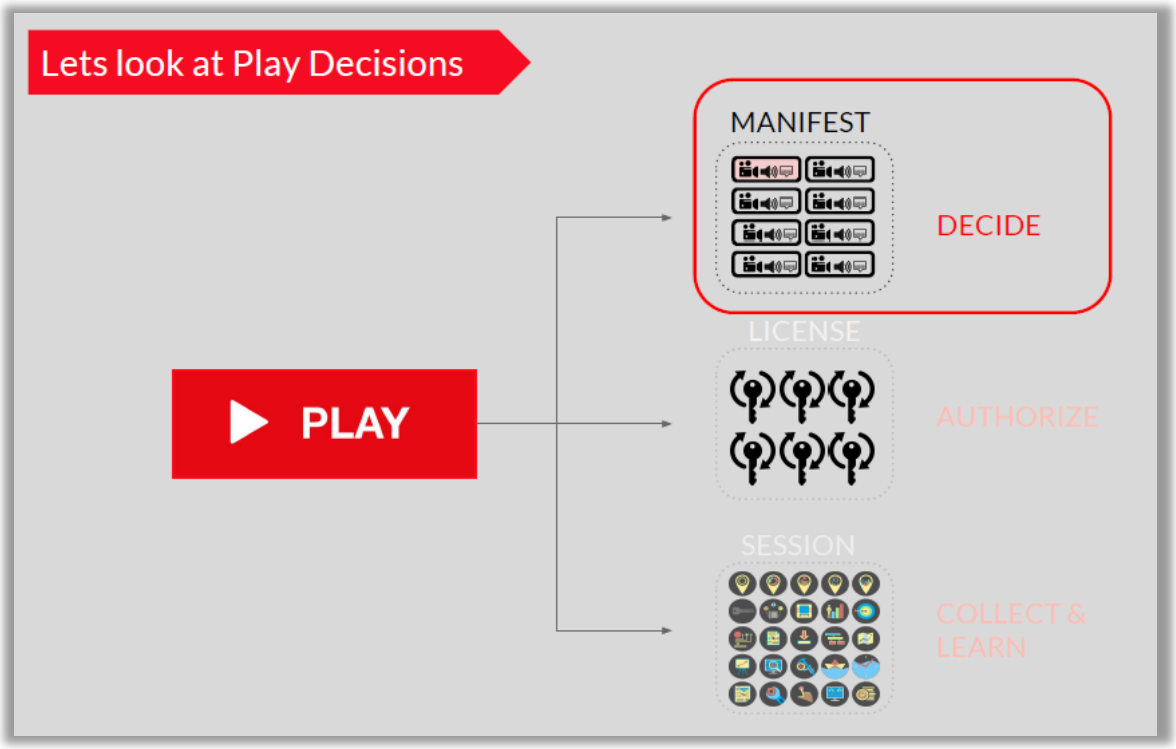


[103] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 2, 4.

> 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.

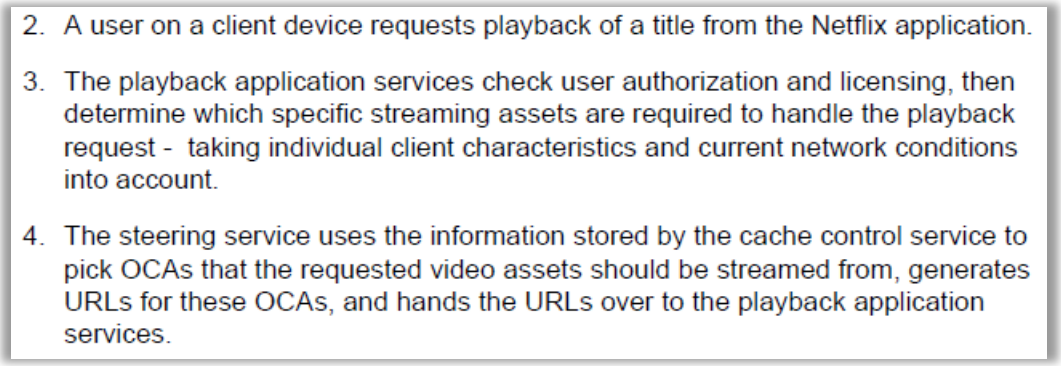
The playback device could be, for example, a PC running Windows 10 using the Edge browser from Microsoft. The request identifies a piece of content, for example, using a movie*ID.* And the request includes a product identifier, for example, Win10 PC/Edge. The request includes information necessary to 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:[104]

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



357. 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:[105]

[105] *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 movie*ID.* 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.[106]

[106] 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:

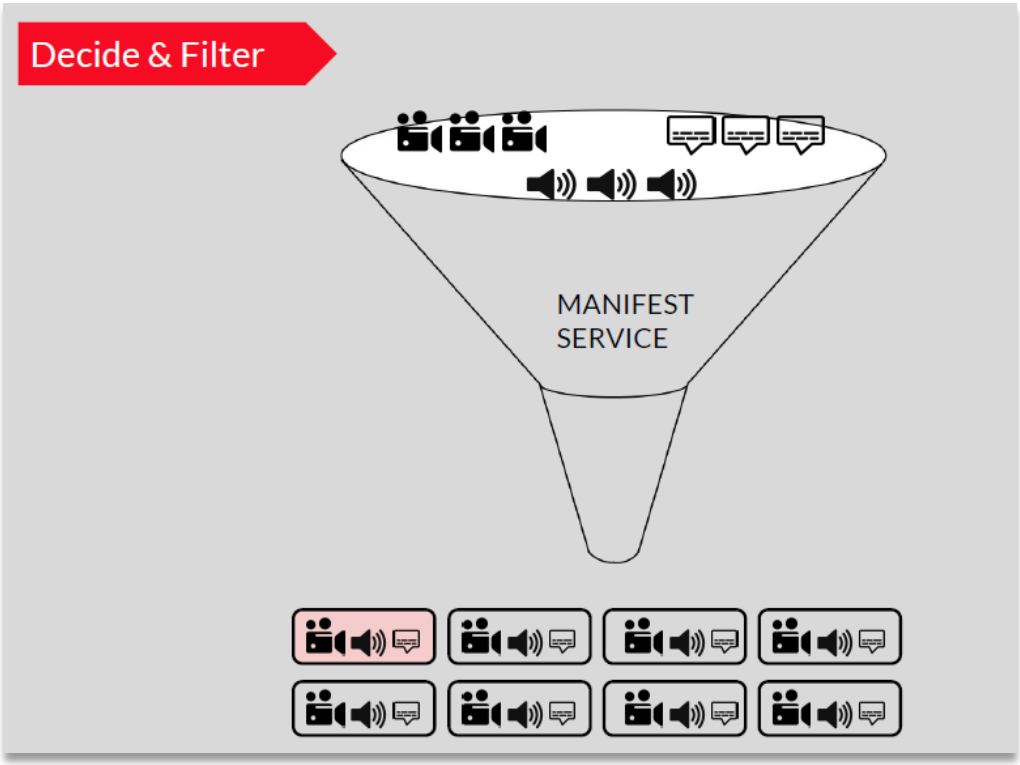
Monitor 1	Monitor 2	Expected Stream Quality
HDCP2.2 (active)	HDCP2.2 (active)	4K UHD
HDCP2.2 (active)	HDCP1.X (connected but not active)	4K UHD
HDCP2.2 (active)	HDCP1.X (active)	FHD

SLI configuration

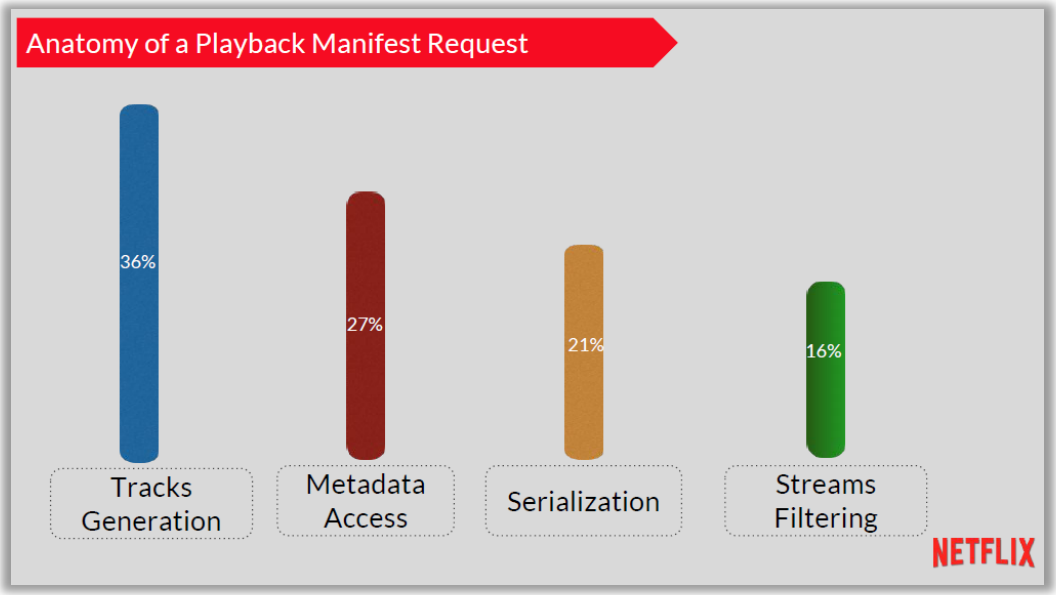
Currently, 4K UHD streaming is not supported for SLI/LDA configurations. However, if the GPUs are not linked together in SLI/LDA mode, 4K UHD streaming will work fine if all of the active monitors are HDCP2.2 capable.

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.

358. 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.[107]



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

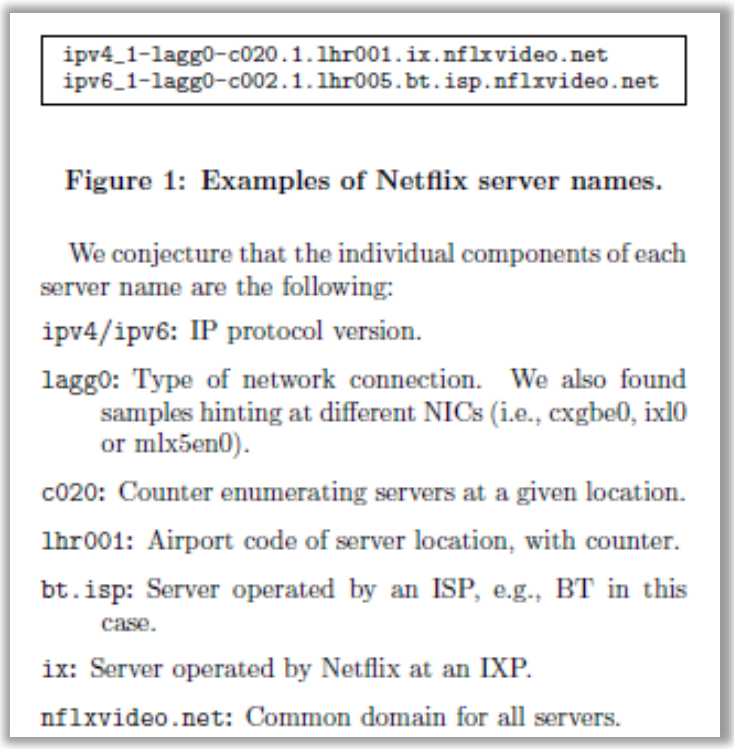


Netflix generates a different manifest based on the device capability (or capabilities) using its playback server system. Netflix filters the list of assets (bitrate/resolution/format) based on the device capabilities. For example, as described in the previous paragraph, Netflix will stream 4K/UltraHD 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.[108] Netflix's playback server system maintains a database of product identifiers and associated device capabilities.

359. 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.[109]

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

[109] *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.

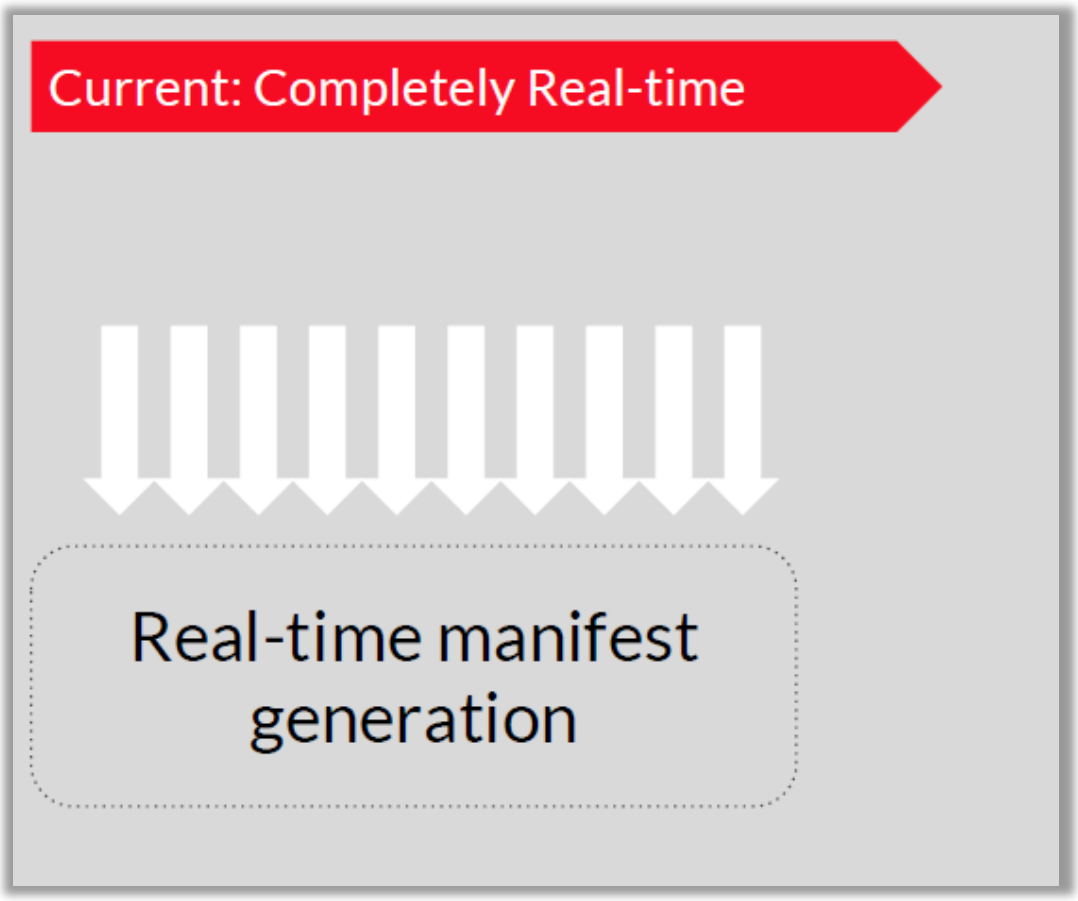
`ix`: Server operated by Netflix at an IXP.

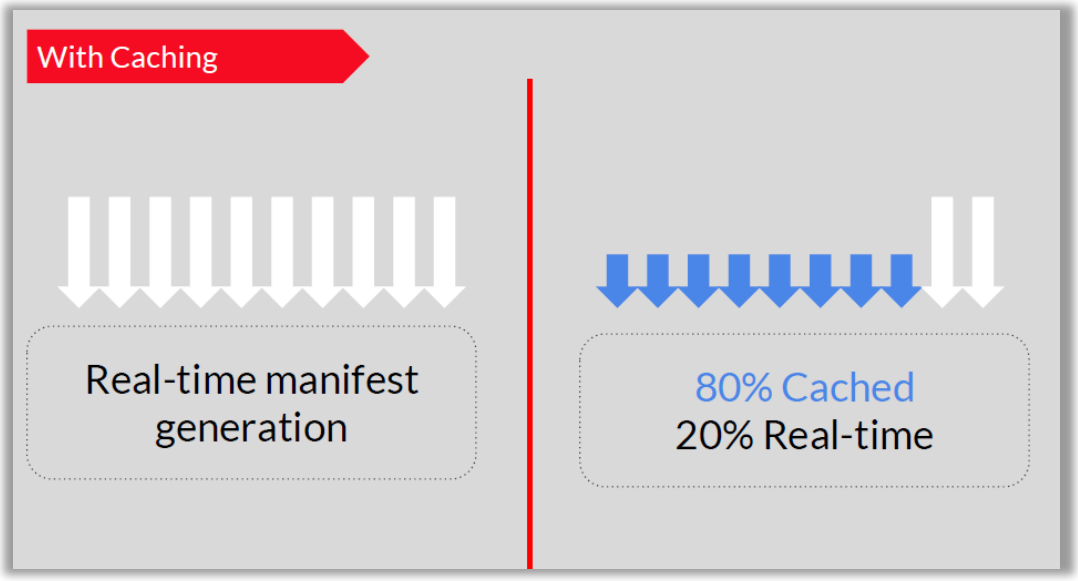
`nflxvideo.net`: Common domain for all servers.

As illustrated in the following exemplary Netflix presentation, Netflix generates the manifest, either in real-time or pre-cached:[110]

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

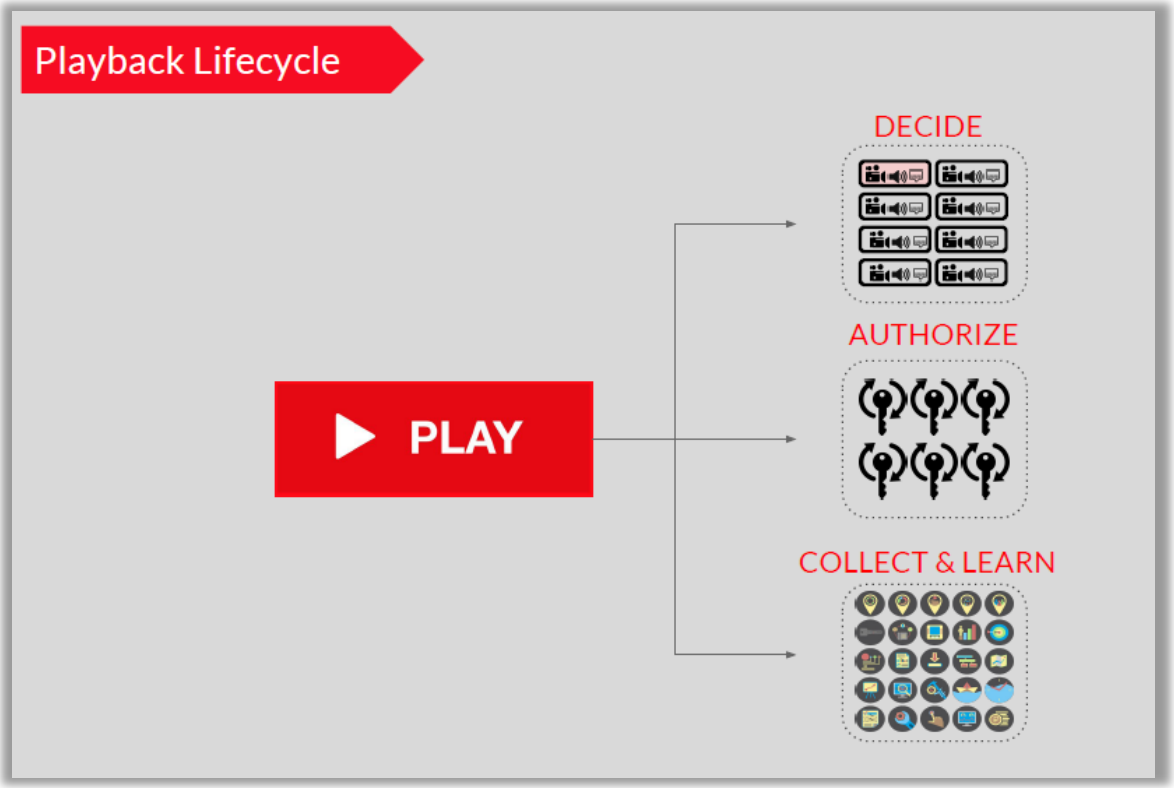
[110] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 17-18.





360. 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:[111]

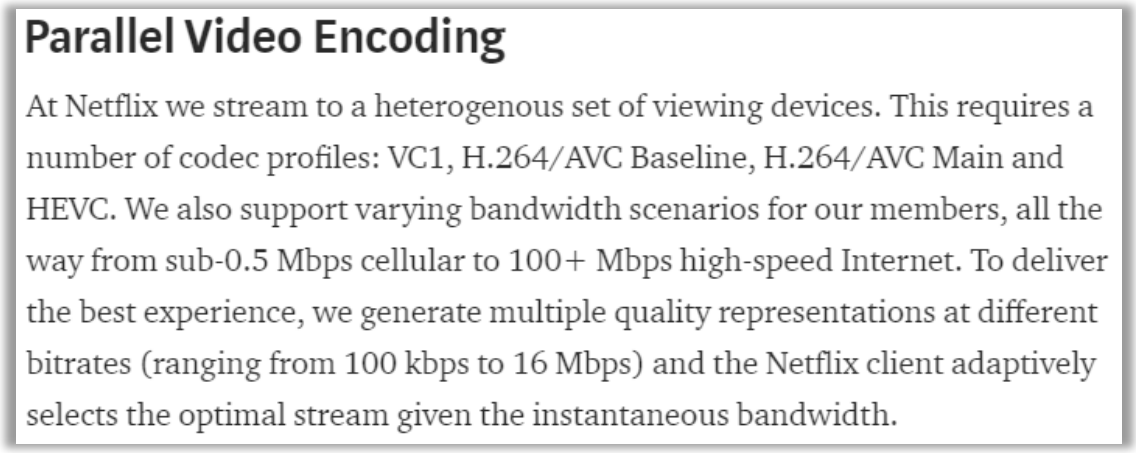


Further, Netflix admits that it does adaptive streaming using the manifests, suggesting that the playback device uses the manifest to determine the assets to request for playback on the device:[112]

---

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

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



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

362. 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.

363. 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).

364. 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

365. The allegations of paragraphs 1-364 of this Complaint are incorporated by reference as though fully set forth herein.

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

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

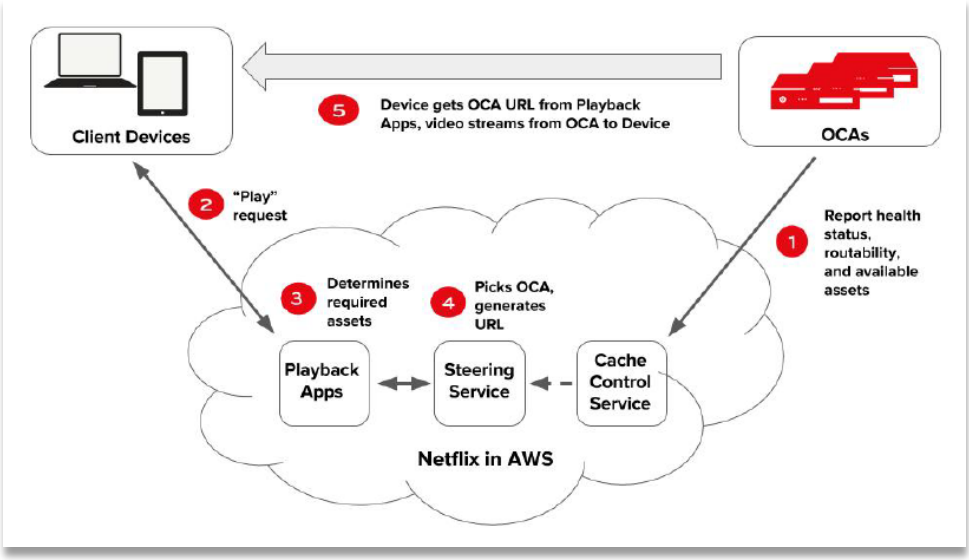
its Netflix service, which provides a system and method for automatically generating top level index files (collectively, the "Accused '515 Infringing Products").

368. 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 369-375 below.

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

370. 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."[113]

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



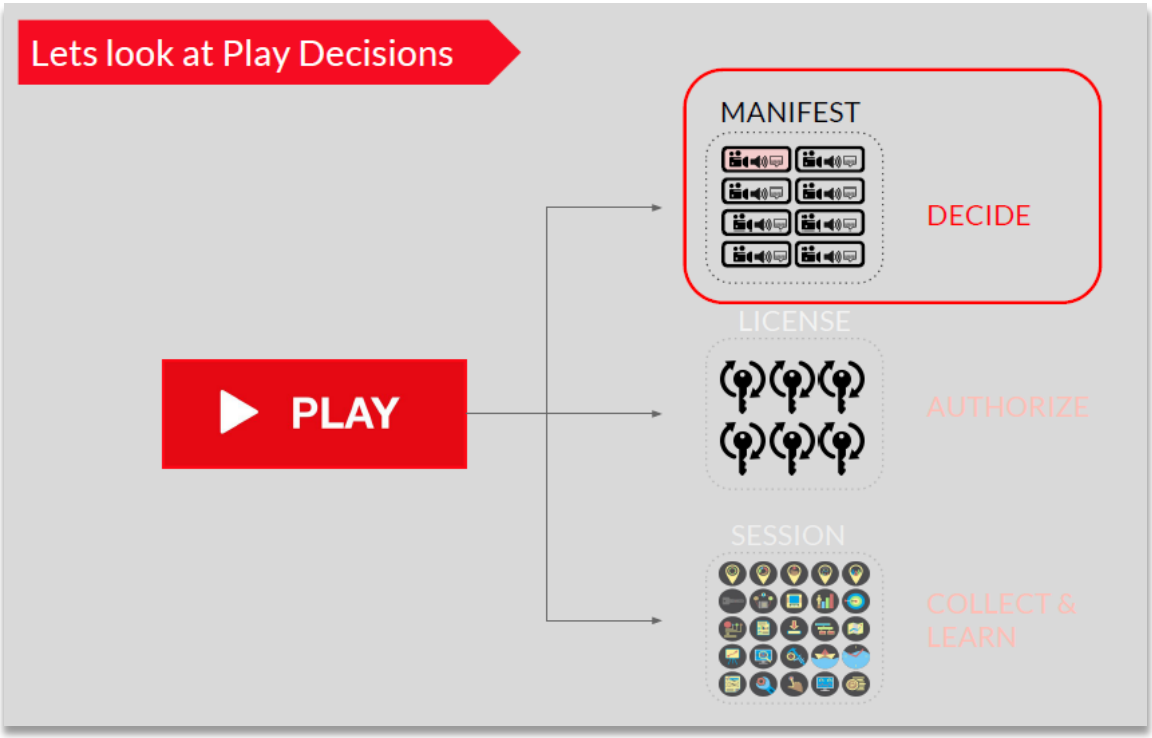
2. A user on a client device requests playback of a title from the Netflix application.

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 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:[114]

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



371. 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.

372. 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.[115]

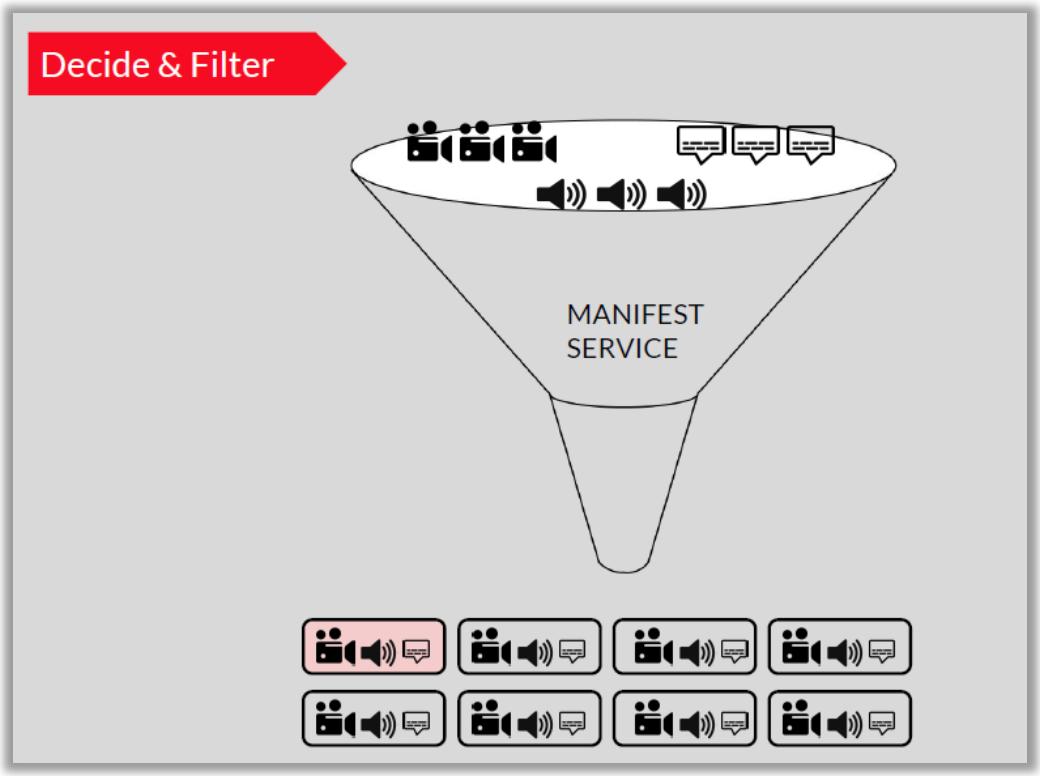
> 2. A user on a client device requests playback of a title from the Netflix application.
>
> 3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics 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.

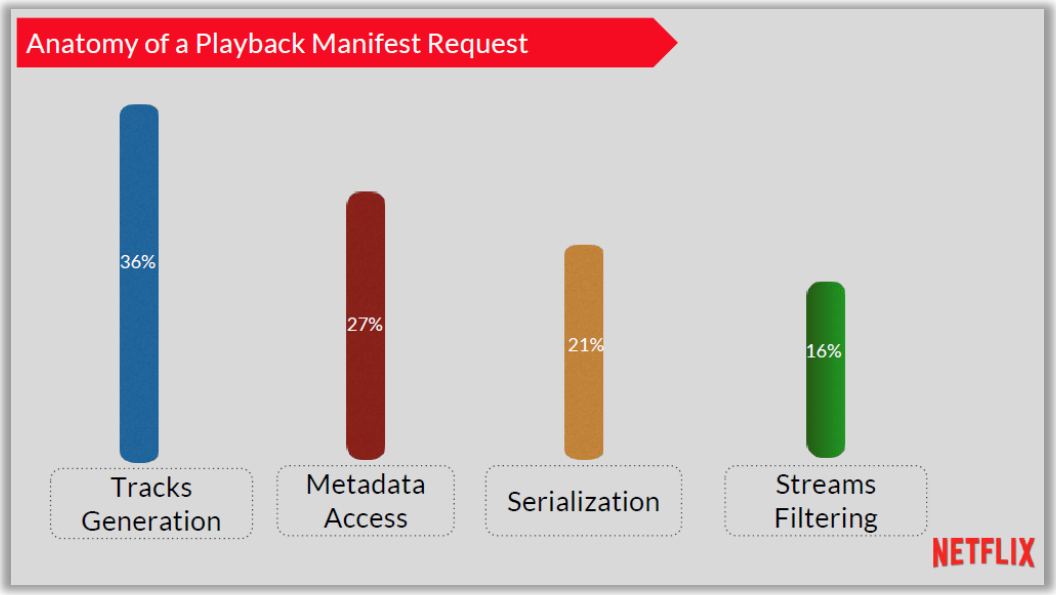
The Netflix playback server has a list of different streams associated with the requested piece of content, for example, using the movieID—streams in different formats for different device capabilities. Multiple resolutions and bitrates exist for the content associated with the movie*ID*.

373. 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.[116]

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

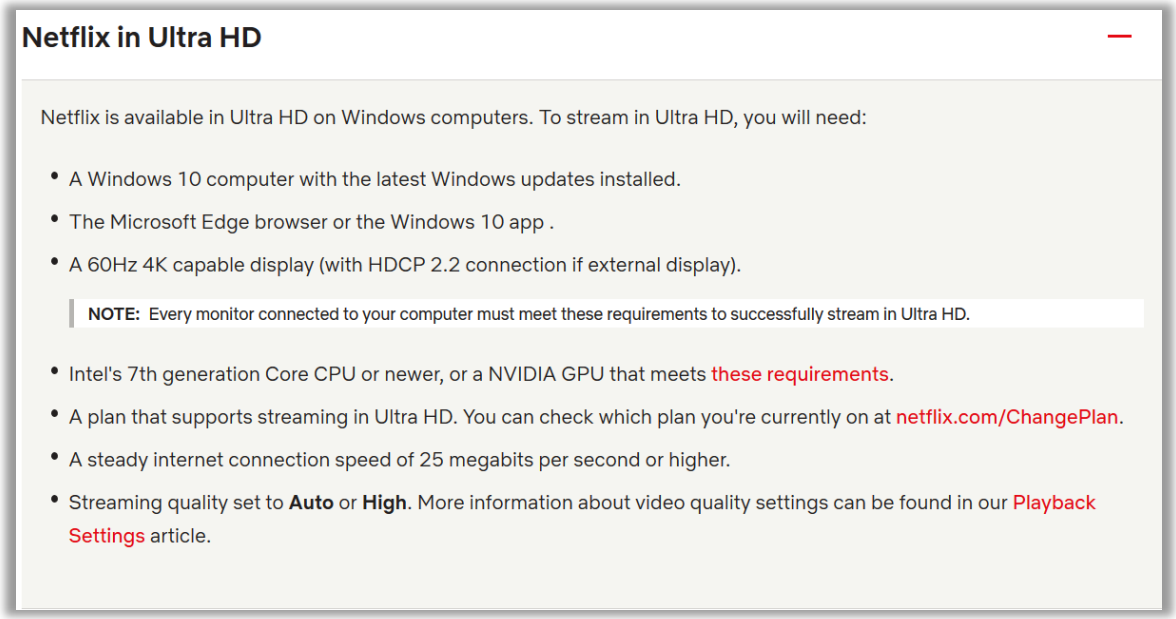
[116] *See* Rangarajan, Suudhan, *Scaling Playback Services*, https://www.slideshare.net/SuudhanRangarajan/scaling-playback-services, at 10, 23.





Netflix generates a different manifest based on the device 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.[117]



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).

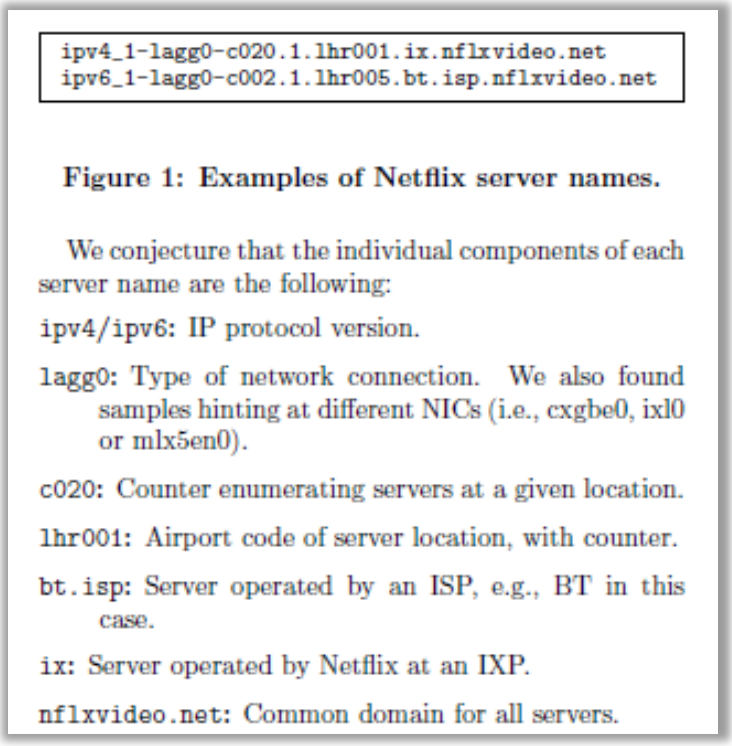
> **NOTE:** Every monitor connected to your computer must meet these requirements to successfully stream in Ultra HD.

- 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.

374. 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.[118]

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

[118] *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.

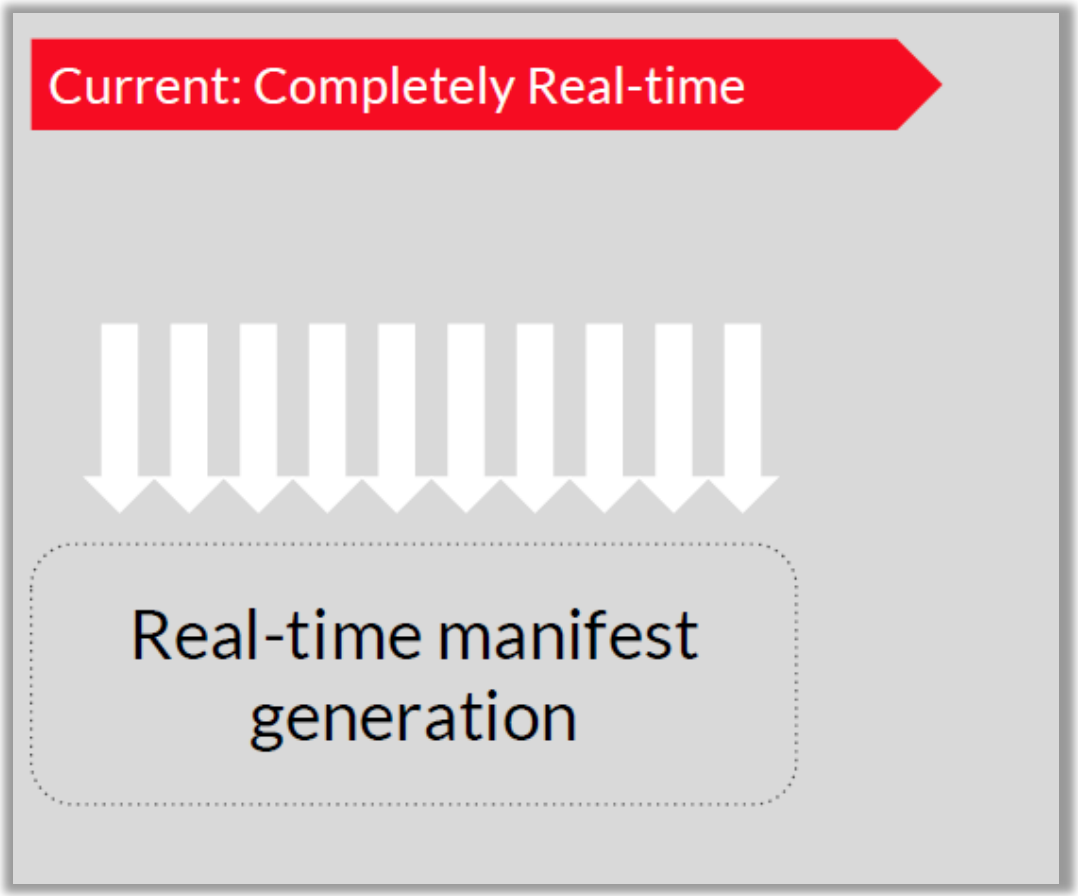
`bt.isp`: Server operated by an ISP, e.g., BT in this case.

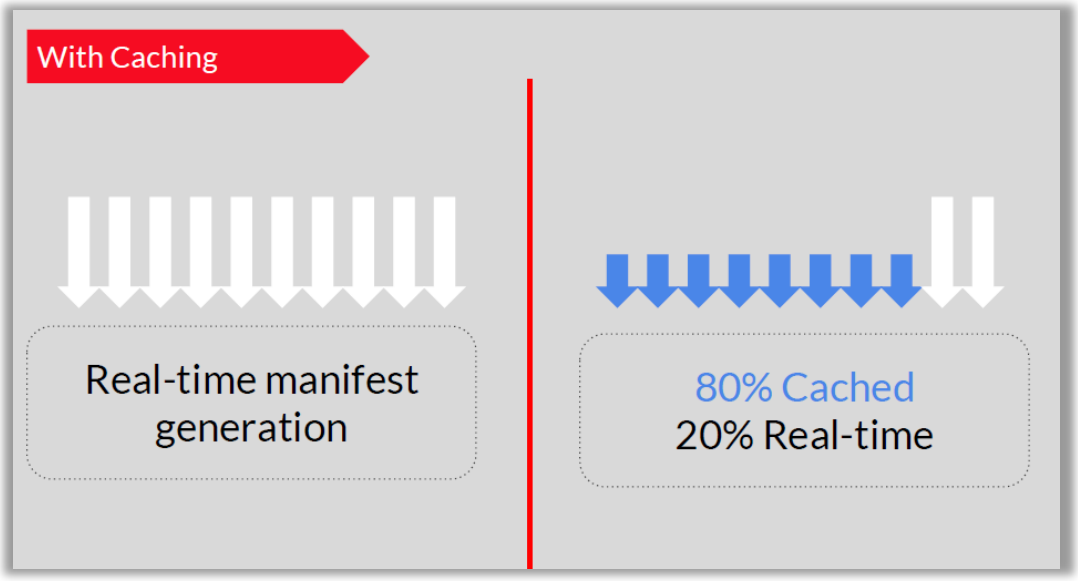
`ix`: Server operated by Netflix at an IXP.

`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:[119]

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





375. 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).

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

377. 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.

378. 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).

379. 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 380-389, in violation of 35 U.S.C. § 271(b).

380. 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.

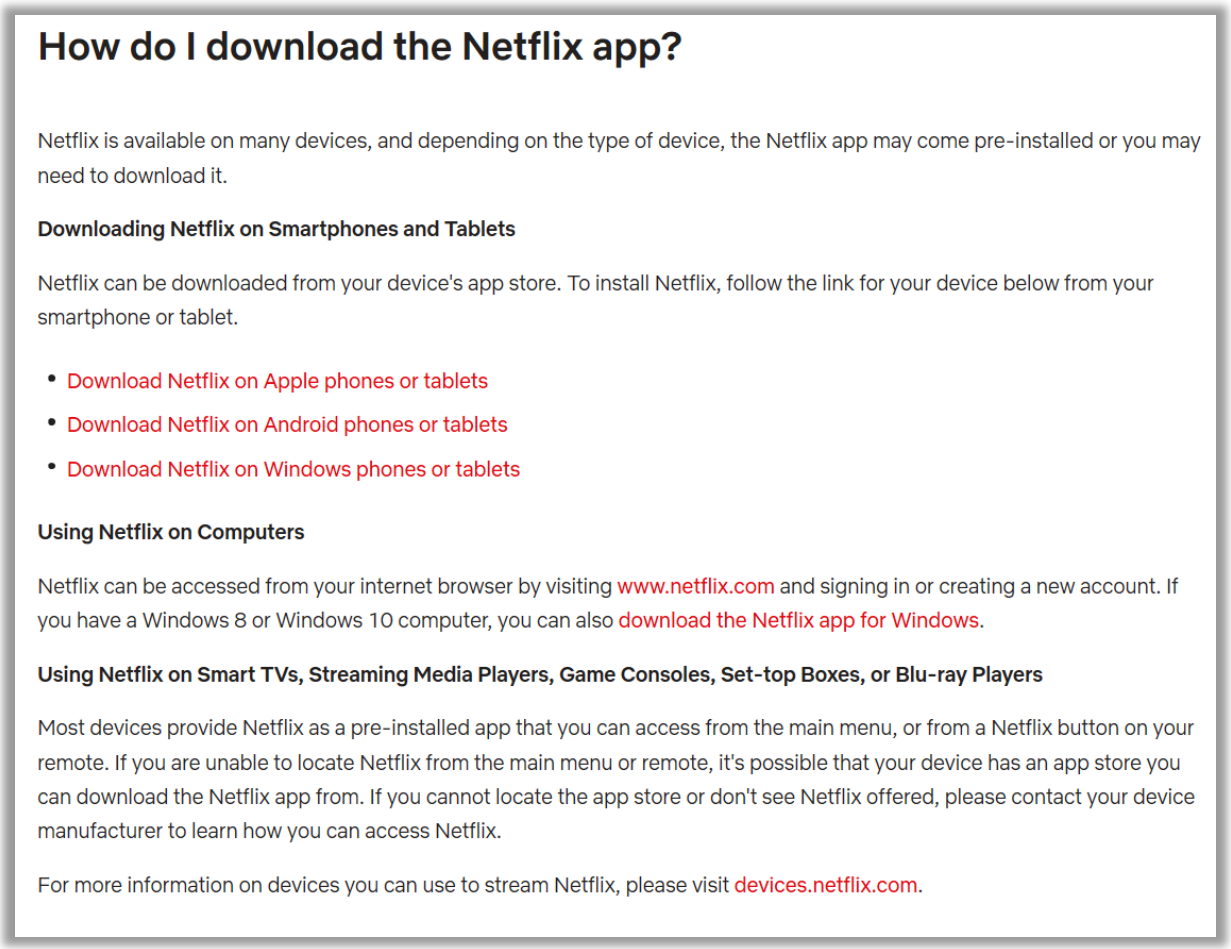
381. At least as of the date of this Complaint, Netflix knows that it provides and specifically intends to provide an application and service to be used with a CE playback device that, when used as intended, meets the limitations of claim 16.

382. 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.

383. 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:[120]

[120] 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.

384. 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.[121]

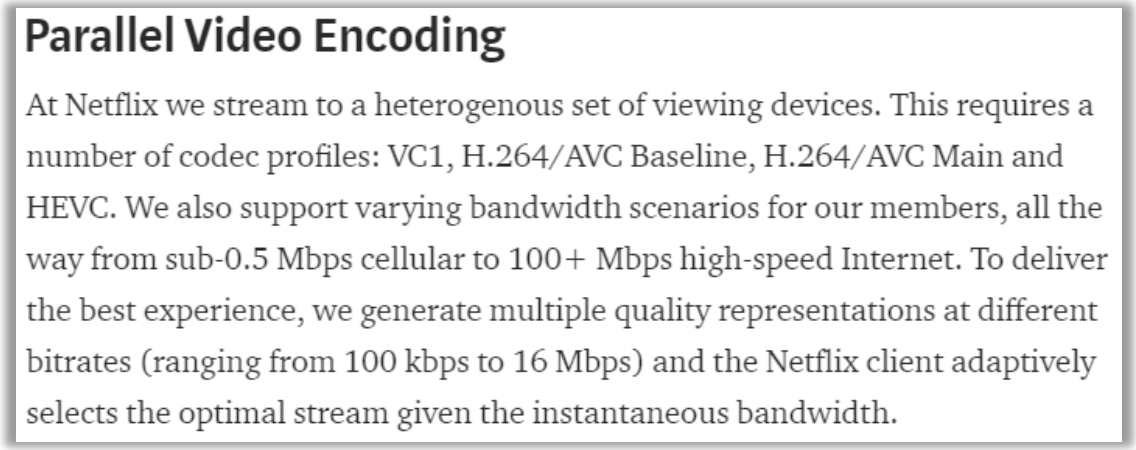
385. The Netflix application "configures the processor to request, using the playback device, a top level index file from a playback server, where the request identifies a piece of content and includes a software version indicating a version number for an adaptive streaming software component implemented 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 movie*ID*. The request further includes a version number for an adaptive streaming software component implemented on the device, for example, the Edge browser.

386. 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.

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

[121] *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.[122]



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.

388. 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.

389. 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.

390. 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.

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

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

391. The allegations of paragraphs 1-390 of this Complaint are incorporated by reference as though fully set forth herein.

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

393. 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”).

394. 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 395-412 below.

395. 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.[123] 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.

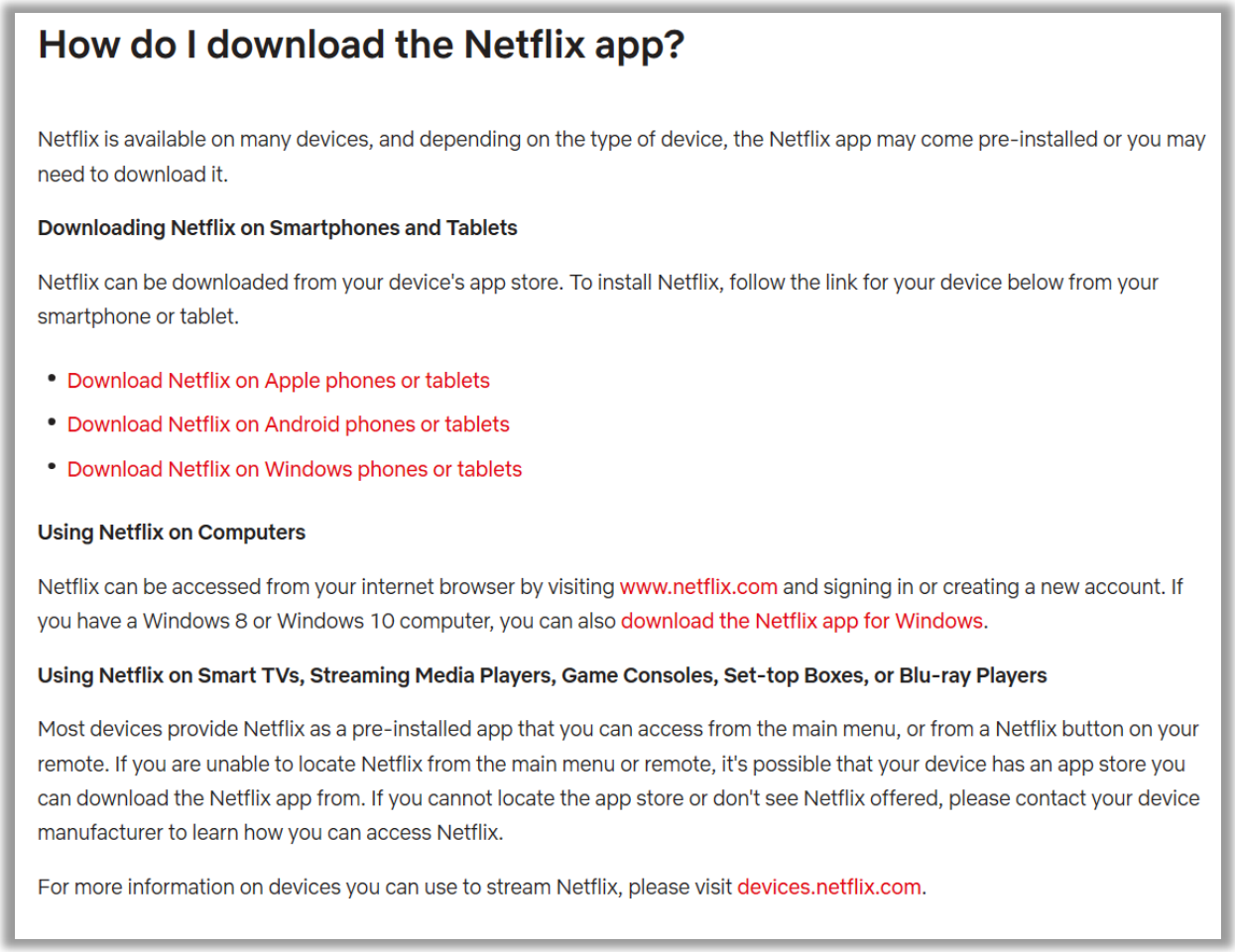
[123] https://medium.com/netflix-techblog/update-on-html5-video-for-netflix-fbb57e7d7ca0.

> 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.

396. 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.

397. 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":[124]

[124] 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.

398. 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.

399. 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* ¶ 398.

400. 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[125] and Microsoft PIFF Specification[126] 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.

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

[126] 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").

401. 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* ¶ 400.

402. 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.

403. 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")[127] or a Sample

[127] Microsoft PIFF specification at 22.

Encryption Box ("senc")[128] 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").

404. 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* ¶ 403.

405. 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

[128] 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.[129]

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.[130] 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."[131] 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.

406. 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.

[129] Microsoft PIFF Specification at 20.

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

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

407. 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.[132]

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:[133]

[132] Microsoft PIFF specification at 17.

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

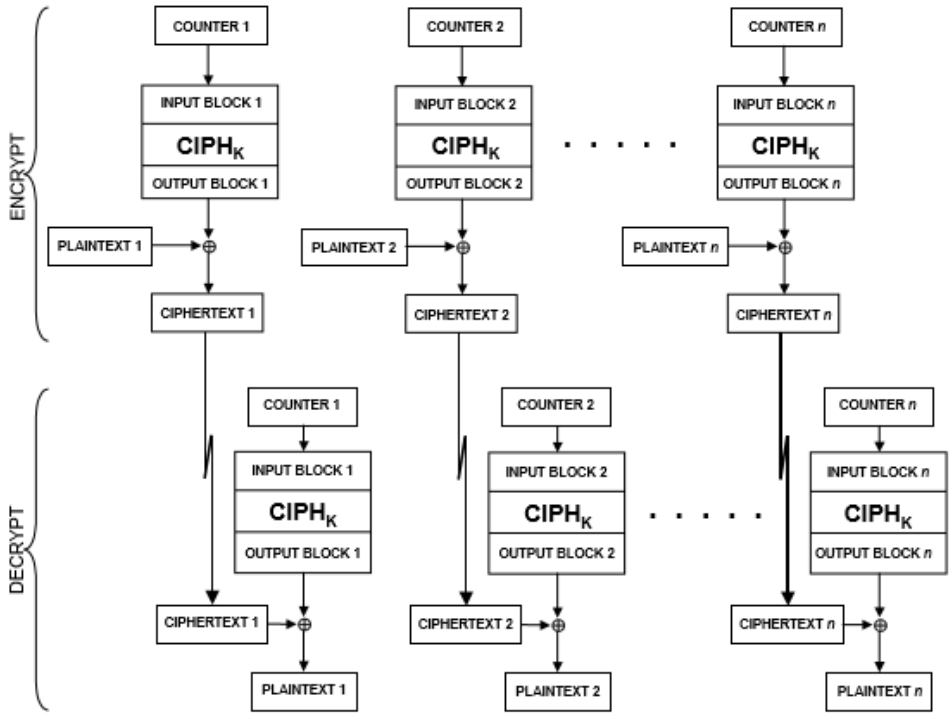


Figure 5: The CTR Mode

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."[134] 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

[134] 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.

408. 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* ¶ 407.

409. 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.

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

411. 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.

412. 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).

413. 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 414-416, in violation of 35 U.S.C. § 271(b).

414. 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 claint 1.

415. 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.

416. 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 395-412.

417. 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

418. The allegations of paragraphs 1-417 of this Complaint are incorporated by reference as though fully set forth herein.

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

420. 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").

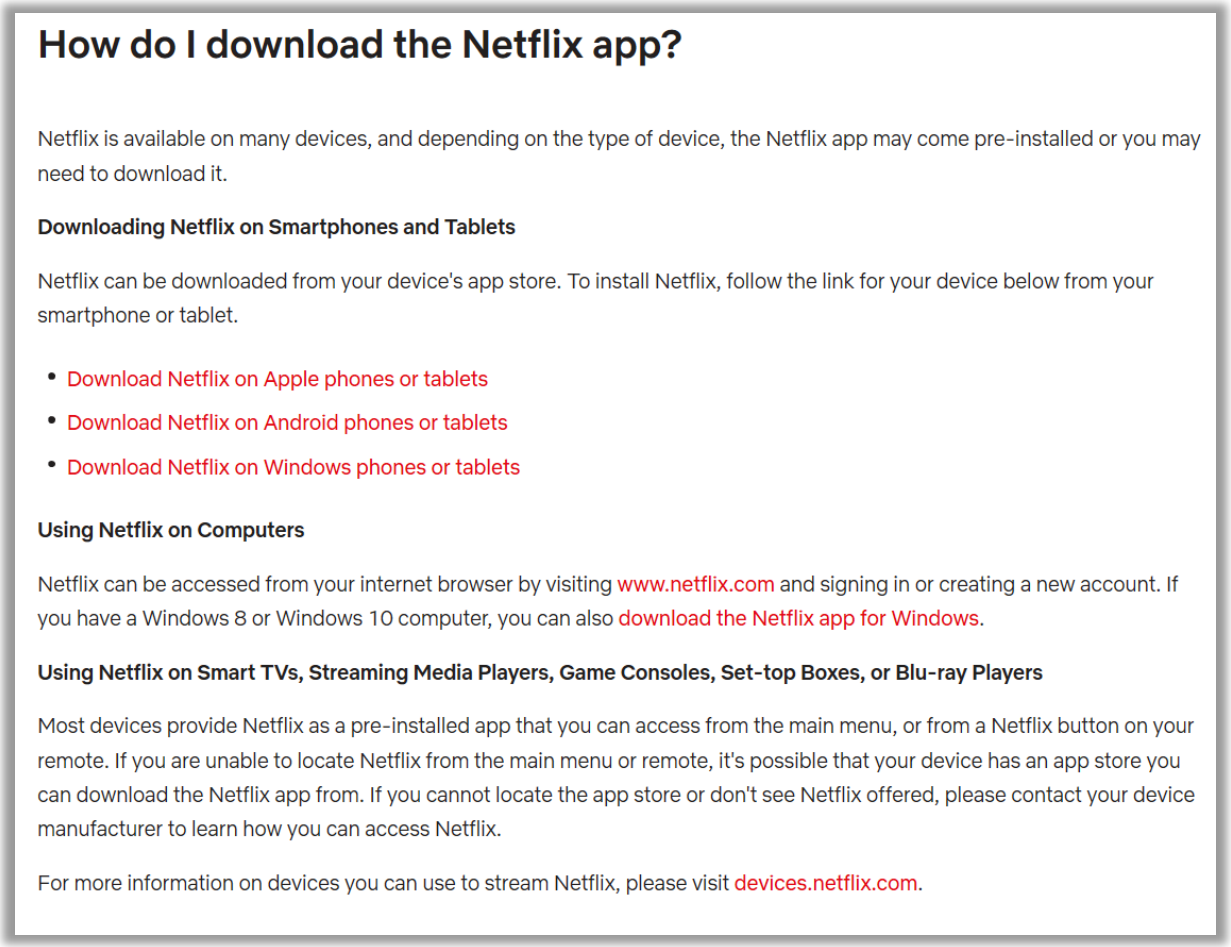
421. Upon information and belief, the Accused '588 Infringing Products directly infringe at least claim 1 of the '588 patent at least in the exemplary manner described in paragraphs 422-436 below.

422. 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.

423. 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.

424. 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":[135]

[135] 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.

425. 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.[136] The Netflix manifest includes the information contained within an MPD file.

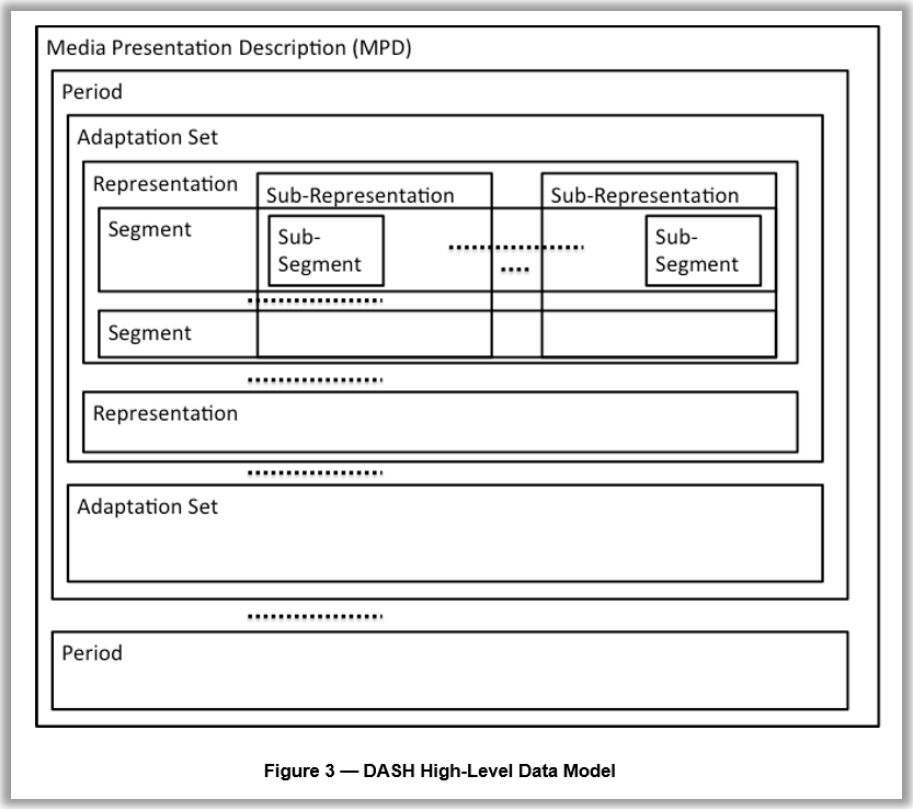
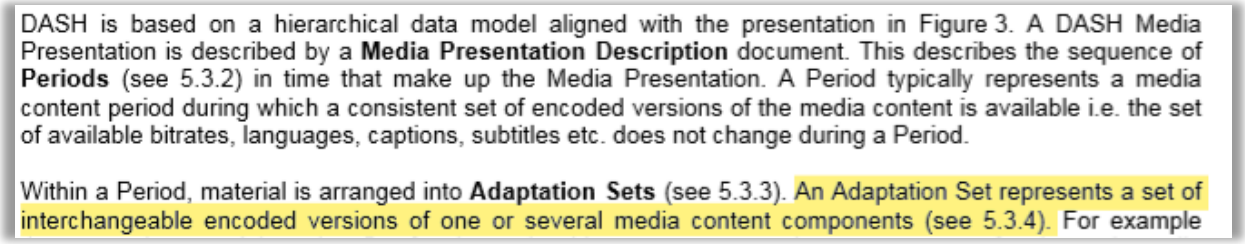


Figure 3 — DASH High-Level Data Model

[136] 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.

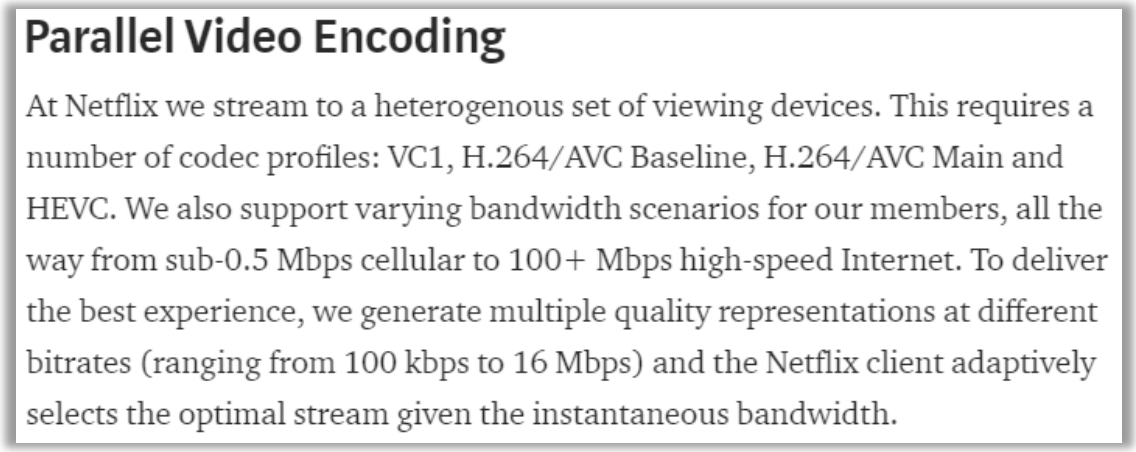
Within a Period, material is arranged into **Adaptation Sets** (see 5.3.3). An Adaptation Set represents a set of interchangeable encoded versions of one or several media content components (see 5.3.4). For example

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.

426. 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.

427. 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"[137] and takes "current network conditions into account" during device playback:[138]



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.

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

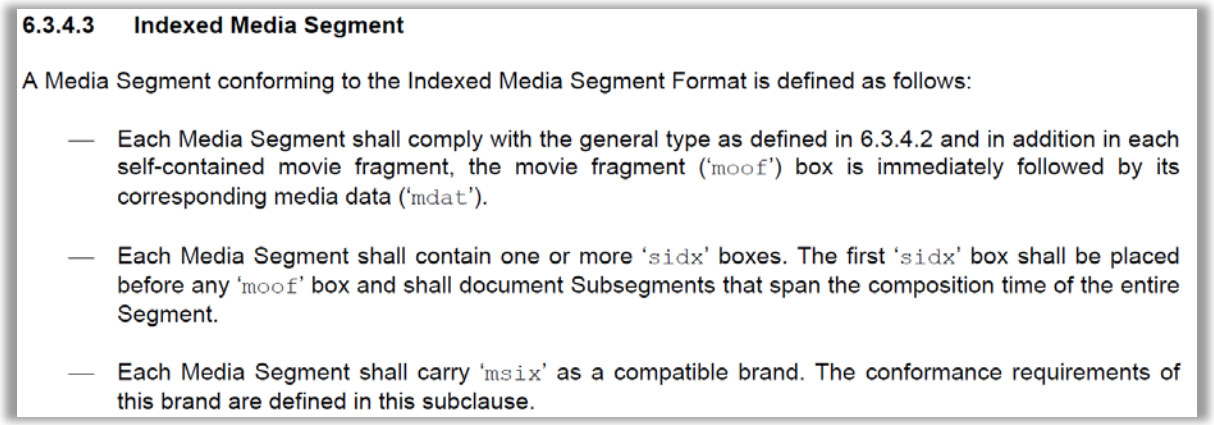
[138] *See Open Connect Overview*, https://openconnect.netflix.com/Open-Connect-Overview.pdf, at 4.

> 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.

428. 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* ¶¶ 425-427.

429. 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:[139]

[139] 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.

430. 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.

431. 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:[140]

[140] 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.

432. 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).

433. 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.

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

435. 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.

436. 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).

437. 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 438-439, in violation of 35 U.S.C. § 271(b).

438. 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.

439. 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 422-436.

440. 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: August 21, 2019

ROBINS KAPLAN LLP

By: */s/ Aaron R. Fahrenkrog*

Aaron R. Fahrenkrog

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*)
CSeidl@RobinsKaplan.com
Aaron R. Fahrenkrog (*pro hac vice*)
AFahrenkrog@RobinsKaplan.com
Bryan J. Mechell (*pro hac vice*)
BMechell@RobinsKaplan.com
William E. Manske (*pro hac vice*)
WManske@RobinsKaplan.com
Shui Li (*pro hac vice*)
SLi@RobinsKaplan.com
Emily J. Tremblay (*pro hac vice*)
ETremblay@RobinsKaplan.com
Mary Pheng (*pro hac vice*)
MPheng@RobinsKaplan.com
Rajin Singh Olson (*pro hac vice*)
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