UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DISNEY ENTERTAINMENT & SPORTS LLC,
Petitioner

v.

ADEIA GUIDES INC.,
Patent Owner.

Case No. IPR2026-00054
U.S. Patent No. 9,762,639

DECLARATION OF DR. BRIAN W. ANTHONY IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 9,762,639

I. INTRODUCTION 1
II. QUALIFICATIONS 1
III. MATERIALS CONSIDERED 4
IV. LEGAL STANDARDS 7
A. Anticipation 8
B. Obviousness 9
C. Claim Construction 12
V. LEVEL OF ORDINARY SKILL IN THE ART 13
VI. SUMMARY OF GROUNDS 14
VII. THE '639 PATENT 15
A. Overview of the '639 Patent 15
B. Prosecution History of the '639 Patent 17
VIII. TECHNOLOGY BACKGROUND 19
A. Streaming media via a data communication network 19
B. Receiving a URL in a request from a client, which includes information indicative of a requested media source 20
C. Determining an identity of the first client 22
D. Determining the device type based on the identity of the client 23
E. Generating a manifest in accordance with a chunking protocol based on the determination of the device type 25
F. Sending the manifest to the client 31
G. Including information in the manifest for streaming an advertisement or other media source during playback of the requested media source 31
H. Including a beaconing URL in the manifest which, when used, is indicative of playback of at least one advertisement 33
I. Including a beaconing URL in the manifest which, when used, is indicative that a certain point has been reached in playback of the requested media source 37
IX. CLAIM CONSTRUCTION 40
A. "Chunking protocol" (Claim 8) 40

Petitioner Disney Ex-1002, 0002

B. "At least one beaconing URL which, when used, is indicative of [playback of at least one advertisement / a certain point has been reached in playback of the requested media source" (Claims 14-15) 40

C. Order of steps 8[b] to 8[e] (Claim 8) 41

D. Order of steps 8[g] to 8[j] (Claim 8) 41

X. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART 41

A. Lewis (Ex-1005) 41

B. Pyle (Ex-1006) 44

C. Barraclough (Ex-1007) 46

D. Jain (Ex-1008) 47

XI. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS 49

A. Grounds 1-2: Claims 8 and 13 are obvious over Lewis alone or Lewis in view of Barraclough. 49

1. A POSITA would have been motivated to combine Lewis's teachings with Barraclough's teachings and would have had a reasonable expectation of success. 49

2. Independent Claim 8 55

a. Element 8[pre]: A method of streaming media via a data communications network, the method comprising: 55

b. Element 8[a]: receiving a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source; 56

c. Element 8[b]: determining an identity of the first client; 59

d. Element 8[c]: determining a first device type based on the identity of the first client; 62

Petitioner Disney Ex-1002, 0003

e. Element 8[d]: based at least in part on the determination of the first device type, generating, with a processor, a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network;.....68

f. Element 8[e]: sending the first manifest to the first client; 75

g. Element 8[f]: receiving the URL in a second request from a second client; 75

h. Element 8[g]: determining an identity of the second client, the identity of the second client different from the identity of the first client;.....78

i. Element 8[h]: determining a second device type based on the identity of the second client, the second device type different from the first device type; 80

j. Element 8[i]: based at least in part on the determination of the second device type, generating, with the processor, a second manifest having information for streaming the one or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and 84

k. Element 8[j]: sending the second manifest to the second client. 89

3. Dependent Claim 13 90

a. Element 13[pre]: The method of streaming media via the data communications network as recited in claim 8, further comprising: 90

Petitioner Disney Ex-1002, 0004

b. Element 13[a]: receiving data regarding the requested media source, wherein the data includes information indicative of at least one point, during playback of the requested media source, at which at least one other media source is to be shown; 90

c. Element 13[b]: receiving information regarding the at least one other media source; and.....93

d. Element 13[c]: including, in either or both of the first manifest and the second manifest, information for streaming one or more segments of the at least one other media source at the at least one point during the playback of the requested media source. 95

B. Grounds 3-4: Claims 14-15 are obvious over Lewis in view of Jain or Lewis-Barraclough in view of Jain. 98

1. A POSITA would have been motivated to combine Lewis's or Lewis-Barraclough's teachings with Jain's teachings and would have had a reasonable expectation of success.....98

2. Dependent Claims 14-15.....106

a. Claim 14: The method of streaming media via the data communications network as recited in claim 13, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of playback of at least one advertisement. 106

b. Claim 15: The method of streaming media via the data communications network as recited in claim 8, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of a certain point has been reached in playback of the requested media source.....109

C. Grounds 5-6: Claim 8 is obvious over Pyle alone or Pyle in view of Barraclough.....112

1. A POSITA would have been motivated to combine Pyle's teachings with Barraclough's teachings and would have

Petitioner Disney Ex-1002, 0005

had a reasonable expectation of success. 112

2. Independent Claim 8 120

a. Element 8[pre]: A method of streaming media via a data communications network, the method comprising: 120

b. Element 8[a]: receiving a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source; 121

c. Element 8[b]: determining an identity of the first client; 124

d. Element 8[c]: determining a first device type based on the identity of the first client; 128

e. Element 8[d]: based at least in part on the determination of the first device type, generating, with a processor, a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network; 131

f. Element 8[e]: sending the first manifest to the first client; 136

g. Element 8[f]: receiving the URL in a second request from a second client; 137

h. Element 8[g]: determining an identity of the second client, the identity of the second client different from the identity of the first client; 139

i. Element 8[h]: determining a second device type based on the identity of the second client, the second device type different from the first device type; 142

Petitioner Disney Ex-1002, 0006

j. Element 8[i]: based at least in part on the determination of the second device type, generating, with the processor, a second manifest having information for streaming the one or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and 144

k. Element 8[j]: sending the second manifest to the second client. 148

D. Grounds 7-8: Claims 13-15 are obvious over Pyle in view of Jain or Pyle-Barraclough in view of Jain.....149

1. A POSITA would have been motivated to combine Pyle's or Pyle-Barraclough's teachings with Jain's teachings and would have had a reasonable expectation of success. 149

2. Dependent Claims 13-15.....155

a. Claim 13.....155

b. Claim 14: The method of streaming media via the data communications network as recited in claim 13, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of playback of at least one advertisement. 161

c. Claim 15: The method of streaming media via the data communications network as recited in claim 8, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of a certain point has been reached in playback of the requested media source.....162

XII. CONCLUSION.....164

Petitioner Disney Ex-1002, 0007

I. INTRODUCTION

1. I have been retained by Disney Entertainment & Sports LLC ("Petitioner"), as an independent expert in this proceeding before the Patent Trial and Appeal Board ("PTAB" or "Board"). I understand that Petitioner is requesting that the Board institute an inter partes review ("IPR") proceeding of U.S. Patent No. 9,762,639 ("the '639 Patent") (Ex 1001), currently assigned to Adeia Guides Inc. ("Patent Owner" or "PO").

2. I am not and have never been an employee of Petitioner. I am being compensated at my usual and customary rate of $500 per hour. No part of my compensation depends on the outcome of this proceeding, and I have no other interest in this proceeding.

3. I have been asked to provide my independent analysis of the '639 Patent in light of the prior art publications cited below. I have also been asked to consider the state of the art and prior art available as of December 22, 2010. Based on the prior art discussed in this declaration, it is my opinion that Claims 8 and 13-15 of the '639 Patent are unpatentable for the reasons provided below.

II. QUALIFICATIONS

4. As indicated in my *curriculum vitae* ("CV"), included as Ex-1003, I am currently a Principal Research Scientist at the Massachusetts Institute of Technology ("MIT"). My CV includes additional information about my professional history and

Petitioner Disney Ex-1002, 0008

contains further details on my experience, publications, patents, and other qualifications to render an expert opinion. Herein, I highlight experiences relevant to the technology of the patent at issue.

5. I earned a Bachelor of Science in Engineering from Carnegie Mellon University in 1994 and a Master's degree in Engineering from MIT in 1998. My thesis topic related to anisotropic wave guides, image processing, timeseries analysis, applied to non-destructive testing. In 2006, I earned my Ph.D. in Engineering from MIT researching in the AI Lab. My research focused on high-performance computation, signal processing, and video-instrumentation system design, video search, human-interfaces for algorithm interaction, and video compression.

6. In 1997, I co-founded Xcitex Inc., a company that specialized in video-acquisition and motion-analysis software. As CTO and first engineer at Xcitex, I designed and built video analytics and streaming products for scientific markets, including GUIs, compression, and key frame technologies and led team to do the same. I served as the Chief Technology Officer and directed and managed product development until 2006. I served as the technical and scientific advisor to eXstream Solutions, one of the earliest video streaming companies. My work there focused on digital media application development, multimedia content and rights management, hardware and network architecture, and digital design and consulting.

Petitioner Disney Ex-1002, 0009

7. I joined MIT in 2006 and have variously served as the Director of the Master of Engineering in Advance Manufacturing and Design Program, Co-Director of the Medical Electronics Device Realization Center. Many of the courses, projects, and papers my students undertake involve technologies relevant to the '639 Patent, for example video and image processing applied to manufacturing system and medical imaging systems.

8. Furthermore, my research focuses on imaging, video processing, human interface systems. I teach or have taught courses in Electrical Engineering, Controls, Optics, Communications Protocols, Signal Processing, Video Processing, Imaging, Smart Systems Design, all pertinent subject matter to the '639 Patent.

9. For decades I have applied principles of video analytics, system design, and compression to domains such as manufacturing imaging, motion capture, and medical imaging. I have extensive experience in video processing, video analytics, GUIs, compression, and key frame detection, and have published widely on video processing and transmission across multiple fields. I was awarded and Emmy Award from the National Academy of Television Arts & Sciences for technical innovation in Broadcast television related to video processing and analysis. I have published approximately 100 papers and have received a number of best paper and distinguished paper awards. I am a co-author of a number of papers that relate to the technology in the '639 Patent, such as:

Petitioner Disney Ex-1002, 0010

- Anthony, Brian W., and Fitriani Chua. "Computationally efficient optimal video comparison for machine monitoring and process control." Journal of Manufacturing Science and Engineering 139, no. 10 (2017): 101007.
- Sun, S.Y., Gilbertson, M. and Anthony, B.W., 2014, September. Probe localization for freehand 3D ultrasound by tracking skin features. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 365-372). Cham: Springer International Publishing.
- Magana-Salgado, Uriel, Praneeth Namburi, Micha Feigin-Almon, Roger Pallares-Lopez, and Brian Anthony. "A comparison of point-tracking algorithms in ultrasound videos from the upper limb." BioMedical Engineering 22, no. 1 (2023): 52
- Kim, David Donghyun, and Brian Anthony. "Design and fabrication of desktop fiber manufacturing kit for education." In Dynamic Systems and Control Conference, vol. 58295, p. V003T31A005. American Society of Mechanical Engineers, 2017.

III. MATERIALS CONSIDERED

10. In forming my opinions, I reviewed the following documents, in addition to others I have cited in my declaration. I also relied on my own knowledge

Petitioner Disney Ex-1002, 0011

of and experience in the fields of video processing and instrumentation, system and user interface design, human machine interfaces, web and communication technology, interactive streaming technologies, user interfaces, software engineering and algorithm design.

Ex. No.	Description
Ex-1001	U.S. Patent No. 9,762,639 ("the '639 Patent")
Ex-1004	Prosecution History of U.S. Patent No. 9,762,639
Ex-1005	U.S. Patent No. 8,677,428 to Lewis ("Lewis")
Ex-1006	U.S. Patent No. 8,782,268 to Pyle ("Pyle")
Ex-1007	U.S. Patent No. 9,047,235 to Barraclough ("Barraclough")
Ex-1008	U.S. Patent Publication No. 2010/268573 to Jain ("Jain")
Ex-1009	INTENTIONALLY LEFT BLANK
Ex-1010	INTENTIONALLY LEFT BLANK
Ex-1011	INTENTIONALLY LEFT BLANK
Ex-1012	INTENTIONALLY LEFT BLANK
Ex-1013	INTENTIONALLY LEFT BLANK
Ex-1014	INTENTIONALLY LEFT BLANK
Ex-1015	INTENTIONALLY LEFT BLANK
Ex-1016	INTENTIONALLY LEFT BLANK
Ex-1017	INTENTIONALLY LEFT BLANK
Ex-1018	INTENTIONALLY LEFT BLANK
Ex-1019	INTENTIONALLY LEFT BLANK

Petitioner Disney Ex-1002, 0012

Ex. No.	Description
Ex-1020	Claim Mapping Table
Ex-1021	U.S. Patent No. 9,143,545 to Barraclough ("Barraclough'545")
Ex-1022	U.S. Patent Publication No. 2009/0037515 to Zapata ("Zapata")
Ex-1023	U.S. Patent No. 8,527,647 to Gopalakrishnan ("Gopalakrishnan")
Ex-1024	U.S. Patent Publication No. 2009/0282162 to Mehrotra ("Mehrotra")
Ex-1025	*Adeia Techs., Inc. v. The Walt Disney Co.*, No. 1:24-cv-01231, Defendants' Supplemental Disclosure of Proposed Claim Constructions (D. Del. Sept. 16, 2025)
Ex-1026	*Adeia Techs., Inc. v. The Walt Disney Co.*, No. 1:24-cv-01231, Plaintiffs' Initial Proposed Claim Constructions (D. Del. Sept. 5, 2025)
Ex-1027	Prosecution History of U.S. Patent No. 9,197,688
Ex-1028	U.S. Patent Publication No. 2012/0222063 to Mao ("Mao")
Ex-1029	U.S. Patent Publication No. 2011/0314130 to Strasman ("Strasman")
Ex-1030	Juan Esteban Hurtado Guzman, Development and Implementation of an Adaptive HTTP Streaming Framework for H.264/MVC Coded Media ("Guzman")
Ex-1031	U.S. Patent No. 9,209,934 to Luby ("Luby")
Ex-1032	U.S. Patent Publication No. 2009/0288076 to Johnson ("Johnson")
Ex-1033	Jit Ghosh, IIS Smooth Streaming -Enhancing Silverlight Video Experiences with Contextual Data, MSDN Magazine, 32 (Mar. 2010) ("MSDN Magazine")
Ex-1034	U.S. Patent No. 8,621,044 to Sood ("Sood")

Petitioner Disney Ex-1002, 0013

Ex. No.	Description
Ex-1035	Delivering Live and On-Demand Smooth Streaming, Microsoft Silverlight, available at https://web.archive.org/web/20100702193600/https://download.microsoft.com/download/4/E/5/4E599FBB-6E34-4A74-B3C5-1391CBOFD55F/Delivering Live and On-Demand Smooth Streaming.pdf (July 2, 2010) ("Smooth Streaming White Paper")
Ex-1036	European Patent No. 2,223,470 to Christian ("Christian")
Ex-1037	U.S. Patent No. 7,584,435 to Bailey ("Bailey")
Ex-1038	U.S. Patent Publication No. 2002/0103875 to Krishnan ("Krishnan")
Ex-1039	Frédéric Gabin et al., *3GPP Mobile Multimedia Streaming Standards*, IEEE Signal Processing Mag., 134–38 (Nov. 2010)
Ex-1040	U.S. Patent No. 6,389,473 to Carmel ("Carmel")
Ex-1041	U.S. Patent No. 7,818,444 to Brueck ("Brueck")
Ex-1042	U.S. Patent Publication No. 2010/0235528 to Bocharov ("Bocharov")
Ex-1043	Ali C. Begen et al., *Watching Video over the Web, Part 2: Applications, Standardization, and Open Issues*, IEEE Internet Comput., 59–62 (May/June 2011)

IV. LEGAL STANDARDS

11. In forming my opinions and considering the subject matter of the '639 Patent and its claims in light of the prior art, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

Petitioner Disney Ex-1002, 0014

12. I understand that earlier publications and patents may act to render a patent unpatentable for one of two reasons: (1) anticipation, and (2) obviousness.

A. Anticipation

13. As explained to me by counsel, the claims of a patent are anticipated by a prior art reference if each and every element of the claim is found either explicitly or inherently in the reference. I understand that inherency requires a showing that the missing descriptive matter in the claim is necessarily present in the allegedly anticipating reference, and that it would have been so recognized by a person of ordinary skill in the art ("POSITA").

14. I understand that when a challenged claim covers several structures, either generically or as alternatives, the claim is deemed anticipated if any of the structures within the scope of the claim is found in the prior art reference.

15. Although anticipation typically involves the analysis of a single prior art reference, I understand that additional references may be used to show that the prior art reference has enabling disclosure (i.e., allows a POSITA to make the invention without undue experimentation), to explain the meaning of a term used in the prior art reference, and/or to show that a characteristic is inherent in the prior art reference.

Petitioner Disney Ex-1002, 0015

B. Obviousness

16. As explained to me by counsel, a claim is invalid as obvious if it would have been obvious to a person of ordinary skill in the art at the time the alleged invention was made. This means that even if all of the elements of the claim cannot be found in a single prior art reference that would anticipate the claim, a person of ordinary skill in the art who was aware of the prior art would have been able to come up with the claimed invention. This may be the case, for example, where the missing element represents only an insubstantial different over the prior art or a reconfiguration of a known system. I understand that in an obviousness determination, the person of ordinary skill in the art is presumed to have knowledge of all material prior art.

17. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

18. I understand that when a product is available, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill in the art can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and a person of ordinary skill in the art would

recognize that it would improve similar devices in the same way, using the technique would have been obvious.

19. I understand that whether a prior art reference renders a patent claim unpatentable as obvious is determine from the perspective of a person of ordinary skill in the art at the time of the alleged invention. I have been told that there is not requirement that the prior art contain an express suggestion to combine known elements to achieve the claimed invention, but a suggestion to combine known elements to achieve the claimed invention may come from the prior art, as filtered through the knowledge of one skilled in the art. In addition, I have been told that the inferences and creative steps a person of ordinary skill in the art would employ are relevant to the determination of obviousness.

20. I understand that one may consider, e.g., whether (1) the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness; (2) there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; (3) the claimed innovation applies a known technique that had been used to improve a similar device or method in a similar way; (4) the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art; (5) the

Petitioner Disney Ex-1002, 0017

invention merely substituted one known element for another known element in order to obtain predictable results; (6) the invention merely applies a known technique to a known device, method, or product to yield predictable results; or (7) known work in one field of endeavor may have prompted variations of it for use in either the same field or a different one based on design incentives or other market forces that would have been predictable to a person of ordinary skill in the art.

21. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of the invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—that is, a connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

22. I am not aware of any allegations by the named inventor of the ’705 Patent or any assignee of the ’639 Patent that any secondary considerations tend to rebut the obviousness of the ’639 Patent.

Petitioner Disney Ex-1002, 0018

23. Additionally, I understand that in considering obviousness, it is important not to use the benefit of hindsight derived from the patent under consideration.

C. Claim Construction

24. I understand that the United States Patent and Trademark Office interprets claim terms in an *inter partes* review proceeding under the same claim construction standard that is used in a United States federal court. I understand that under this standard, the meaning of claim terms is considered from the viewpoint of a POSITA at the time of the alleged invention.

25. I have been informed that claim terms are generally given their ordinary and customary meaning as understood by one of ordinary skill in the art in light of the specification and the prosecution history pertaining to the patent. I understand, however, that claims terms are generally not limited by the embodiments described in the specification.

26. I understand that in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

Petitioner Disney Ex-1002, 0019

V. LEVEL OF ORDINARY SKILL IN THE ART

27. When interpreting a patent, I understand that it is important to identify the relevant art pertaining to that patent, as well as the level of ordinary skill in that art at the time of the claimed invention. The "art" is the field of technology to which the patent is related.

28. I am informed and understand that the POSITA is a hypothetical person who is presumed to know the relevant prior art. I understand that the actual inventor's skill is not determinative of the level of ordinary skill. I further understand that factors that may be considered in determining level of skill include: (i) the types of problems encountered in the art; (ii) prior art solutions to those problems; (iii) the rapidity with which innovations are made; (iv) the sophistication of the technology; and (v) the educational level of active workers in the field. I understand that not all such factors may be present in every case, and one or more of them may predominate.

29. I understand that a POSITA is one who is presumed to be aware of all pertinent art, thinks along conventional wisdom in the art, and is a person of ordinary creativity.

30. A person of ordinary skill in the art at the relevant time ("POSITA") would have had a bachelor's degree in electrical engineering, computer science, computer engineering, or a related field, and two-to-three years of experience in the

Petitioner Disney Ex-1002, 0020

research, design, development, or testing of interactive streaming technologies, computing and software systems, user interfaces, and platforms, with additional education substituting for experience and vice versa.

31. Based on my knowledge, skill, and experience, I have an understanding of the capabilities of a person of ordinary skill in the relevant art. For example, from my industry experience, designing and developing dynamic and reconfigurable user interfaces for commercial software applications used in the control and configuration of electronics, cameras, and data acquisition system and in designing and developing dynamic and reconfigurable user interfaces for the control and configuration of data and video processing and information extraction algorithms, I am familiar with what a POSITA would have known and found predictable in the art. From teaching and supervising my post-graduate students, I also have an understanding of the knowledge that a person with this academic experience possesses. Furthermore, I possessed the knowledge of a POSITA myself at least as of 1997.

VI. SUMMARY OF GROUNDS

32. I understand that Petitioner challenges the validity of Claims 8 and 13-15 of the ’639 Patent. It is my opinion that the following grounds demonstrate that the claims of the ’639 Patent are invalid.

Petitioner Disney Ex-1002, 0021

Ground	Summary
1-2	Claims 8 and 13 are obvious over Lewis (Ex-1005) alone or Lewis in view of Barraclough (Ex-1007)
3-4	Claims 14-15 are obvious over Lewis in view of Jain (Ex-1008) or Lewis-Barraclough in view of Jain
5-6	Claim 8 is obvious over Pyle (Ex-1006) alone or Pyle in view of Barraclough
7-8	Claims 13-15 are obvious over Pyle in view of Jain or Pyle-Barraclough in view of Jain

VII. THE '639 PATENT

A. Overview of the '639 Patent

33. The '639 Patent relates to techniques "for dynamically rewriting the timestamps in streams of media that transition from one source to another" during the "creation of a client manifest" so that the requested segments of media (i.e., "chunks") are "continuous through the transition." Ex-1001, 2:12-19. Specifically, the '639 Patent teaches "chunk-based media streaming in which a media file or live stream is processed into smaller chunks and stored," for example, in a server of the cloud-hosted integrated multi-node pipelining system (CHIMPS) 110, "for serving to a client 145." Ex-1001, 6:31-35.

34. Figure 6[1] below depicts an exemplary method for providing a manifest to a requesting client. Ex-1001, Fig. 6. At step 625, "a request to stream the live

[1] Throughout, emphasis and annotations are added unless otherwise specified.

Petitioner Disney Ex-1002, 0022

media content is received" from the client. Ex-1001, 13:23-25. The '639 Patent teaches that the client request may be "a URL request to the CHIMPS 110, which can provide a manifest file." Ex-1001, 6:35-40. At step 635, "[t]he manifest file is created," which "includes information for streaming one or more segments of live media content via a data communications network" such as a list of Universal Resource Locators (URLs) "that can be used to request chunks of the live media content." Ex-1001, 13:26-31. The '639 Patent further teaches that the chunks are transcoded "into various formats … that can accommodate the various types of networks, clients, and/or end user devices used to stream the live content." Ex-1001, 8:50-56; *see also id.*, 2:44-46 ("determining a type of client associated with the request"), 5:50-53 (system provides media to many clients on many end user devices). At step 645, "the manifest file is sent" to the client. Ex-1001, 13:42-43. The independent claim at issue below, Claim 8, recites these steps for requests from both a first client and second client having different device types, which results in the generation of a first manifest file and a second manifest file according to a first chunking protocol and a second chunking protocol, respectively. Ex-1001, Cl. 8.

Petitioner Disney Ex-1002, 0023

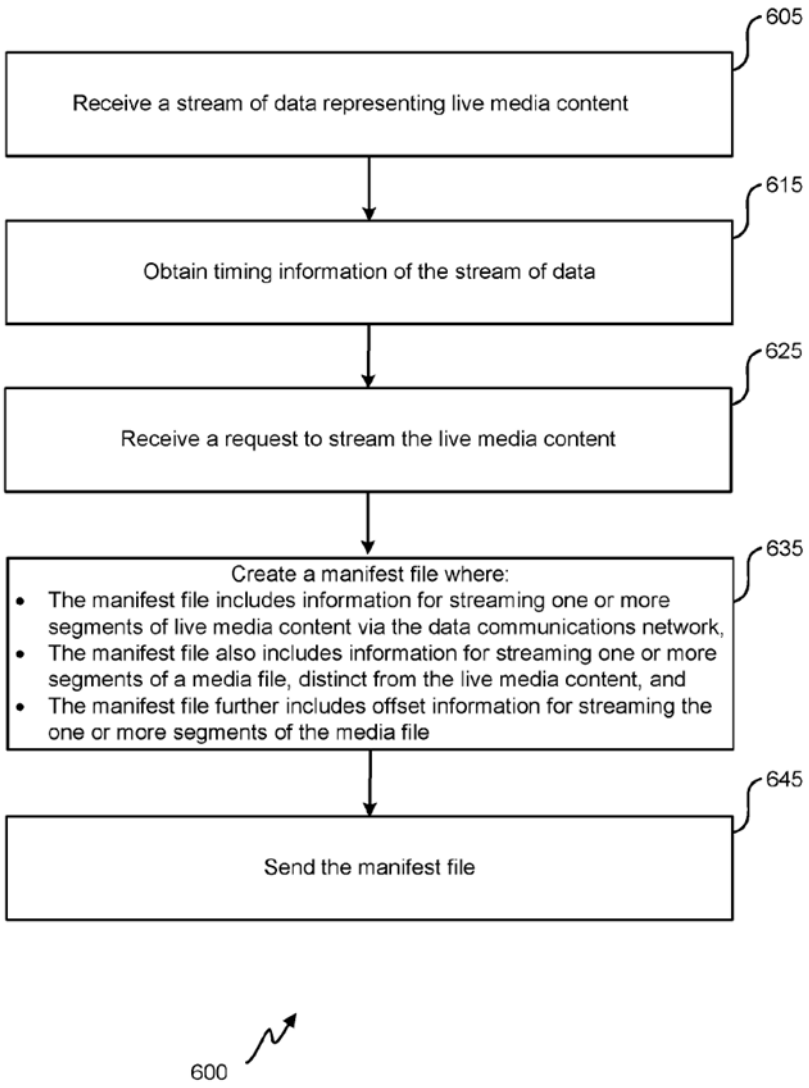


FIG. 6

B. Prosecution History of the '639 Patent

35. The '639 Patent was filed on October 28, 2016, and claims priority through a chain of 10 continuation applications (many now issued patents), the

Petitioner Disney Ex-1002, 0024

earliest of which was filed December 22, 2010.[2] The '639 Patent also claims priority to provisional application no. 61/884,709, filed on September 30, 2013.

36. I understand from Counsel that the prior art combinations and arguments in this Petition were not before the PTO during prosecution of the '639 Patent. Ex-1004. However, Applicant submitted a nine-page IDS, which cited 232 references including the publication version (US2012/0047542) of the Lewis reference (Ex-1005) described in Grounds 1-4 herein. Ex-1004, 86-96. Applicant also filed a terminal disclaimer against the parent '688 Patent to obviate a double patenting rejection. Ex-1004, 1201-202. The Examiner did not issue any prior art rejections. Ex-1004, 1208-16.

37. I understand from Counsel that during prosecution of the parent '688 Patent, the then-pending claims were rejected under 35 U.S.C. §103 as obvious over Biderman (US2010/0169303) in view of the publication version of Lewis. Ex-1027, 87-94. Applicant argued that the Biderman-Lewis combination did not teach that the manifest file includes "offset information based on the timing information of the stream of data" and that Lewis does not teach "providing, in the manifest file, an indication of a discontinuity in the stream of the live media content … based on the determination of the type of client," "receiving a response to the request, the

[2] One of those continuation applications is U.S. Patent No. 9,197,688 ("the '688 Patent"), discussed below.

Petitioner Disney Ex-1002, 0025

response including timing information," "wherein sending the request for the timing information is based on the receipt of the indication of the period of time in the live media content designated for advertising," and "wherein the timing information of the stream of data includes one or more timestamps associated with one or more segments of the stream of data." Ex-1027, 108-119. After the Examiner issued a second rejection over the Biderman-Lewis combination (Ex-1027, 123-133), Applicant amended the claims to recite that the manifest file includes offset information "indicative of a value by which at least one timestamp of the one or more segments of the media file is to be offset." Ex-1027, 170-178.

38. The limitations that Applicant argued and/or added to overcome Lewis during prosecution of the parent '688 Patent are not at issue in the '639 Patent. In my opinion, the Examiner materially erred by not rejecting Claims 8 and 13-15 of the '639 Patent over Lewis during prosecution because, as demonstrated in Grounds 1-4 below, Lewis in combination with Barraclough and/or Jain renders obvious each limitation of Claims 8 and 13-15.

VIII. TECHNOLOGY BACKGROUND

A. Streaming media via a data communication network

39. Streaming media via a data communication network was a known concept by 2010. For example, U.S. Patent No. 8,527,647 to Gopalakrishnan (Ex-1023, "Gopalakrishnan" discloses "[a] technique for controlling the streaming of

Petitioner Disney Ex-1002, 0026

content through a network." Ex-1023, Abstract. Similarly, U.S. Patent No. 9,143,545 to Barraclough (Ex-1024, "Barraclough'545") contains teachings of "[m]edia content [that] is provided for remote playback over a packet-based network." Ex-1023, Abstract; *see also* Ex-1028 (Mao) at Figs. 1, 3.

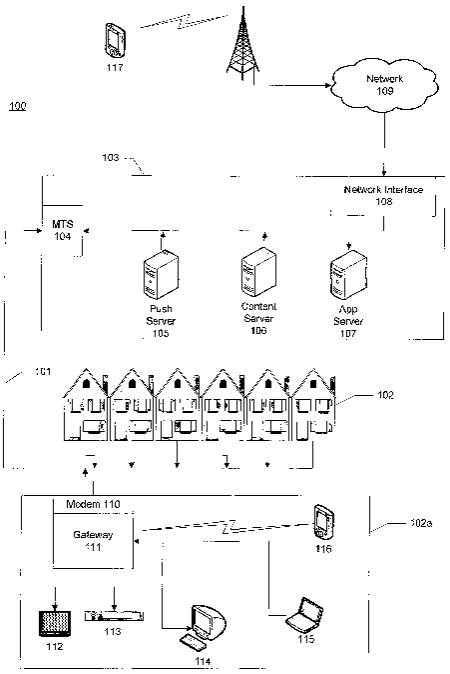


FIG. 1

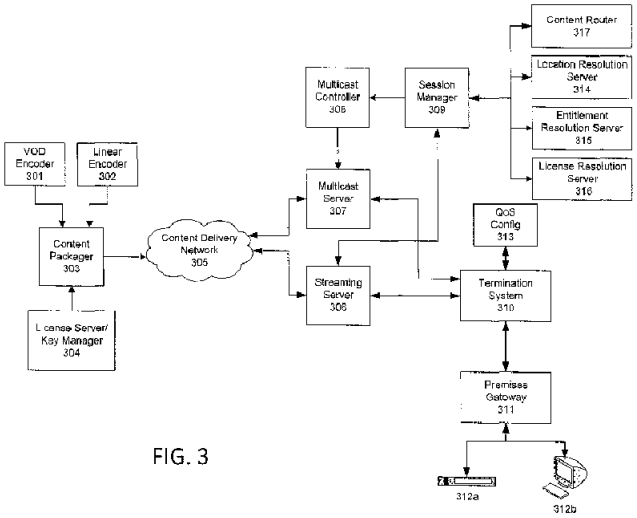


FIG. 3

B. Receiving a URL in a request from a client, which includes information indicative of a requested media source

40. Receiving a request from a client, which includes information indicative of a requested media source was a known concept by 2010. For example, Guzman (Ex-1030, "Guzman") teaches "Requesting a media resource. The first step is requesting a media resource. Media resources are identified and located on the network by uniform resource locators (URLs). A HTTP client or a media player

Petitioner Disney Ex-1002, 0027

initiates the HTTP request." Ex-1030, 47. Similarly, Gopalakrishnan discloses that "[u]ser devices 184 can request content from a website provider, the website provider representing an example of a content provider." Ex-1023, 7:26-28.

41. It was also known by 2010 that a request to stream media content could be in the form of a URL. For example, U.S. Patent Publication No. 2011/0314130 to Strasman (Ex-1029, "Strasman") teaches that a "manifest file include[sic] retrieval information (for example—URL, pointers to addresses in the URL) required for retrieving the media file to be streamed to the client." Ex-1029 ¶77. Similarly, Guzman teaches that a "[c]lient requests a media resource identified by an URL." Ex-1030 at 46; *see also* Ex-1021 (Barraclough'545), 12:5-8 ("The request is made, for example, by clicking on and/or entering an appropriate URL (Uniform Resource Locator) for a web interface hosted by the browser gateway 220."); Ex-1039 (Gabin), 135 ("each representation describes a series of segments by HTTP URLs").

42. Additionally, it was known by 2010 that requests to stream a media source could be received from a plurality of clients. For example, Strasman teaches that "[s]tage 154 includes receiving a plurality of manifest file requests from a plurality of clients that are hosted on user devices, wherein each manifest file request received from a client includes a request to receive retrieval information required for a retrieval of a manifest file associated with a requested media file that is requested

Petitioner Disney Ex-1002, 0028

by the client." Ex-1029 ¶107. Similarly, U.S. Patent No. 9,209,934 (Ex-1031, "Luby") discloses that "a block-request streaming system (BRSS) comprises one or more clients that make requests to one or more content servers (for example, HTTP servers, FTP Servers, etc.)." Ex-1031, 13:56-59.

C. Determining an identity of the first client

43. Determining an identity of the client making the request was a known concept by 2010. For example, Barraclough'545 teaches that a "browser gateway 220 authenticates the remote user at the remote-user device 210 using one or more of a variety of approaches. In some applications, the browser gateway 220 presents the web-based interface 221 with fields to facilitate the entry of authentication information at the remote-user device 210 (e.g., the user enters a user name and password via the web-based interface)." Ex-1021, 12:15-21. Similarly, U.S. Patent Publication No. 2009/0282162 to Mehrotra (Ex-1021, "Mehrotra") teaches that a "web activity monitoring application collects web activities data from the user device 130 (e.g., site ID, video page URL, video file URL, start and end timestamp and any additional metadata about videosite information, URL information, time, etc.) and additionally assigns a unique ID, such as a globally unique identifier or "GUID," to each device. Ex-1021 ¶29.

Petitioner Disney Ex-1002, 0029

D. Determining the device type based on the identity of the client

44. Determining the device type based on the identity of the client was a known concept by 2010. For example, U.S. Patent Publication No. 2009/0037515 to Zapata (Ex-1022, "Zapata") teaches a "routine 'collectInformation'" method for data transfer where the system collects sufficient "user and user-device information … to identify the" current user by, for example, querying "a user or application running on a user's PC for information needed to identify that the user or user application is the user for which deployment is requested." Ex-1022 ¶192. Queries may involve, for example, "request and receipt of passwords, user identifiers, and other information that may be matched to previously acquired information stored by the server on which the routine 'collectInformation' runs, and may alternatively involve a wide variety of other user-identification techniques, including biometric analysis of user-supplied data, or alternative methods of identifying the user." *Id.* ¶192. Zapata teaches that the routine "collectInformation" then collects "additional user-related information, including the PC network address for the user's PC, … characteristics and capabilities of the user's PC or other device, and other such information." *Id.* ¶193.

45. Similarly, Barraclough'545 teaches that "[i]n some embodiments, the remote user (registered or guest) provides information to the host server characterizing a class or classification of the remote-user device, and this data is

Petitioner Disney Ex-1002, 0030

stored together with configuration information specifying a data configuration format to use in sending data to the remote user device. In some applications, this information is provided directly, such as by the remote user making selections from a hosted web page to indicate a desired data type. In other applications, this information is provided indirectly, where the host server and/or PC (via the agent software) determines the type of remote device at which the remote user will be accessing delivered data." Ex-1021, 6:11-22; *see also* Ex-1024 (Mehrotra) ¶2 ("For bitrate-switching streaming solutions that rely on server-side awareness of the needs of each specific client, additional server-side resources may be required. For example, locating bitrate selection decision making at the server may require additional server-side software and/or hardware resources. In addition, servers may require information about clients, such as client capabilities (e.g., hard-ware and software), and network bandwidth information").

46. Additionally, it was known by 2010 that there could be a plurality of different device types. For example, Zapata also teaches that the routine "collectInformation" uses "additional queries or access to previously stored information" to determine "the target devices for the currently considered user to which the service application is to be deployed." Ex-1022 ¶193. The collected information "for the user and for the user's target devices is stored in a user/target list that directs creation and deployment of [dynamically created service applications

Petitioner Disney Ex-1002, 0031

(DCAs)]" in later steps. *Id.*; *see also id.* ¶199 (describing the "routine 'determineDeviceType'" process); Ex-1023 (Gopalakrishnan), 7:28-30 ("[T]he user devices can be individual endpoint devices, such as smartphones, laptop computers, pad computer, and the like."), 14:14-19 ("The mobile stations 202 can be handheld wireless devices, such as cell phones, mobile phones, smartphones, Personal Digital Assistants (PDA), pad computers, handheld gaming devices etc, that can wirelessly communicate using radio frequency (RF) communication signals.").

E. Generating a manifest in accordance with a chunking protocol based on the determination of the device type

47. Generating a manifest based on the determination of the device type was a known concept by 2010. For example, Barraclough'545 discloses that "[o]nce classified, information characterizing the classification for each remote device is stored (e.g., at the data source) and used in selectively configuring or otherwise presenting data to the remote device." Ex-1021, 4:31-35. Similarly, U.S. Patent Publication No. 2009/0288076 (Ex-1032, "Johnson") teaches that "[t]he composite manifest file includes various information about the updates obtained from the databases on the content servers …, destination locations for the updates on the destination media player, and, optionally, additional information obtained from the databases on the content servers …." Ex-1032 ¶57. Guzman also teaches that its "framework[,] as other major commercial solutions uses a manifest file. The manifest file is a structured collection of data that is accessible to the HTTP-

Petitioner Disney Ex-1002, 0032

Streaming Client." Ex-1030 at 41, 33 ("Manifest file is available in different formats E.g. 3GPPs' adaptive HTTP streaming (AHS) defines a Media Presentation Description (MPD), Adobes' Dynamic HTTP Streaming, uses the proprietary FMF manifest, Apple HTTP live streaming uses a M3U playlist file, Microsoft's' Smooth streaming also use a server manifest file (i.e. SMIL document). The manifest file introduced in this framework follows the guidelines of Adaptive HTTP streaming. 3GPP's specification."); *see also* Ex-1023 (Gopalakrishnan), 27:46-48 ("The manifest file 500 can be edited in various different ways to control the set of available versions from which the client can select.").

48. Additionally, it was known by 2010 that the manifest would have included information for streaming one or more segments of the requested media source via the data communication network. For example, Gopalakrishnan teaches that "[t]he manifest file specifies the template URLs for segments of the video streams at each bit rate … In general, the manifest file is a file that includes URIs (e.g., URLs) to identify the location of corresponding content. Once the manifest file is received, the client requests appropriate segments (based on the playback location in the video sequence) of the appropriate bit rate content (based on the rate at which the client is receiving the video data) in a sequence as the playout progresses." Ex-1023, 23:17-27. Similarly, Guzman discloses "[t]he HTTP-Streaming client requests and downloads the manifest file with media data information to present the

Petitioner Disney Ex-1002, 0033

streaming service to the user." Ex-1030 at 41. Strasman also teaches that "[t]he manifest file include retrieval information (for example—URL, pointers to addresses in the URL) required for retrieving the media file to be streamed to the client." Ex-1029 ¶77.

49. It was further known by 2010 that the content of the manifest would differ depending on the device type such that each different device type would have different content in the manifest for the same requested media source. For example, Johnson teaches "[t]he ability to dynamically form a manifest file that describes the specific update components requested for a specific media player 140." Ex-1032 ¶38; *see also* Ex-1030 (Guzman), 33 ("Manifest file is available in different formats E.g. 3GPPs' adaptive HTTP streaming (AHS) defines a Media Presentation Description (MPD), Adobes' Dynamic HTTP Streaming, uses the proprietary FMF manifest, Apple HTTP live streaming uses a M3U playlist file, Microsoft's' Smooth streaming also use a server manifest file (i.e. SMIL document). The manifest file introduced in this framework follows the guidelines of Adaptive HTTP streaming. 3GPP's specification.").

50. It was also known by 2010 that the manifest would be generated using a processor. For example, Strasman teaches a "system … wherein the processing module is arranged to generate a new modified manifest file for each requested media file, wherein the new modified manifest file comprises retrieval information

Petitioner Disney Ex-1002, 0034

previously removed from the modified manifest file and is associated with the selected version of the part of the requested media file associated with the modified manifest file." Ex-1029, Cl. 19. Johnson also teaches that "[t]he update server application and/or the update generator may have the following capabilities: … The ability to dynamically form a manifest file that describes the specific update components requested for a specific media player." Ex-1032 ¶36.

51. Moreover, it was known by 2010 that the manifest would be generated in accordance with a chunking protocol. For example, Apple's HTTP Live Streaming ("HLS") chunking protocol and Microsoft's Smooth Streaming Transport ("SST") chunking protocol were well known. Ex-1043 (Begen), 59-60 ("[T]he provider can choose to reuse its data network services, such as HTTP servers and caches, and employ pull-based adaptive streaming methods such as Apple's HTTP Live Streaming."), 62-63 (Microsoft's SST is a "complete specification[] for an adaptive streaming container and streaming protocol," which "use[s] the standard MP4 fragmented file structure. … Apple has similarly offered its HTTP Live Streaming, both on the Web and in an IETF draft. Both the Microsoft and Apple specifications reflect the current adaptive streaming products that the two companies offer." (citing *Smooth Streaming Transport Protocol*, Microsoft specification, Sept. 2009; http://go.microsoft.com/?linkid=9682896)).

Petitioner Disney Ex-1002, 0035

52. Moreover, Gopalakrishnan discloses a technique that involves "changing the rate at which content chunks are sent from an intermediate network device in order to cause a client to request content chunks that are encoded at a different bit rate." Ex-1023, Abstract. Moreover, Gopalakrishnan further teaches that "[p]layback using an HTTP adaptive bit rate streaming protocol begins with the client requesting a manifest file, whose URL is typically embedded in the parent web page. The manifest file specifies the template URLs for segments of the video streams at each bit rate and may be referred to in some HTTP adaptive bit rate streaming protocols as a 'playlist file' or a 'variant playlist file.' In general, the manifest file is a file that includes URIs (e.g., URLs) to identify the location of corresponding content. Once the manifest file is received, the client requests appropriate segments (based on the playback location in the video sequence) of the appropriate bit rate content (based on the rate at which the client is receiving the video data) in a sequence as the playout progresses." Ex-1023, 23:15-27. Similarly, Jit Ghosh, IIS Smooth Streaming-Enhancing Silverlight Video Experiences with Contextual Data, MSDN Magazine, 32 (Mar. 2010) (Ex-1033, "MSDN Magazine") teaches "[f]or a given video stream in the client manifest, the video track is broken into many fragments of two-second duration, and each c element in the manifest represents metadata for the fragment." Ex-1033 at 35; *see also* Ex-1029 (Strasman) ¶9 ("Algorithms implemented in the client decide which bit rate chunk to request

Petitioner Disney Ex-1002, 0036

from the server. … Occasionally the client may request a chunk with a bit rate which is higher than the last received chunk."); Ex-1034 (Sood), 3:40-43 ("The server may provide chunks in multiple encodings so that the client can, for example, switch quickly to chunks of a different bit rate or playback speed."), 4:7-8 (for example, "this chunk represents offset 1:10 in a 10 minute video clip"); Ex-1039 (Gabin), 134-137 (describing HTTP streaming protocol); Ex-1040 (Carmel), 2:4-6 ("The data stream is divided into a sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed."), 2:22-28 ("Preferably each segment or slice is contained in a separate, respective file. Alternatively, the segments or slices may all be contained in a single indexed file, which is streamed to the client in a series of packets, each covering a range of one or more indices. HTTP version 1 supports this sort of file streaming."), 7:23-25 ("Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T_1, T_2, T_3, etc."); Ex-1041 (Brueck), 3:36-39 ("The method also includes segmenting the media content into a plurality of streamlets, each streamlet comprising a portion of the media content having a predetermined length of time."), 7:62-8:5; Ex-1042 (Bocharov), Fig. 5, ¶14 ("A smooth streaming system is described herein that provides a stateless protocol between the client and server in which the server embeds incremental information in media fragments (i.e., chunks) that eliminates the usage of a typical control channel.").

Petitioner Disney Ex-1002, 0037

F. Sending the manifest to the client

53. Sending the manifest to the client was a known concept by 2010. For example, Gopalakrishnan discloses, "[f]or example, a manifest file is received at an intermediate network device in response to a request from a client for a content element, the manifest file is then edited at the intermediate network device, and then the edited manifest file is sent to the client." Ex-1023, 2:15-19. Strasman also teaches "[g]enerating a modified manifest file for each requested media file by removing retrieval information associated with all non-selected versions of requested media files; wherein each modified manifest file comprises retrieval information required only for a retrieval of a selected version of a requested media file associated with the modified manifest file; and sending to the clients the modified manifest files." Ex-1029 ¶14.

G. Including information in the manifest for streaming an advertisement or other media source during playback of the requested media source

54. Receiving data including information indicative of at least one point during playback of the requested media source at which at least one other media source, such as an advertisement, is to be shown was a known concept by 2010. For example, U.S. Patent No. 8,621,044 (Ex-1034, "Sood") teaches that "[i]n some embodiments, the adaptive streaming system supports inline advertising and other non-audiovisual data (e.g., captions, comments, and so forth) … An event

Petitioner Disney Ex-1002, 0038

coordinator may press a button during production when it is time for a commercial, causing the system to insert an advertising marker in the media stream metadata. When the client receives the advertising marker, the client may request and receive chunks associated with a previously identified advertisement. For example, the server may provide a list of potential advertisements in an initial manifest." Ex-1034, 10:28-39; *see also* Ex-1028 (Mao) ¶22 ("Another application server may be responsible for formatting and inserting advertisements in a video stream and/or content item being transmitted to the premises 102.").

55. Including in the manifest information for streaming one or more segments of the at least one other media source, such as an advertisement, at the at least one point during playback of the requested media source was also a known concept by 2010. For example, Delivering Live and On-Demand Smooth Streaming, Microsoft Silverlight, available at https://web.archive.org/web/20100702193600/https://download.microsoft.com/download/4/E/5/4E599FBB-6E34-4A74-B3C5-1391CBOFD55F/Delivering Live and On-Demand Smooth Streaming.pdf (July 2, 2010) (Ex-1035, "Smooth Streaming White Paper") discloses "[a] program's manifest file can also support an advertising insert track, which contains information about live ads or ad pods—all tightly synchronized to the program's video and audio content. For example, an ad campaign created inside DoubleClick's DART® system could be inserted inside a

Petitioner Disney Ex-1002, 0039

manifest file at a specific time, for playback inside a media player. Using a live ad insertion application helps ensure that an advertisement will always be shown at the correct time within the program." Ex-1035, 23; *see also* Ex-1028 (Mao) ¶22.

H. Including a beaconing URL in the manifest which, when used, is indicative of playback of at least one advertisement

56. Including a beaconing URL in the manifest which, when used, is indicative of playback of at least one advertisement was a known concept by 2010. For example, Sood teaches that "[f]or a live event, it may be unknown at the start of the event when commercial breaks will occur. An event coordinator may press a button during production when it is time for a commercial, causing the system to inset an advertising marker in the media stream metadata. When the client receives the advertising marker, the client may request and receive chunks associated with a previously identified advertisement. … [T]he server may provide a list of potential advertisements in an initial manifest." Ex-1034, 10:30-39. Moreover, Luby teaches that "[c]hanges in an adaptive HTTP streaming media presentation description may be needed when content from multiple different sources is spliced together, for example, when advertising has been inserted. The content from different sources may differ in a variety of ways. Another reason, during live presentations, is that it may be necessary to change the URLs used for content files to provide for fail-over from one live origin server to another." Ex-1031, 37:64-38:4. Additionally, "[t]o avoid frequent updates of the media presentation description to get knowledge on

Petitioner Disney Ex-1002, 0040

potential updates, it is advantageous to signal any such updates along with the media segments. There may be provided an additional element or elements within the media segments themselves which may indicate that updated metadata such as the media presentation description is available and has to be accessed to within a certain amount of time to successfully continue creation of accessible segment lists. In addition, such elements may provide a file identifier, such as a URL, or information that may be used to construct a file identifier, for the updated metadata file." *Id.*, 41:49-60.

57. URL beacons were well known "as a module or device that broadcasts or transmits the web address in accordance with a predetermined open standard communication protocol." Ex-1038 (Krishnan) ¶37 ("The beacons … can be implemented by any known beacon that broadcasts or transmits signals (referred to as beacon signal)."). Bailey further teaches that "the URL beacon 23 of the Internet appliance 20 is used to transmit electronically or 'broadcast' to any external receiving device the web address of the web site (or Internet site) which is currently accessed by the Internet appliance 20. … Other receiving devices may also receive the web address transmission from the URL beacon 23. These other receiving devices may be a PDA (personal digital assistant), a pager, a watch, a cellular phone, or any other kind of information appliance." Ex-1038 (Krishnan) ¶35. Moreover, "the remote operator 10 can also receive web addresses from the web address beacon

Petitioner Disney Ex-1002, 0041

15," which functions substantially the same "as that of the URL beacon 23" and "is typically associated with a web site or Internet site and transmits the web address of that web site." Ex-1038 (Krishnan) ¶¶36, 41 ("the Internet appliance remote operator 10 can also remotely receive beacon signals containing web addresses from web address beacons (e.g., beacon 15) that are associated with web sites or from URL beacons (e.g., URL beacon 23) that are embedded in Internet appliances (e.g., the Internet appliance 20)"), 46 ("The beacon sender 52 is used to electronically transmit web addresses, one address at a time, to external Internet."), 48 ("The beacon receiver 53 is used to electronically receive or capture web addresses transmitted from external and remote web address beacons (e.g., the web address beacon 15 or URL beacon 23 of FIG. 1), one address at a time.").

58. *See also* Ex-1037 (Bailey), 1:56-2:8 ("A web beacon (also known as a web bug) is a particular type of embedded content where the content itself is irrelevant, but the request for content carries useful information. For example, a web beacon is often a transparent image having very small dimensions, such as 1 pixel by 1 pixel. This image is small enough to be invisible to the user. When a client is rendering a web page that includes a web beacon, the web beacon causes the client to send a resource request to a server such as a data collection server. The web beacon may include a script (or other code) that causes the client to include, in the resource request, additional information about the user and the user's environment.

Petitioner Disney Ex-1002, 0042

The additional information can include the data from a cookie, or other information about the client's operating environment or status. Where the server indicated by the web beacon code is a data collection server, the data collection server may, in response to the request, cause the client to set an additional cookie for identification for tracking purposes. In this manner, the web beacon request can be used to indicate to a data collection server that a particular web page is being rendered."); Ex-1036 (Christian) ¶¶20 ("[A] web beacon is generated by placing a small amount of code, or beacon code, into production web pages. Clients request and are served the production web page from a web server in a data center. When the production web page is processed by the client, the beacon code causes the client browser to retrieve web beacons from beacon servers specified in the beacon code. The requests are performed in the background of the client so no interference occurs with the current web page load from the web server."), 21 ("a globally unique wildcard hostname is generated for a web beacon download and that beacon URL is embedded in an existing web result and sent to the client which proceeds to download that image"), 25 ("The beacon server responds to the request by sending a single pixel, zero-content image, or HTTP 204 to the client, depending upon the nature of the request."), 26 ("The process begins when client 200 requests a web page and is served the web page from web server 202. Within the web page is beacon code, that instructs client 200 to retrieve a web beacon, or a zero-content image, from a location

Petitioner Disney Ex-1002, 0043

specified in the beacon code. When the web page is processed by the client 200, retrieval of the web beacon begins."), 27 ("A globally unique hostname (e.g., '12345.dnsb.company.com') is used as the location of the web beacon. Client 200 sends the request for the web beacon to LDNS resolver 204, that is owned by the ISP of client 200, in order to resolve the hostname into an IP address. … The beacon resolver 206 then returns to the LDNS resolver 204 the IP address of the beacon collection server 208 where the client may request the web beacon. The LDNS resolver 204 forwards the IP address of the beacon collection server 208 to client 200. Client 200 then sends the HTTP request ('12345.dnsb.company. com') to the beacon collection server 208. The beacon collection server 208 receives the request from the client and logs the IP address of client 200 and the unique wildcard of the hostname (i.e., '12345'). The beacon collection server 208 responds to the client 200 with a zerocontent image or an HTTP 204 (no download) status.").

I. Including a beaconing URL in the manifest which, when used, is indicative that a certain point has been reached in playback of the requested media source

59. Including a beaconing URL in the manifest which, when used, is indicative that a certain point has been reached in playback of the requested media source was a known concept by 2010. For example, Guzman discloses that "all major commercial and standard solutions for HTTP adaptive streaming use a sort of manifest file to introduce the media resource to the client. The manifest file is a

Petitioner Disney Ex-1002, 0044

structured collection of metadata and segments URL location. The HTTP-Streaming client requests and downloads the manifest file prior the start of the streaming session." Ex-1030, 33; *see also* Ex-1029 (Strasman) ¶122 ("Method 180 can force clients to send multiple manifest file requests by sending to the clients modified manifest files that allow the retrieval of only one part per media file—thus providing time based segmentation of the media files."), Fig. 7 ("Delaying the transmission of the acknowledgement message until reaching a time that corresponds to the allocated bit rate"); Ex-1034 (Sood), 4:4-8 ("The metadata may describe useful information about the media content, such as the bit rate of the media content, where the media content fits into a larger element (e.g., this chunk represents offset 1:10 in a 10 minute video clip"), Cl. 7 ("a clock synchronization component configured to synchronize clocks of the server and the client so that the server and client can identify particular chunks based on time"); Ex-1031 (Luby), 18:45-65 ("In order to formulate partial GET requests for Movie Fragments, the client may be informed of the byte offset and start time in decoding or presentation time of all media components contained in the fragments within the file or segment and also which fragments begin or contain a Random Access Points (and so are suitable to be used as switch points between alternative representations), wherein this information is often referred to as the segment indexing or segment map. The start time in decoding or presentation time may be expressed directly or may be expressed as deltas relative

Petitioner Disney Ex-1002, 0045

to a reference time. This time and byte offset indexing information may require at least 8 bytes of data per movie fragment. As an example, for a two hour movie contained within a single file, with 500 ms movie fragments, this would be a total of about 112 kilobytes of data. Downloading all of this data when starting a presentation may result in a significant additional startup delay. However, the time and byte offset data can be encoded hierarchically, so that the client can quickly find a small chunk of time and offset data relevant to the point in the presentation at which it wishes to start."), Fig. 6.

60. *See also* Ex-1037 (Bailey), 4:65-5:17 ("Web page 102 includes beacon code, which in one embodiment is a pointer to a beacon …. For purposes of the following description, a beacon is any element that is embedded in a web page 102 which is loaded automatically by browser 110 that references an external server 106 and is used to monitor traffic. The beacon code can be provided as a script (such as a JavaScript script) to be executed by scripting engine 116. The beacon code causes client machine 107 to generate resource requests 105 to data collection server 106. These resource requests 105 are usually dynamically generated according to the script instructions. Data collection server 106 records such requests in a log 108, and can also record ***additional information associated with the request*** (***such as*** the date and ***time***, and possibly some identifying information that may be encoded in the resource request).").

Petitioner Disney Ex-1002, 0046

IX. CLAIM CONSTRUCTION

61. For the purposes of my Declaration, I identify the following claim construction issues. I do not currently believe that any other terms require express construction.

A. "Chunking protocol" (Claim 8)

62. I understand from Counsel that in the Disney litigation, Petitioner has proposed that the recited "chunking protocol" in Claim 8 is a "protocol that dynamically creates segments of media based on a request." Ex-1025, 2; *see also* Ex-1001, 2:15-19, 8:16-19, 9:42-47, 13:62-67. PO has proposed that "[p]lain and ordinary meaning" should apply. Ex-1026, 1. As I explain below, in my opinion, there is no meaningful difference between these constructions as applied to the prior art references at issue in my Declaration.

B. "At least one beaconing URL which, when used, is indicative of [playback of at least one advertisement / a certain point has been reached in playback of the requested media source" (Claims 14-15)

63. I understand from Counsel that in the Disney litigation, Petitioner has proposed that these limitations are indefinite. *See* Ex-1025, 2. PO has proposed that "[p]lain and ordinary meaning" should apply. Ex-1026, 1. For the purposes of this proceeding only, I have applied these limitations to include PO's apparent interpretation of the term–i.e., that a URL is used to signal or track playback of the advertisement or requested media source.

C. Order of steps 8[b] to 8[e] (Claim 8)

64. I understand from Counsel that in the Disney litigation, Petitioner has proposed that steps 8[b] to 8[e] "must be performed in order" (Ex-1025, 2) and PO has proposed that these steps "do not need to be performed in sequential order." Ex-1026, 1. As I explain below, in my opinion, there is no meaningful difference between these constructions as applied to the prior art references at issue in my Declaration because the prior art teaches performing the steps in the order recited by the claims.

D. Order of steps 8[g] to 8[j] (Claim 8)

65. I understand from Counsel that in the Disney litigation, Petitioner has proposed that steps 8[g] to 8[j] "must be performed in order" (Ex-1025, 2) and PO has proposed that these steps "do not need to be performed in sequential order." Ex-1026, 2. As I explain below, in my opinion, there is no meaningful difference between these constructions as applied to the prior art references at issue in my Declaration because the prior art teaches performing the steps in the order recited by the claims.

X. BRIEF DESCRIPTION OF THE APPLIED PRIOR ART

A. Lewis (Ex-1005)

66. Lewis (U.S. Patent No. 8,677,428) was filed on August 20, 2010. As explained to me by counsel, I understand that Lewis qualifies as prior art under at least 35 U.S.C. §102(e).

Petitioner Disney Ex-1002, 0048

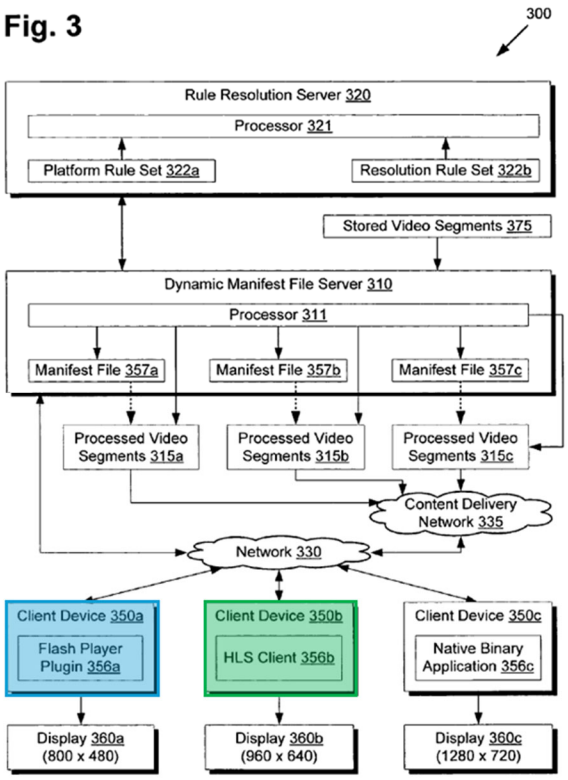
67. Lewis teaches a "method for rule-based dynamic server-side streaming manifest files." Ex-1005, Abstract. Lewis teaches that "a client device 150 may send a request" by executing media player application 156 and clicking "on a link directed to dynamic manifest file server 110 for access to live video stream content." Ex-1005, 3:16-28. Lewis further teaches that "[m]edia player application 156 may then send a request, such as a HTTP GET request over network 130 to dynamic manifest file server 110," which forwards "various parameters received from the request originating from client device 150 to rule resolution server 120 … to evaluate various business rules and create a dynamically tailored manifest file accordingly." Ex-1005, 3:28-38. The manifest file is then "passed back to client device 150 over network 130 and placed into memory 155 as manifest file 157." Ex-1005, 3:38-41.

68. As shown in Figure 3 below, Lewis's "dynamic manifest file server 310 provides manifest files for a diverse range of client device platforms, including Flash Player plugin 356a at client device 350a" (blue) and "HTTP Live Streaming client 356b at client device 350b" (green). Ex-1005, 5:60-65, Fig. 3. For instance, Lewis teaches that for client device 350a, which uses Flash Player plug-in support, "dynamic manifest file server 310 should preferably generate a F4M Flash Zeri manifest file, or manifest file 357a, referencing F4F Flash video files, or processed video segments 315a." Ex-1005, 6:5-10. "Processed video segments 315a … may be resized and cropped according to resolution rule set 322b to optimally fit the 800

Petitioner Disney Ex-1002, 0049

by 480 resolution provided by display 360a." Ex-1005, 6:10-14. Dynamic manifest file server 310 then provides manifest file 357a "in the Flash Zeri HTTP streaming format for facilitated playback by Flash Player plugin 356a," which accesses "the referenced video content in processed video segments 315a over network 330." Ex-1005, 6:14-21. Lewis further teaches that for client device 350b, which uses HLS or HTTP Live Streaming support, "dynamic manifest file server 310 should preferably generate a M3U8 manifest file, or manifest file 357b, referencing MPEG transport stream video files, or processed video segments 315b." Ex-1005, 6:22-27. "Processed video segments 315b … may be resized and cropped according to resolution rule set 322b to optimally fit the 960 by 640 resolution provided by display 360b." Ex-1005, 6:28-31. Dynamic manifest file server 310 then provides "manifest file 357b in the HTTP Live Streaming format for facilitated playback by HLS client 356b," which accesses "the referenced video content in processed video segments 315b over network 330." Ex-1005, 6:31-38.

Petitioner Disney Ex-1002, 0050



B. Pyle (Ex-1006)

69. Pyle (U.S. Patent No. 8,782,268) was filed on November 3, 2010. As explained to me by counsel, I understand that Lewis qualifies as prior art under at least 35 U.S.C. §102(e).

70. Pyle teaches "dynamic composition of media for streaming to consuming devices." Ex-1006, 1:14-15. Pyle teaches that "a request for the content can be transmitted (e.g., by a client) or received (e.g., by a server)," which includes "a set of characteristics or attributes defining a suitable representation of the content." Ex-1006, 32:45-49. The client requests are, for example, "time-based segment request URLs" (Ex-1006, 5:4-7, 5:21-22), which specify or describe the

"locations" of the content 206 "included in one or more manifest 204" (Ex-1006, 11:33-36) as well as "local configuration" information used "for determining a suitable representation … of content 206." Ex-1006, 12:19-23. Pyle further teaches that "a response to the request can be transmitted" to the client, which includes "a suitable manifest associated with the suitable representation of the content." Ex-1006, 32:43-53.

71. As shown in Figure 12 below, Pyle teaches computing objects or devices 1220 (blue), 1222, 1224, 1226, and 1228 (green), which "can be thought of as clients." Ex-1006. 34:21-22, Fig. 12. Pyle further teaches a composition component "configured to select a particular selected manifest 418 from the set of available manifests 204 based upon data included in a request 420 for content 206." Ex-1006, 10:22-27. For example, if the content is a movie, a requesting device such as computing device 1220 can be "a HD television with very large form factor," which "can further be configured to only present media that is rated PG-13 (unless parent codes are entered) and is coupled to a reliable network with high bandwidth in an English-speaking household." Ex-1006, 10:27-33. As a second example, a requesting device such as computing device 1228 "can be a smart phone with a more modest form factor and resolution support, and is currently in transit at high speeds resulting in a less reliable network connection" where "the default language … is set to German," and "user preference indicates hearing impairment." Ex-1006, 10:33-

Petitioner Disney Ex-1002, 0052

39, 10:53-56. Pyle explains that "[a]lthough both requesting devices can request the very same movie … it is readily apparent that these two distinct devices and/or configurations or preferences, will not demand the same representation 208 of that movie." Ex-1006, 10:40-44. Composition component 210 thus selects a suitable manifest and transmits it to the requesting device. Ex-1006, 10:53-56.

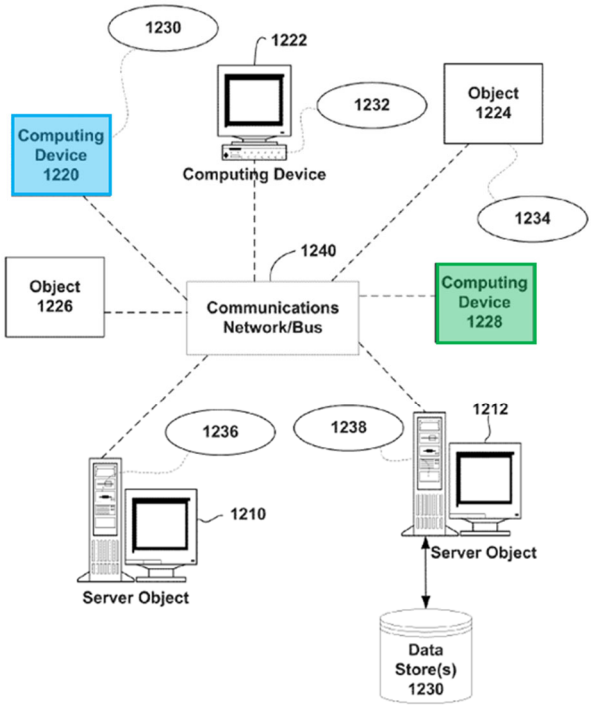


FIG. 12

C. Barraclough (Ex-1007)

72. Barraclough (U.S. Patent No. 9,047,235) was filed on December 28, 2007. As explained to me by counsel, I understand that Lewis qualifies as prior art under at least 35 U.S.C. §102(e).

73. Barraclough teaches "approaches to media content routing, management and associated applications for making a registered user's media remotely accessible." Ex-1007, 2:25-27. Barraclough teaches that "[w]hen the registered user … wishes to access a playlist at a remote playback device, the user accesses a web page hosted by the host server, by way of a remote Internet access device such as a hand-held wireless telephone-type device." Ex-1007, 7:52-56. The user then "provides authentication information, such as the user name and password … and in response the host authenticates the user as an authorized user, and facilitates media playlist access." Ex-1007, 7:56-62. Barraclough further teaches that "the device making the request is classified and the device class is used to map to an appropriate encoding of the file to deliver." Ex-1007, 16:22-26. For example, "the device classification is retrieved from data stored for a particular user (e.g., at the data storage arrangement 330)." Ex-1007, 16:26-31. Host server 320 then "locates the appropriate version of the requested media file in the network-based content storage 360 and serves the contents of the file." Ex-1007, 16:31-33, Fig. 3.

D. Jain (Ex-1008)

74. Jain (U.S. Patent Publication No. 2010/268573) was filed on April 17, 2009 and published on October 21, 2010. As explained to me by counsel, I understand that Lewis qualifies as prior art under at least 35 U.S.C. §§102(a) and (e).

Petitioner Disney Ex-1002, 0054

75. Jain teaches an audio beaconing system that includes an application programming interface (API) 113, which "is configured as a beaconing API object" that may reside "on the content server 100, where the API object may be initialized." Ex-1008 ¶¶18-19. Jain further teaches that "video beacons can be embedded within an action script that will be running within the video Flash Player's run time environment on web page 110," such that when the page is accessed and the script is activated, the video beacon is triggered "which extracts and store[s] URL information on a server (e.g., content server 100), and launches the video Flash Player." Ex-1008 ¶26. When the beacon is triggered, "a pre-recorded audio file containing a special embedded compatible code (e.g., CBET)" is loaded from a server, which is "utilized for beacon 131 to transmit for a given period of time (e.g., every x seconds)." Ex-1008 ¶26. Moreover, as shown in Figure 2 below, Jain teaches "additional beacon events" 203, where as an "advertisement is served, an event beacon, which may include advertisement URL data, is transmitted." Ex-1008 ¶33, Fig. 2 (yellow). Jain teaches that the "beacon also transmits a timestamp" and "[a]t the conclusion of the video, a video end beacon is transmitted to indicate the user/panelist is no longer viewing specific media." Ex-1008 ¶33.

Petitioner Disney Ex-1002, 0055

200	Time →	0-3 sec	4 sec ---------- 180 ---------- 300 sec				301-315 sec	316—599 sec	600 sec
201	Activities→	Video Loaded	Video Played				Ad served	Video continues	Video End
202	Audio Beacon detected by portable device →	Continuous Audio representing the website (e.g., Hulu.com)							
203	Additional beacon events →	URLs (Hulu.com & Video ID)					AD Event Beacon (co-relation URL)		Video End Beacon
204	Audio matching events detected by portable device →		Audio Signature Sample 1	Audio Signature Sample 2	Audio Signature Sample 3 (Content Match Achieved)	Same Signature Sample 4 (No Change)	Audio Signature Sample 5 (Different Content couldn't match)	Audio Signature Sample 5 (Same Video again)	No Signature (But Can't assume video is completed
205	Overlay portable device data	X			X		X	X	X
206	End- User Experience	URL of site where the video was played			Content Matched		Ad Served		Video Played till the End

FIG. 2

XI. DETAILED EXPLANATION OF THE UNPATENTABILITY GROUNDS

A. Grounds 1-2: Claims 8 and 13 are obvious over Lewis alone or Lewis in view of Barraclough.

1. A POSITA would have been motivated to combine Lewis's teachings with Barraclough's teachings and would have had a reasonable expectation of success.

76. A POSITA would have been motivated to combine the teachings of Lewis and Barraclough and had a reasonable expectation of success in doing so because each address server side selection of streaming media in response to the requesting device's, for example, characteristics and/or capabilities. Ex-1005, Abstract, 6:6-38, Fig. 3; Ex-1007, 3:22-36, 14:60-15:3. Used together, these complementary techniques would predictably allow format/encoding selection from

Petitioner Disney Ex-1002, 0056

both live request data and per user stored data, reducing decision latency and computation while achieving the same delivery objective.

77. Lewis teaches that "parameter data from the HTTP GET request" include details such as, for example, "browser or operating platform, … screen resolution of display 160, and other device, display, or user parameters, which may then be evaluated against a plurality of rules as applied to live video segments 175." Ex-1005, 8:42-52. For example, Lewis teaches that "when ***client device 350a*** requests video content represented by stored video segments 375, dynamic manifest file server 310 may ***provide manifest file 357a in the Flash Zeri HTTP streaming format*** for facilitated playback by Flash Player plugin 356a" and "when ***client device 350b*** requests video content represented by stored video segments 375, dynamic manifest file server 310 may ***provide manifest file 357b in the HTTP Live Streaming format*** for facilitated playback by HLS client 356b." Ex-1005, 6:6-38. A POSITA would thus have understood that Lewis teaches determining the device type (i.e., Flash Player Plugin 356a and HLS client 356b) based on the identity of the client (i.e., client device 350a and client device 350b, respectively).

78. Moreover, Barraclough teaches that "[w]hen the host server 320 receives a request for a media file either directly or by way of a playlist request, the device making the request is classified and the device class is used to map to an appropriate encoding of the file to deliver." Ex-1007, 16:22-26. Additionally,

Petitioner Disney Ex-1002, 0057

Barraclough teaches that in some applications, "the ***device classification is retrieved from data stored for a particular user*** (e.g., at the data storage arrangement 330)." Ex-1007, 16:26-31. Barraclough teaches that host server 320 then "locates the appropriate version of the requested media file in the network-based content storage 360 and serves the contents of the file." Ex-1007, 16:31-33.

79. Thus, in view of both Lewis and Barraclough, a POSITA would have been motivated to determine the device type based on the identity of the client. Lewis provides live, per request capability inference; Barraclough adds per user stored classification. Combining them predictably reduces rule evaluation overhead/latency while achieving the same selection goal. For example, Lewis's method teaches to use device classification based on the identity of the device, and Barraclough teaches complimentary information stored for the particular user of a particular device. These teachings would have been implemented together because storing the user's device types, as taught by Barraclough, would further Lewis's goals of "client customization" and "user and device targeting." Ex-1005, Abstract. Storing the user's device types in a profile or similar means would make Lewis's dynamic manifest generation more efficient because the system would know in advance the limited number of devices that a particular user may make stream requests from, which would allow the system overtime to respond quicker to requests. For example, if the system knows that a user only makes requests from a

Petitioner Disney Ex-1002, 0058

smartphone or an HD TV, the system only has to generate manifests for those two formats and does not have to analyze a long list of formats for a particular user to determine the device type. In the Lewis-Barraclough combination, Lewis's dynamic manifest file server determines the first device type (i.e., Flash Player Plugin 356a) and second device type (i.e., HLS Client 356b) based on data stored for the particular client (i.e., first client device 350a and second client device 350b, respectively), as taught by Barraclough.

80. In addition to the motivations above, a POSITA would have understood that Lewis and Barraclough are directed to the same core problem and provide complementary solutions. Specifically, Barraclough solves what to deliver (e.g., playlist items, in suitable formats/bitrates, from cloud storage) and who/when to deliver (e.g., host server authentication and control). Lewis solves how to package the actual stream at request time via a device appropriate manifest with segment lists (and ads). Combining these references thus produces a cloud playlist service that returns a dynamic manifest per playlist item/session, an obvious fit because Barraclough already contemplates server-side formatting for device/bandwidth and server-controlled streaming.

81. A POSITA would have further understood that the combination would provide the predictable benefits of device and bandwidth targeting and monetization via server-side ad insertion. Specifically, for device and bandwidth targeting,

Petitioner Disney Ex-1002, 0059

Barraclough's host decides the format/bitrate for a device or link; and Lewis's rules already tailor manifests for HLS/Flash and specific resolutions, and expose multi-bitrate ladders for graceful adaptation. Swapping in Lewis's packaging at Barraclough's "delivery" step therefore predictably yields per device manifests that reflect the host's selections, with no new research needed. For monetization via server-side ad insertion, Barraclough focuses on access and formatting; and Lewis adds precise, server-side ad insertion (mid roll, pre roll) by cutting/pointing manifests at ad segments—even for live streams. Integrating Lewis's manifest generation into Barraclough's delivery stage therefore allows the host to monetize any playlist stream with consistent server-side control.

82. Additionally, a POSITA would have understood that the combination would allow for natural layering of the "playlist" to "manifest." Specifically, in Barraclough, a playlist is the user/service selection and cloud organization of content items (including multiple stored formats/bitrates). In Lewis, a manifest is the per session, per device playback instruction list (segments/ads) for one of those items. Treating the playlist selection as the input and the manifest as the output in the combination is therefore a straightforward, modular integration that aligns with how HLS/DASH systems are architected.

83. Finally, a POSITA would have understood that the combination would address market pressure and is simply common sense. Specifically, both references

Petitioner Disney Ex-1002, 0060

explicitly address heterogeneous devices and variable networks. Combining server-side formatting (Barraclough) with server-side manifesting (Lewis) addresses those pressures in the most direct, industry standard way, improving scalability and user experience while keeping clients simple. A POSITA would therefore have find the combination to provide an expected, routine engineering step.

84. As I also explained in the Technological Background section (*see* Section VIII.D), this was a known concept by 2010. *See, e.g.*, Ex-1022 (Zapata) ¶¶192-193, 199; Ex-1021 (Barraclough'545), 6:11-22; Ex-1024 (Mehrotra) ¶2; Ex-1023 (Gopalakrishnan), 7:28-30, 14:14-19.

85. A POSITA would have had a reasonable expectation of success in combining the teachings of Lewis and Barraclough. Both describe methods for streaming media content in response to a request based on characteristics and/or capabilities of the requesting device in a similar manner such as by storing the media content in different formats corresponding to different device types and selecting a suitable format based on the requesting device type. Ex-1005, Abstract, 6:6-38, Fig. 3; Ex-1007, 3:22-36, 14:60-15:3. A POSITA would thus have understood that these references disclose interrelated teachings based on well-understood technologies that would have been amenable to various well-understood and predictable combinations. For example, Barraclough's per user classification complements

Petitioner Disney Ex-1002, 0061

Lewis's request driven dynamic manifest selection and would have been expected to work together predictably.

2. Independent Claim 8

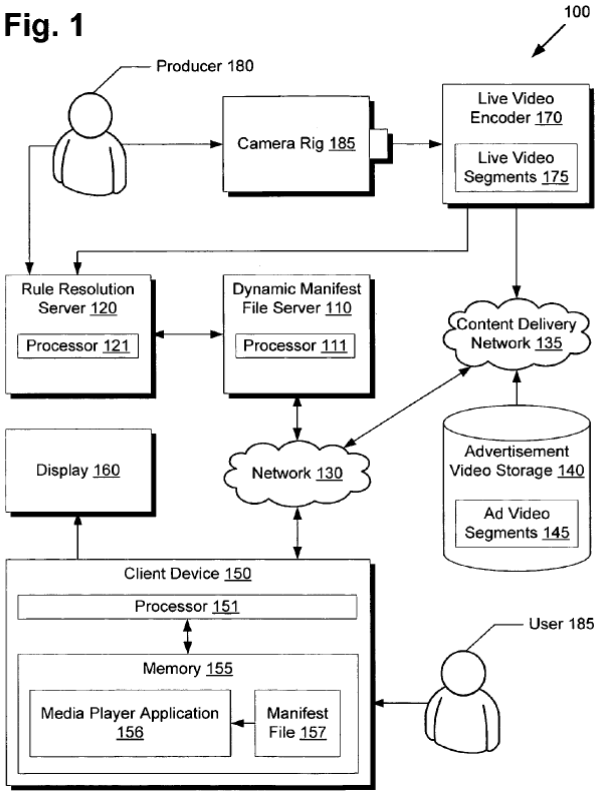
a. Element 8[pre]: A method of streaming media via a data communications network, the method comprising:

86. In my opinion, Lewis or Lewis in view of Barraclough ("Lewis-Barraclough") teaches[3] the preamble, to the extent it is limiting.

87. Lewis teaches a "method for rule-based dynamic server-side streaming manifest files," which includes "receiving a request to provide … video content for playback," "generating a dynamic manifest file referencing the [] video content," and "providing the dynamic manifest file in response to the request." Ex-1005, Abstract. As shown in Figure 1 below, Lewis further teaches that the media content is stored and distributed over content delivery network 135 comprising "a network of servers." Ex-1005, 3:4-6. As I also explained in the Technological Background section (*see* Section VIII.A), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), Abstract; Ex-1028 (Mao) at Figs. 1, 3; Ex-1023 (Gopalakrishnan), Abstract.

[3] I understand from Counsel that the term "teaches" includes both express teachings and those fairly suggested to a person of ordinary skill in the art.

Petitioner Disney Ex-1002, 0062



b. Element 8[a]: receiving a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source;

88. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[a].

89. Lewis teaches receiving a URL in a first request from a first client that indicates a requested media source. Specifically, Lewis teaches that "a client device 150 may send a request to dynamic manifest file server 110 for live video content." Ex-1005, 3:16-18. The user directs the client device "to execute media player application 156," which may comprise a web browser, "by clicking on an application icon shown on display 160." Ex-1005, 3:21-25. The user then navigates "to a video streaming portal site accessible over network 130 to ***click on a link directed to***

dynamic manifest file server 110 for access to live video stream content." Ex-1005, 3:25-28. Lewis further teaches that "[m]edia player application 156 may then send a request, such as ***a HTTP GET request*** over network 130 to dynamic manifest file server 110," which then forwards "various parameters received from the request originating from client device 150 to rule resolution server 120 … to evaluate various business rules and create a dynamically tailored manifest file accordingly." Ex-1005, 3:28-38. Moreover, as shown in Figure 3 below, Lewis allows users of a plurality of client devices including "client devices 350a through 350c" (Ex-1005, 5:37), to request content. Ex-1005, Fig. 3, 6:5-21 ("***client device 350a*** requests video content represented by stored video segments 375"). The HTTP GET request received in a request from a user of client device 350a (blue) corresponds to the recited "universal source locator (URL) in a first request from a first client" received by the dynamic manifest file server. The URL includes information indicative of the requested media source, which in Lewis is the requested "live video stream content." In Lewis, the "link directed to dynamic manifest file server 110" that the user clicks is itself a URL, and Lewis immediately explains that the client then issues an "HTTP GET request" to that server, i.e., a client request whose request-target is the URL identifying the resource to be retrieved (the live video stream). That URL therefore "includes information indicative of a requested media source" because it points the server to the specific live video stream endpoint that the user selected. Lewis further

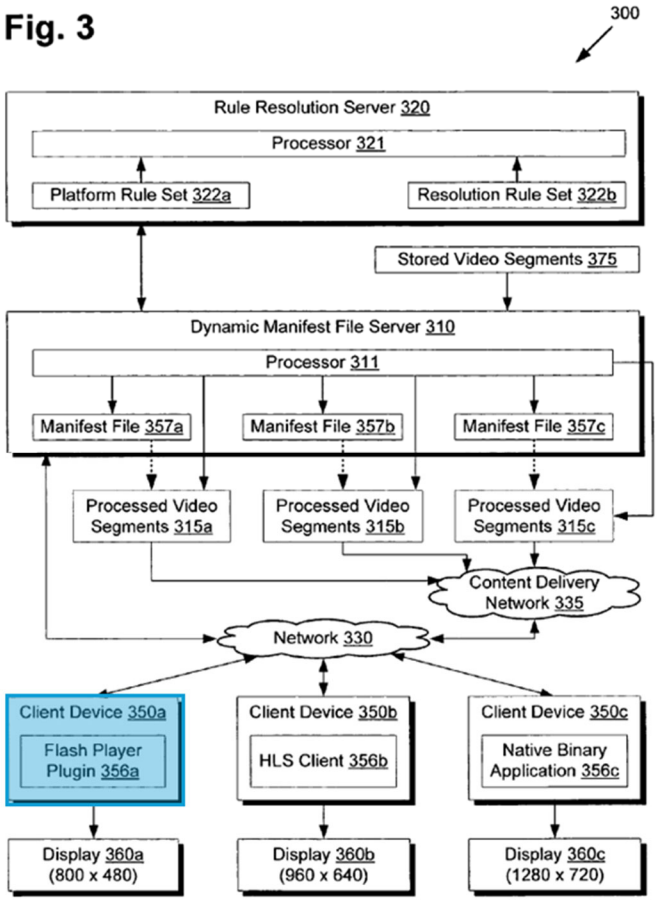
Petitioner Disney Ex-1002, 0064

shows concrete first-client examples: client device 350a (e.g., Flash player) and 350b (HLS client) each "requests video content represented by stored video segments 375." Those requests are made to server 310/110 via HTTP and are resolved into device-appropriate manifests, but the trigger is the client's URL hit to the streaming endpoint, again teaching receipt of a URL in a first client request that indicates the media source to be played.

90. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that Barraclough independently corroborates this URL-based request model for streaming media: its host server "receives a request for a media file either directly or by way of a playlist request," classifies the requesting device, and "locates the appropriate version of the requested media file … and serves the contents of the file." In Barraclough's web-based flow (browser gateway 114; remote device accessing a hosted web page), the "request for a media file" is issued via the network to the host server, and, just as in Lewis, identifies the media to be delivered (directly or by playlist ID embedded in the link), satisfying that the received request URL carries information indicative of the requested media source.

91. As I also explained in the Technological Background section (*see* Section VIII.B), this was a known concept by 2010. *See, e.g.*, Ex-1030 (Guzman), 46-47; Ex-1023 (Gopalakrishnan), 7:26-28; Ex-1029 (Strasman) ¶¶77, 107; Ex-1021 (Barraclough'545), 12:5-8; Ex-1039 (Gabin), 135; Ex-1031 (Luby), 13:56-59.

Petitioner Disney Ex-1002, 0065



c. Element 8[b]: determining an identity of the first client;

92. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[b].

93. Lewis teaches to determine the identity of the first client. For example, Lewis teaches that a URL's HTTP GET parameters may include client details such as "the ***client IP address***, … ***device identifiers***, … or ***user parameters***, which may then be evaluated against a plurality of rules as applied to live video segments 175." Ex-1005, 8:47-52. Lewis also teaches that "a ***user tracking profile*** associated with client device 350a may be analyzed to formulate a targeted advertising campaign for

the manifest file 357a … to be most relevant and interesting for ***the user of client device 350a***." Ex-1005, 7:13-19. Moreover, as shown in Figure 3 below, Lewis's system is designed to identify and distinguish different types of client devices that users may use: "dynamic manifest file server 310 provides manifest files for a diverse range of client device platforms, including Flash Player plugin 356a ***at client device 350a***, HTTP Live Streaming client 356b ***at client device 350b***, and native binary application 356c ***at client device 356c***." Ex-1005, 5:60-65, Fig. 3 (blue). The "user of client device 350a" corresponds to the recited "identity of the first client." Therefore, the first client's identity is determined by the rule resolution server using, for example, the client IP address, device identifier, and/or the user tracking profile, which are analyzed by the server to generate the manifest file. In Figure 3, distinct manifests (e.g., 357a vs. 357b) are returned for requests from different client devices (350a vs. 350b), which a POSITA would have understood requires the server to recognize which specific client originated the request, i.e., to determine the identity of the first client based on the parameters and/or associated profile. Ex-1005, 6:5-21; 5:60-65; Fig. 3.

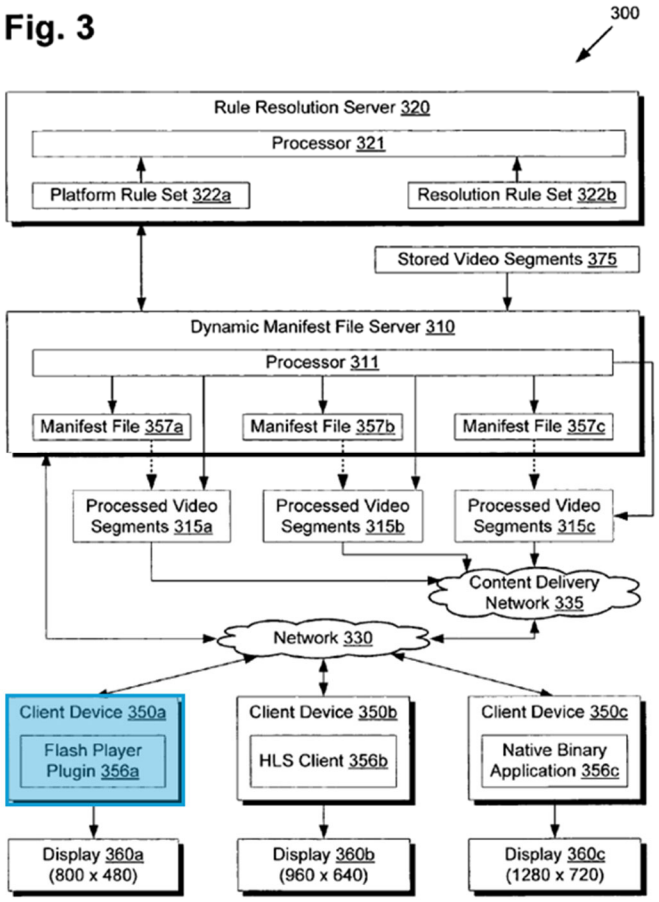
94. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the alternative, Lewis in combination with Barraclough also teaches determining the client's identity: Barraclough states that when the host server receives a request, "the device making the request is classified and the device

Petitioner Disney Ex-1002, 0067

class is used to map to an appropriate encoding," and that "the device classification is retrieved from data stored for a particular user," thereby explicitly tying the request to a specific user/device record. Ex-1007, 16:22-31. A POSITA would understand that using Barraclough's per-user stored classification together with Lewis's request parameters and user tracking profile yields a routine, server-side determination of the identity of the first client at the time of the first request. Ex-1005, 3:28-38, 7:13-19; Ex-1007, 16:22-31.

95. As I also explained in the Technological Background section (*see* Section VIII.C), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 12:15-21; Ex-1021 (Mehrotra) ¶29.

Petitioner Disney Ex-1002, 0068



d. Element 8[c]: determining a first device type based on the identity of the first client;

96. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[c].

97. Ground 1: Lewis teaches determining the first device type based on the first client's identity. For example, Lewis teaches that a device type may be determined using a user's "parameter data from the HTTP GET request," which includes device-specific details such as, for example, "browser or operating platform, … screen resolution of display 160, and other device, display, or user parameters, which may then be evaluated against a plurality of rules as applied to

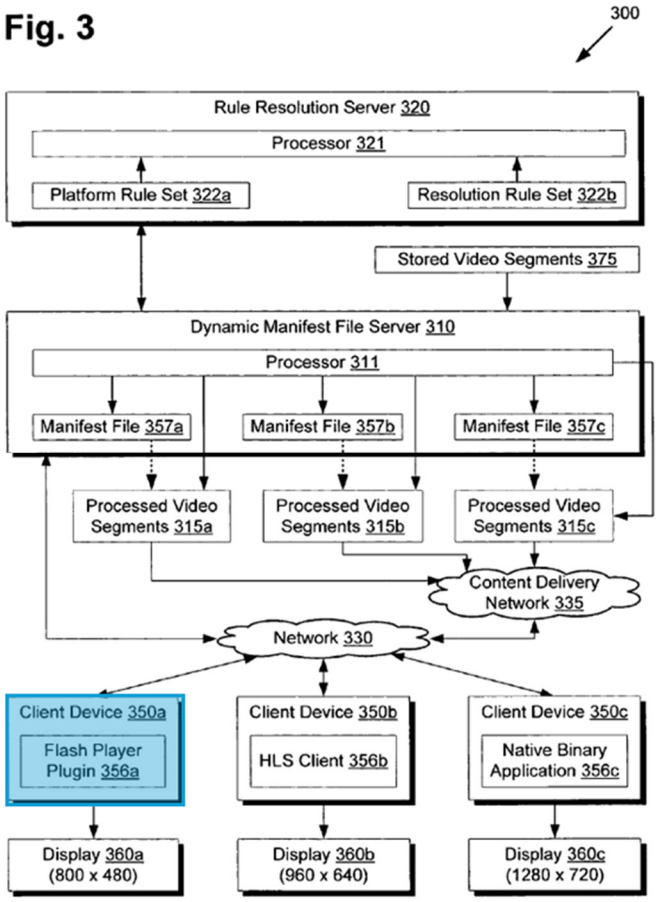
Petitioner Disney Ex-1002, 0069

live video segments 175." Ex-1005, 8:42-52; *see also id*., 3:18-21 (client device 150 "may comprise a laptop or desktop computer, a mobile phone, a video game console, a digital video recorder, a set top box, or another network enabled media device"). Moreover, Lewis also teaches "client or ***device targeting, wherein video content is processed and customized according to particular client device parameters***." Ex-1005, 5:58-60. As shown in Figure 3 below, Lewis teaches that its "dynamic manifest file server 310 provides manifest files for a diverse range of client device platforms, including ***Flash Player plugin 356a at client device 350a***." Ex-1005, 5:60-65. Lewis further teaches that platform rule set 322a includes "various rules as how to customize video content based on the target device platform to be supported." Ex-1005, 5:65-67. Each of the users of the exemplary client devices in Figure 3 (e.g., ***350a***, 350b, 350c) has a corresponding device type (e.g., ***Flash Player Plugin 356a***, HLS Client 356b, Native Binary Application 356c, respectively), which the server determines to generate the appropriate manifest file for that particular device. Ex-1005, Fig. 3 (blue). In other words, the server first recognizes which client issued the request (from the request's identifying parameters) and then classifies that client into a "target device platform," which is the device type used to drive manifest selection. For instance, Lewis teaches that for the user using "***client device 350a***" (i.e., the identity of the first client), "platform rule set 322a may dictate that if a request ***originates from a client device indicating Flash Player plug-in support***,

Petitioner Disney Ex-1002, 0070

then dynamic manifest file server 310 should preferably generate a F4M Flash Zeri manifest file, or manifest file 357a, referencing F4F Flash video files, or processed video segments 315a." Ex-1005, 6:5-21. Flash Player Plugin 356a corresponds to the recited "first device type." Lewis thus teaches determining the first device type (i.e., Flash Player Plugin 356a) based on the identity of the first client (i.e., the user using client device 350a). Lewis further describes forwarding "various parameters received from the request originating from client device [150/350x] to rule resolution server 120 … to evaluate various business rules and create a dynamically tailored manifest file accordingly," confirming that device-type determination occurs as a rule-driven classification that depends on the identified requesting client. Ex-1005, 3:28-38. And because Lewis emphasizes "client customization" and "user and device targeting," a POSITA would have understood that the identification of the first client (via IP address, browser/OS, plugin support, screen resolution, or other device/user parameters) is what triggers the server to determine the corresponding device type (e.g., Flash vs. HLS vs. native) before generating the tailored manifest. Ex-1005, Abstract; 5:58-67; 7:13-19; 8:42-52.

Petitioner Disney Ex-1002, 0071



98. To the extent PO argues that Lewis's device type determination is not based on the client's identity, this feature is further taught by Barraclough.

99. Ground 2: Barraclough teaches providing media content associated with a playlist over a packet-based network for remote access by a user. Ex-1007, Abstract. Specifically, Barraclough teaches that "[o]nce the network-based content storage 360 is populated with content, the host server 320 serves the populated content to ***authorized users*** at remote packet-based devices." Ex-1007, 16:20-22. Barraclough further teaches that "[w]hen the host server 320 receives a request for a media file either directly or by way of a playlist request, ***the device making the***

request is classified and the device class is ***used to map to an appropriate encoding of the file*** to deliver." Ex-1007, 16:22-26. Additionally, Barraclough teaches that in some applications, "***the device classification is retrieved from data stored for a particular user*** (e.g., at the data storage arrangement 330)." Ex-1007, 16:26-31. The device type classification is thus specifically based on the identity of a particular user that requested the media. Barraclough teaches that host server 320 then "locates the appropriate version of the requested media file in the network-based content storage 360 and serves the contents of the file." Ex-1007, 16:31-33, Fig. 3 (blue). In other words, Barraclough's host server ties the incoming request to a particular user and uses that user-linked record to classify the requesting device, then maps that classification to the version/encoding to serve. That "device class → encoding" selection is the claimed determination of a device type, and because Barraclough expressly performs the classification using information "stored for a particular user," the device-type determination is based on the identity of the first client (the requesting user/device record). Each request triggers this per-user lookup and classification, after which the server retrieves the version corresponding to that device type, directly tracking Element 8[c].

Petitioner Disney Ex-1002, 0073

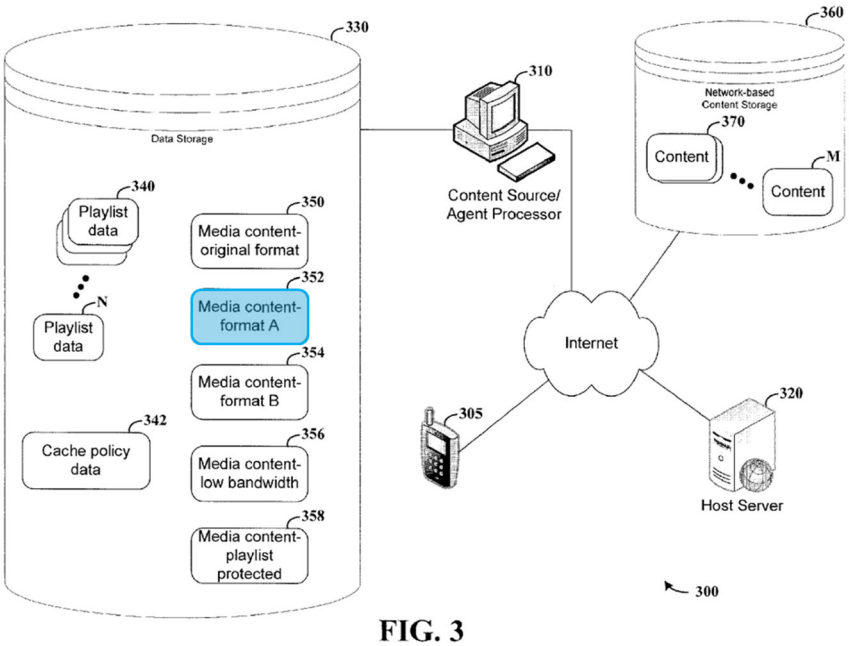


FIG. 3

100. In the Lewis-Barraclough combination, Lewis's dynamic manifest file server determines the first device type (i.e., Flash Player Plugin 356a) based on the particular client (i.e., user using first client device 350a), as taught by both Lewis's "user tracking profile" and Barraclough's "particular user" data storage. A POSITA would have been motivated to combine Lewis-Barraclough and had a reasonable expectation of success in doing so for the reasons described in Section XI.A.1. Barraclough supplies the identity-to-classification link (retrieving the device class from the requesting user's stored profile), and Lewis supplies the platform/device-type rules that drive manifest selection (e.g., platform rule set 322a producing a Flash-type manifest for a Flash-classified client, or an HLS-type manifest for an HLS-classified client). Thus, once the system associates the request with the identity of the first client (per Barraclough), Lewis's rule engine uses that identity-based

Petitioner Disney Ex-1002, 0074

classification to determine the first device type (e.g., Flash Player plugin 356a vs. HLS client 356b) and emits the corresponding manifest. This is a straightforward integration: (1) identify the particular user/client from the request or profile (Lewis or Lewis-Barraclough), (2) retrieve the stored device classification for that user, and (3) apply Lewis's platform rules to resolve that classification into the concrete device type used to generate the tailored manifest, meeting Element 8[c].

101. As I also explained in the Technological Background section (*see* Section VIII.D), this was a known concept by 2010. *See, e.g.*, Ex-1022 (Zapata) ¶¶192-193, 199; Ex-1021 (Barraclough'545), 6:11-22; Ex-1024 (Mehrotra) ¶2; Ex-1023 (Gopalakrishnan), 7:28-30, 14:14-19.

e. Element 8[d]: based at least in part on the determination of the first device type, generating, with a processor, a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network;

102. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[d].

(1) Generating a first manifest with a processor based on determination of first device type

103. Lewis teaches that "[d]ynamic manifest server 310 includes ***processor 311*** and manifest files 357a through 367c." Ex-1005, 5:40-41. As shown in Figure 3 below, "dynamic manifest file server 310 provides manifest files for a diverse range of client device platforms, including Flash Player plugin 356a at client device

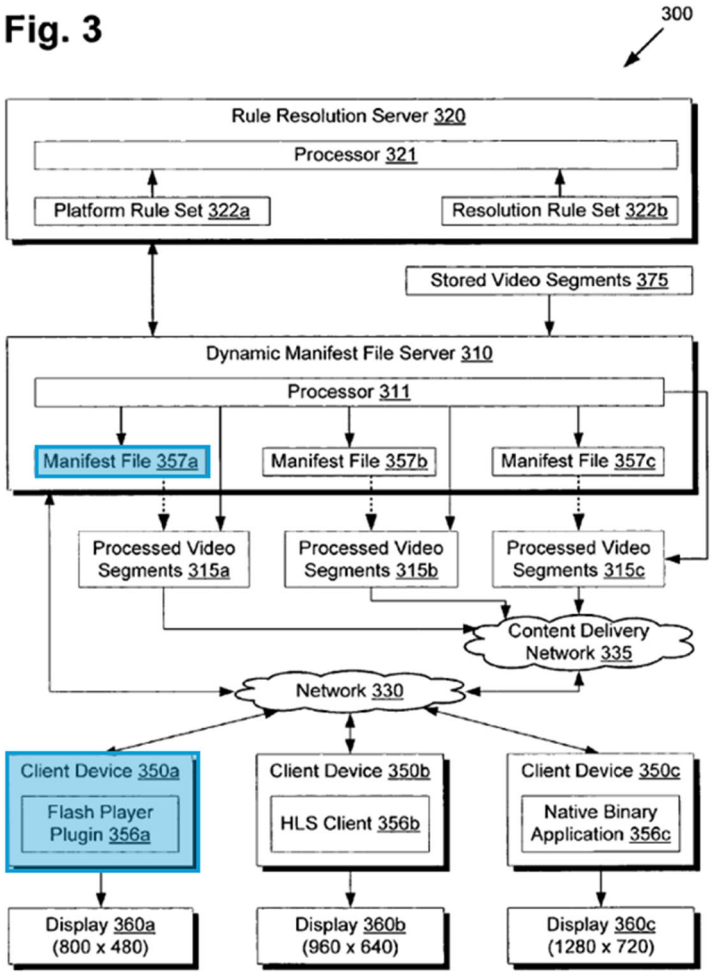
Petitioner Disney Ex-1002, 0075

350a." Ex-1005, 5:60-65. Moreover, Lewis teaches that platform rule set 322a includes "various rules as how to customize video content based on the target device platform to be supported." Ex-1005, 5:65-6:4. For example, "for client device 350a … if a request ***originates from a client device indicating Flash Player plug-in support***, then dynamic manifest file server 310 should preferably ***generate a F4M Flash Zeri manifest file***, ***or manifest file 357a***, referencing F4F Flash video files, or processed video segments 315a." Ex-1005, 6:5-10, Fig. 3 (blue). Manifest file 357a corresponds to the recited "first manifest" based on the first device type. This shows that the processor 311 executes rule-driven logic that first identifies the device type (e.g., Flash-capable) and then generates the corresponding manifest format for that type (e.g., F4M). Thus, the act of generation is "based at least in part on" the earlier device-type determination, exactly as recited in Element 8[d]. Ex-1005, 5:40-6:10.

104. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the Lewis-Barraclough combination, Barraclough's per-user device classification (retrieved "from data stored for a particular user") simply supplies the device-type input that Lewis's processor uses to select the appropriate manifest format and contents. Ex-1007, 16:22-31.

Petitioner Disney Ex-1002, 0076

Fig. 3



(2) Manifest has information for streaming one or more segments of the requested media source via the data communications network

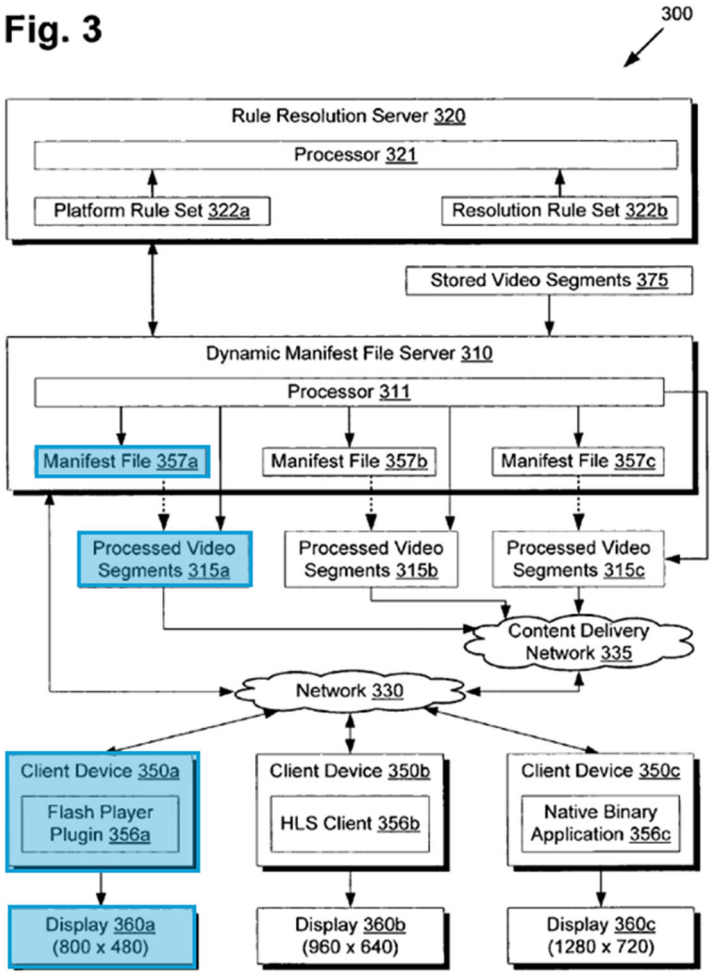
105. Lewis teaches that "***when client device 350a requests video content*** represented by stored video segments 375, dynamic manifest file ***server 310 may provide manifest file 357a in the Flash Zeri HTTP streaming format for facilitated playback*** by Flash Player plugin 356a." Ex-1005, 6:14-18. Lewis further teaches that "Flash Player plugin 356a may then access the referenced video content ***in processed***

video segments 315a over network 330 via content delivery network 335." Ex-1005, 6:19-21.

(3) In accordance with a first chunking protocol

106. Lewis teaches that the first manifest has information for streaming the requested media source segments in accordance with a first chunking protocol. Lewis teaches that for client device 350a, "[p]rocessed video ***segments*** 315a ***may be encoded*** from stored video segments 375, and may be ***resized and cropped*** according to resolution rule set 322b ***to optimally fit*** the 800 by 480 resolution provided by ***display 360a***." Ex-1005, 6:5-14, Fig. 3 (blue). Dynamic manifest file server 310 then provides manifest file 357a "in the Flash Zeri HTTP streaming format for facilitated playback by Flash Player plugin 356a," which accesses "the referenced video content in processed video segments 315a over network 330." Ex-1005, 6:14-21, Fig. 3 (blue). Lewis also teaches that the HTTP streaming format is a protocol that determines how the video segments are fragmented and sent to the client device. Specifically, Lewis teaches that "the live video footage may be ***encoded into MPEG transport stream (TS) fragment files of a fixed length, such as 10 seconds***, for ease of distribution ***using streaming protocols such as HTTP streaming***." Ex-1005, 2:64-3:1. The HTTP streaming protocol was a well-known protocol where media is sent in chunks over HTTP so that the client can start playing it before the entire file is downloaded. *See, e.g.*, Ex-1039 (Gabin), 134-137.

Petitioner Disney Ex-1002, 0078



107. Moreover, Lewis teaches that "if any of the rules require the insertion of content at a time offset starting or ending within a video segment, then ***a cutting or cropping processing procedure may be carried out*** with the affected video segments so that the generated manifest file can insert the content at any playback offset." Ex-1005, 8:65-9:3. For example, the processor for the dynamic manifest file server "may generate processed video segments 215 from live video segments 275 to allow the insertion of ad video segments 245 at 0:00:13 or the thirteen second playback offset within manifest file 257." Ex-1005, 9:3-8. This can be done by "[a]

direct stream copy and cutting procedure" or by re-encoding "to create processed video segments 215." Ex-1005, 9:12-15. This shows that the processor creates or adjusts segment boundaries (i.e., performs chunking) to satisfy rule outcomes (e.g., ad insertion at a precise offset), and then generates the manifest with references to those updated segments, again, generation "based at least in part on" the device type and in accordance with the selected chunking protocol. Ex-1005, 8:65-9:15.

108. Lewis's HTTP chunking protocol, which encodes fragment files of a fixed length into MPEG TS fragment files, when applied to the first manifest generated based on the determination of the first device type (i.e., Flash Player plugin), corresponds to the recited "first chunking protocol." Lewis thus teaches the "chunking protocol" limitation under either parties' construction because it teaches both a "protocol that ***dynamically*** creates segments of media based on a [user's] request" and more broadly teaches a pre-determined protocol that allows data to be sent in a series of smaller, manageable pieces. *See, e.g.*, Ex-1005, 7:4-6 ("To provide faster response time for client devices, video for the most common client configurations may be pre-processed and cached in advance."). Pre-processing/caching "for the most common client configurations" is exactly what a POSITA would recognize as preparing protocol-conformant segments in advance (e.g., F4F or HLS-style segments) so that, when the first device type is identified, the processor can quickly generate the corresponding first manifest pointing to those

Petitioner Disney Ex-1002, 0080

segments. Ex-1005, 7:4-6. And where the segmentation must be adjusted on the fly (e.g., ad insertion), Lewis describes the processor performing the necessary cutting/re-encoding and then issuing the updated manifest, still in accordance with the selected chunking protocol. Ex-1005, 8:65-9:15.

109. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the combination, Barraclough's per-user/device classification simply determines which protocol/format the manifest will follow; Lewis supplies the concrete segment list and offsets consistent with that protocol. Additionally, in the Lewis-Barraclough combination, Barraclough's host server ensures the correct version/encoding is available in network-based storage for the classified device, while Lewis's dynamic manifest file server emits the protocol-appropriate manifest that contains the information required to stream the one or more segments over the data communications network. Ex-1007, 16:20-33; Ex-1005, 6:14-21.

110. As I also explained in the Technological Background section (*see* Section VIII.E), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 4:31-35; Ex-1032 (Johnson) ¶¶36, 38, 57; Ex-1030 (Guzman) at 41, 33; Ex-1023 (Gopalakrishnan), Abstract, 27:46-48, 23:15-27; Ex-1029 (Strasman) ¶¶9, 77, Cl. 19; Ex-1043 (Begen), 59-60, 62-63; Ex-1033 (MSDN Magazine) at 35; Ex-1034 (Sood), 3:40-43, 4:7-8; Ex-1039 (Gabin), 134-137; Ex-

Petitioner Disney Ex-1002, 0081

1040 (Carmel), 2:4-6, 2:22-28, 7:23-25; Ex-1041 (Brueck), 3:36-39, 7:62-8:5; Ex-1042 (Bocharov), Fig. 5, ¶14.

f. Element 8[e]: sending the first manifest to the first client;

111. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[e].

112. Lewis teaches that "when ***client device 350a requests video content*** represented by stored video segments 375, ***dynamic manifest file server 310 may provide manifest file 357a*** in the Flash Zeri HTTP streaming format ***for facilitated playback*** by Flash Player plugin 356a." Ex-1005, 6:14-18; *see also id.*, 3:16-41 ("dynamic manifest file server 110 uses rule resolution server 120 to "create a dynamically tailored manifest file," "which may then be passed back to client device 150 over network 130 and placed into memory 155 as manifest file 157").

113. As I also explained in the Technological Background section (*see* Section VIII.F), this was a known concept by 2010. *See, e.g.*, Ex-1023 (Gopalakrishnan), 2:15-19; Ex-1029 (Strasman) ¶14.

g. Element 8[f]: receiving the URL in a second request from a second client;

114. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[f] for the reasons described in Section XI.A.2.b. Specifically, Lewis allows users of a plurality of client devices including "client devices 350a through 350c" (Ex-1005, 5:37), to request content. Ex-1005, Fig. 3, 6:22-38 ("***client device 350b*** requests

Petitioner Disney Ex-1002, 0082

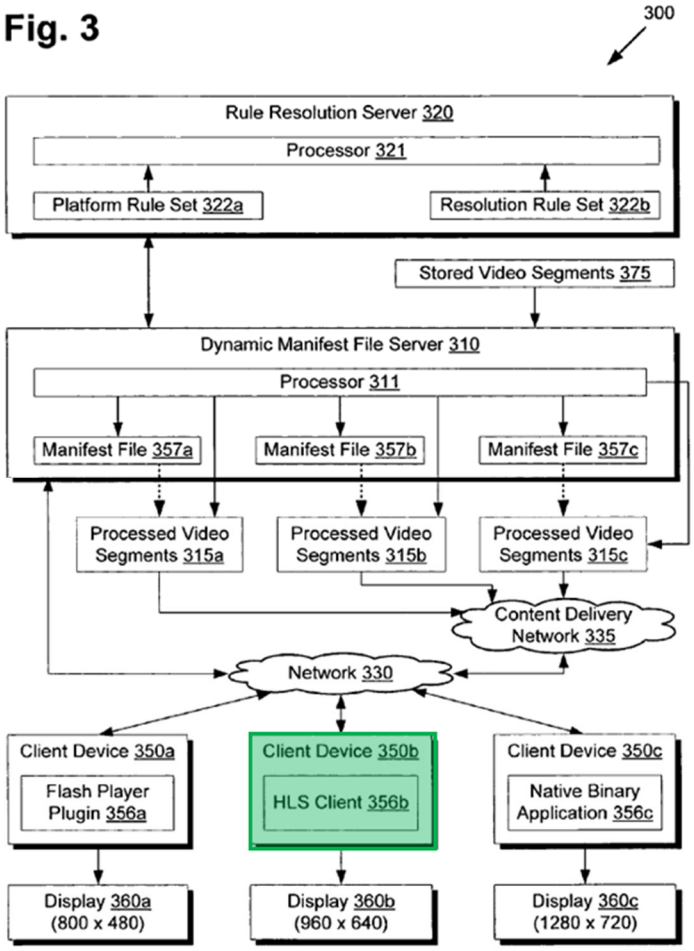
video content represented by stored video segments 375"). As discussed in Section XI.A.2.b, the HTTP GET request received in a request from a user of client device 350b (green) corresponds to the recited "URL in a second request from a second client" received by the dynamic manifest file server. Lewis expressly links the Figure 1 HTTP flow to each of the Figure 3 clients, stating that "client devices 350a through 350c" correspond to "client device 150" of Figure 1; in Figure 1, after the user clicks "a link directed to dynamic manifest file server 110," the media player "send[s] a … HTTP GET request" to that server. A POSITA would therefore have understood that when the second client (350b) requests the same video content, the server again receives a request containing the URL for that streaming endpoint, satisfying the "second request from a second client" requirement. Lewis also presents a parallel third-client case (350c) where "client device 350c requests video content," and the server provides the corresponding manifest; this reinforces that multiple distinct clients each issue their own URL-based requests to the same server.

115. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the Lewis-Barraclough combination, Barraclough teaches that the host server presents an Internet web page through which remote users on different devices submit playlist/media requests, i.e., separate network requests per user/device, and that the host "receives and authenticates playlist requests" and also "receives a request for a media file either directly or by way of a play list request."

Petitioner Disney Ex-1002, 0083

A POSITA would have recognized these web-page requests as URL-addressed requests and that a different remote user/device can issue a second such request to the same endpoint.

116. As I also explained in the Technological Background section (*see* Section VIII.C), this was a known concept by 2010. *See, e.g.*, Ex-1029 (Strasman) ¶107; Ex-1031 (Luby), 13:56-59.



h. Element 8[g]: determining an identity of the second client, the identity of the second client different from the identity of the first client;

117. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[g] for the reasons described in Section XI.A.2.c. Specifically, Lewis teaches that a URL's HTTP GET parameters may include client details such as "the ***client IP address***, … ***device identifiers***, … or ***user parameters***, which may then be evaluated against a plurality of rules as applied to live video segments 175." Ex-1005, 8:47-52. Lewis also teaches that "a user tracking profile associated with" the particular client device (i.e., 350b) "may be analyzed to formulate a targeted advertising campaign for the manifest file … to be most relevant and interesting for the user of client device." Ex-1005, 7:13-19. Moreover, as shown in Figure 3 below, Lewis's system is designed to identify and distinguish different types of client devices that users may use: "dynamic manifest file server 310 provides manifest files for a diverse range of client device platforms, including … HTTP Live Streaming client 356b at ***client device 350b***." Ex-1005, 5:60-65, Fig. 3 (green). The "user of client device 350b" corresponds to the recited "identity of the second client," which is "different from the identity of the first client." The second client's identity is determined by the rule resolution server using, for example, the client IP address, device identifier, and/or user tracking profile, which are analyzed to generate the manifest file. In the multi-client example, distinct requests from client device 350a versus 350b (and their

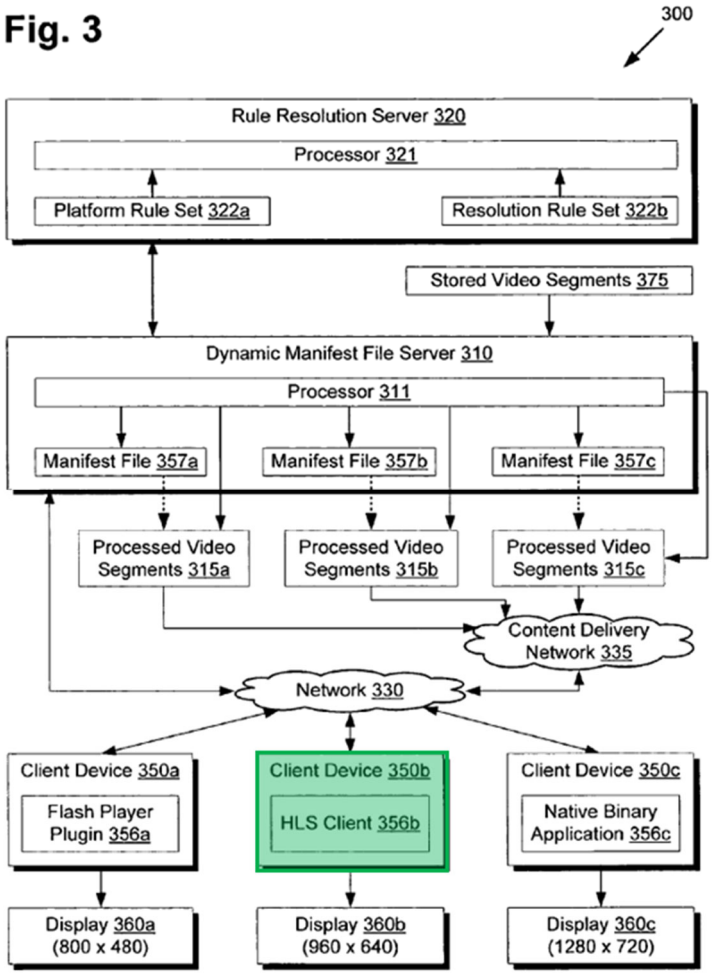
Petitioner Disney Ex-1002, 0085

respective user profiles) result in different manifests (e.g., 357a vs. 357b), confirming that the server recognizes and distinguishes the identity of the second client from that of the first client. Ex-1005, 6:6-38, 5:60-65; Fig. 3. A POSITA would have understood that the presence of unique request identifiers (IP address, device ID, cookies/login tokens, user profile references) enables routine determination that the second client (350b) is different from the first client (350a). Ex-1005, 7:13-19; 8:47-52.

118. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the Lewis-Barraclough combination, Barraclough serves content to authorized users and, upon receiving a request, "the device making the request is classified," with the "device classification … retrieved from data stored for a particular user," thereby linking each request to a specific (and potentially different) user/device record. Ex-1007, 16:20-31. When a subsequent request is associated with a different particular-user record or device record, the system necessarily determines the identity of the second client and that it is different from the first. Ex-1007, 16:20-31.

119. As I also explained in the Technological Background section (*see* Section VIII.C), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 12:15-21; Ex-1021 (Mehrotra) ¶29.

Petitioner Disney Ex-1002, 0086



i. Element 8[h]: determining a second device type based on the identity of the second client, the second device type different from the first device type;

120. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[h].

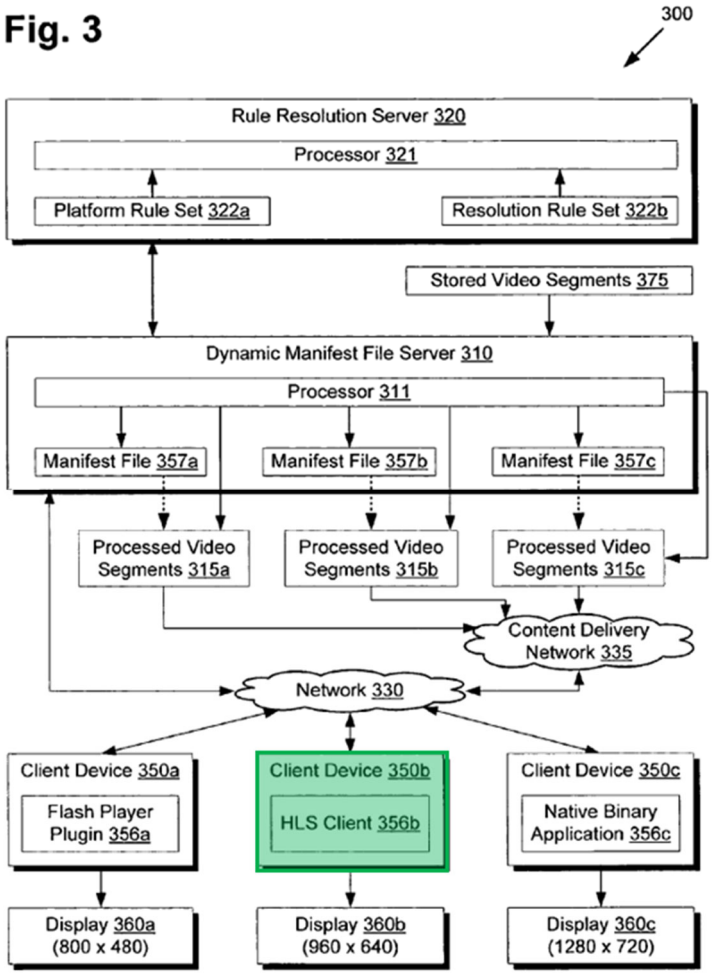
121. Ground 1: As explained in Section XI.A.2.d, each of the users of the exemplary client devices in Figure 3 (e.g., 350a, ***350b***, 350c) has a corresponding device type (e.g., Flash Player Plugin 356a, ***HLS Client 356b***, Native Binary Application 356c, respectively), which the server determines to generate the appropriate manifest file for that particular device. Ex-1005, Fig. 3 (green). For

Petitioner Disney Ex-1002, 0087

example, Lewis teaches that for the user using "client device 350b" (i.e., the identity of the second client), "platform rule set 322a may dictate that if a request ***originates from a client device indicating HLS or HTTP Live Streaming support***, then dynamic manifest file server 310 should preferably generate a M3U8 manifest file, or manifest file 357b, referencing MPEG transport stream video files, or processed video segments 315b." Ex-1005, 6:22-27. HLS Client 356b corresponds to the recited "second device type," which is "different from the first device type." Lewis thus teaches determining the second device type (i.e., HLS Client 356b) based on the identity of the first client (i.e., the user using client device 350b). Lewis expressly contrasts second-client (350b) behavior with first-client (350a) behavior in the same passage: when 350a requests, server 310 returns a Flash/Zeri (F4M) manifest for Flash Player plug-in 356a; when 350b requests, it returns an HLS (M3U8) manifest for HLS client 356b. Ex-1005, 6:6-27. A POSITA would have recognized that this difference in returned manifest formats follows from the server's rule-based classification of the requesting client, i.e., determining a second device type based on the identity of the second client, and that second device type (HLS) is different from the first (Flash). *See also* Ex-1005, 5:58-67 (client/device targeting), 8:42-52 (request parameters identifying client capabilities).

Petitioner Disney Ex-1002, 0088

Fig. 3



122. To the extent PO argues that Lewis's device type determination is not based on the client's identity, this feature is further taught by Barraclough.

123. Ground 2: Barraclough teaches this limitation for the reasons described in Section XI.A.2.d. Ex-1007, 16:26-31 ("the device classification is retrieved from data stored for a particular user (e.g., at the data storage arrangement 330)"), Fig. 3 (green).

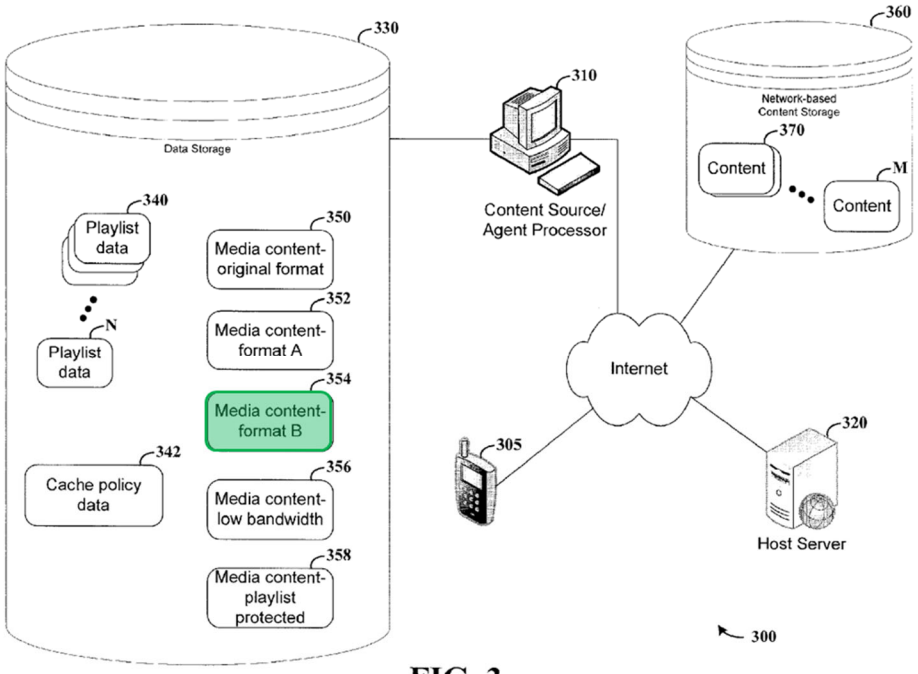


FIG. 3

124. In the Lewis-Barraclough combination, Lewis's dynamic manifest file server determines the second device type (i.e., HLS Client 356b) based on the particular client (i.e., user using second client device 350b), as taught by both Lewis's "user tracking profile" and Barraclough's "particular user" data storage.

125. As I also explained in the Technological Background section (*see* Section VIII.D), this was a known concept by 2010. *See, e.g.*, Ex-1022 (Zapata) ¶¶193, 199; Ex-1023 (Gopalakrishnan), 7:28-30, 14:14-19.

Petitioner Disney Ex-1002, 0090

j. Element 8[i]: based at least in part on the determination of the second device type, generating, with the processor, a second manifest having information for streaming the one or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and

126. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[i] for the reasons described in Section XI.A.2.e.

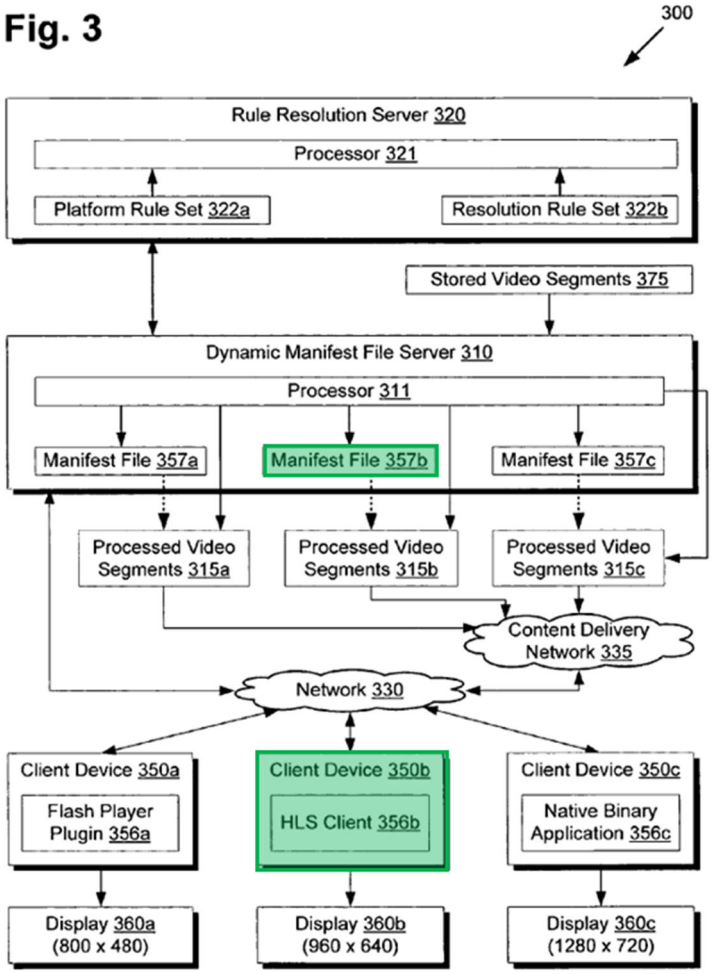
(1) Generating a second manifest with the processor based on determination of second device type, wherein content of the second manifest is different from content of the first manifest

127. Lewis teaches that "dynamic manifest file server 310 provides manifest files for a diverse range of client device platforms, including … HTTP Live Streaming client 356b at client device 350b." Ex-1005, 5:60-65, Fig. 3. For example, for client device 350b, "if a request ***originates from a client device indicating HLS or HTTP Live Streaming support***, then dynamic manifest file server 310 should preferably ***generate a M3U8 manifest file, or manifest file 357b***, referencing MPEG transport stream video files, or processed video segments 315b." Ex-1005, 6:22-27, Fig. 3 (green). Manifest file 357b corresponds to the recited "second manifest" based on the second device type, having different content from that of first manifest file 357a. Because Lewis generates F4M for a Flash-type first client (manifest 357a referencing F4F segments 315a) and M3U8 for an HLS-type second client (manifest

Petitioner Disney Ex-1002, 0091

357b referencing MPEG-TS segments 315b), the content of the second manifest necessarily differs from the first, both in (i) the manifest format (M3U8 vs. F4M) and (ii) the segment references (315b vs. 315a) selected for the second client's device type. Ex-1005, 6:5-27. Lewis also shows that each client's display/resolution rules (e.g., 800×480 for 350a vs. 960×640 for 350b) drive different processing outcomes, so the URI list and metadata in 357b will point to a different set of segments than 357a, further confirming different content. Ex-1005, 6:5-14, 6:28-31.

Fig. 3



Petitioner Disney Ex-1002, 0092

(2) Manifest has information for streaming one or more segments of the requested media source via the data communications network

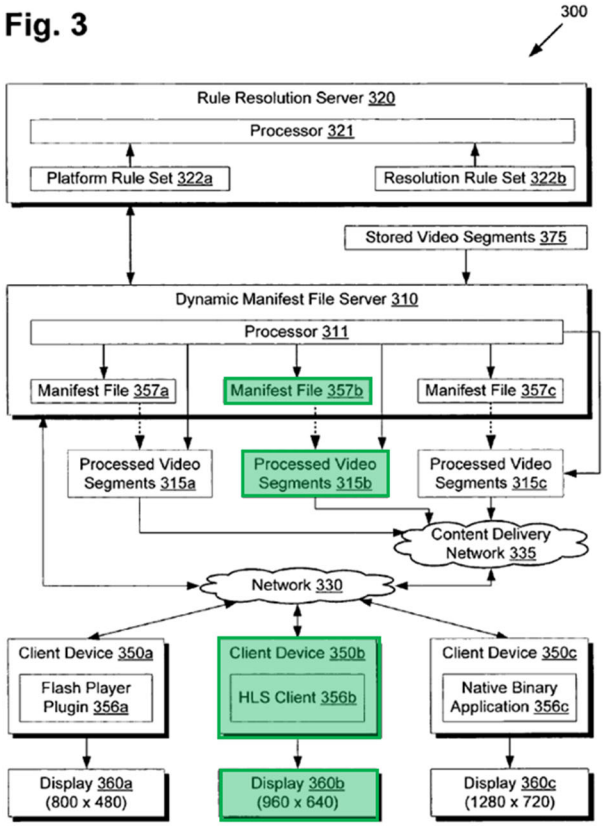
128. Lewis teaches that "***when client device 350b requests video content*** represented by stored video segments 375, dynamic manifest file ***server 310 may provide manifest file 357b in the HTTP Live Streaming format for facilitated playback*** by HLS client 356b." Ex-1005, 6:31-35. Lewis teaches that "HLS client 356b may then access the referenced video content ***in processed video segments 315b over network 330 via content delivery network 335***." Ex-1005, 6:35-38.

(3) In accordance with a second chunking protocol

129. Lewis teaches that the second manifest has information for streaming the requested media source segments in accordance with a second chunking protocol. Lewis teaches that for client device 350b, "[p]rocessed video ***segments*** 315b ***may be encoded*** from stored video segments 375, and may be ***resized and cropped*** according to resolution rule set 322b ***to optimally fit*** the 960 by 640 resolution provided by ***display 360b***." Ex-1005, 6:28-31, Fig. 3 (green). Dynamic manifest file server 310 then provides manifest file 357b "in the HTTP Live Streaming format for facilitated playback by HLS client 356b," which accesses "the referenced video content in processed video segments 315b over network 330 via content delivery network 335." Ex-1005, 6:31-38, Fig. 3 (green). HTTP Live Streaming ("HLS") was a well-known "chunking protocol" implemented by Apple

Petitioner Disney Ex-1002, 0093

by 2010 and, indeed, is the most common chunking protocol in streaming. *See, e.g.*, Ex-1043 (Begen), 59-60, 62-63. The HLS chunking protocol breaks video into small chunks (typically 2-10 seconds), with a playlist file (M3U8) that tells the player which chunks to request next and supports adaptive bitrate streaming so that the user can switch between different quality levels depending on network conditions. Lewis expressly teaches that its manifest file may comprise a M3U8 playlist file. Ex-1005, 3:46-48.



130. *See also id.*, 8:65-9:8 ("***a cutting or cropping processing procedure may be carried out*** with the affected video segments so that the generated manifest file can insert the content at any playback offset"). Lewis's HLS chunking protocol,

Petitioner Disney Ex-1002, 0094

when applied to the second manifest generated based on the determination of the second device type (i.e., HLS Client 356b), corresponds to the recited "second chunking protocol." Moreover, as explained in Section XI.A.2.e, Lewis teaches the "chunking protocol" limitation under either parties' construction. Lewis's rule-driven adaptations (resolution/framerate/region/ad-insertion) are executed before the manifest is returned and thus alter which segments are referenced for each client type; as a result, the second manifest (for the second device type) will list different segments and offsets than the first manifest, another reason its content is different. Ex-1005, 6:55-7:4, 7:23-26, 8:65-9:15.

131. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the Lewis-Barraclough combination (identity-based device-type → second manifest and protocol): Barraclough supplies the identity-based device classification (retrieved "from data stored for a particular user"), which, for a second client, can differ from the first and therefore select a different format/encoding. Ex-1007, 16:22-31. Lewis then uses that second device type to generate a second manifest (e.g., M3U8 for HLS) that references the appropriate segmented assets (315b) for delivery "over network 330 via CDN 335." Ex-1005, 6:31-38. This identity-driven classification → protocol/manifest path yields a second chunking protocol and ensures the second manifest's content differs from the first manifest's

Petitioner Disney Ex-1002, 0095

(format, segment list, and processing outcomes). Ex-1007, 16:22-33; Ex-1005, 6:5-38.

132. As I also explained in the Technological Background section (*see* Section VIII.E), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 4:31-35; Ex-1032 (Johnson) ¶¶36, 38, 57; Ex-1030 (Guzman) at 41, 33; Ex-1023 (Gopalakrishnan), Abstract, 27:46-48, 23:15-27; Ex-1029 (Strasman) ¶¶9, 77, Cl. 19; Ex-1043 (Begen), 59-60, 62-63; Ex-1033 (MSDN Magazine) at 35; Ex-1034 (Sood), 3:40-43, 4:7-8; Ex-1039 (Gabin), 134-137; Ex-1040 (Carmel), 2:4-6, 2:22-28, 7:23-25; Ex-1041 (Brueck), 3:36-39, 7:62-8:5; Ex-1042 (Bocharov), Fig. 5, ¶14.

k. Element 8[j]: sending the second manifest to the second client.

133. In my opinion, Lewis or Lewis-Barraclough teaches Element 8[j] for the reasons described in Section XI.A.2.f.

134. Lewis teaches that "when client device 350b requests video content represented by stored video segments 375, dynamic manifest file ***server 310 may provide manifest file 357b in the HTTP Live Streaming format for facilitated playback*** by HLS client 356b." Ex-1005, 6:31-35; *see also id.*, 3:16-41 ("dynamic manifest file server 110 uses rule resolution server 120 to "create a dynamically tailored manifest file," "which may then be passed back to client device 150 over network 130 and placed into memory 155 as manifest file 157").

Petitioner Disney Ex-1002, 0096

135. As I also explained in the Technological Background section (*see* Section VIII.F), this was a known concept by 2010. *See, e.g.*, Ex-1023 (Gopalakrishnan), 2:15-19; Ex-1029 (Strasman) ¶14.

136. Thus, in my opinion, Lewis or Lewis-Barraclough renders obvious Claim 8.

3. Dependent Claim 13

a. Element 13[pre]: The method of streaming media via the data communications network as recited in claim 8, further comprising:

137. In my opinion, Lewis or Lewis-Barraclough teaches the preamble. *See* Section XI.A.2.a.

b. Element 13[a]: receiving data regarding the requested media source, wherein the data includes information indicative of at least one point, during playback of the requested media source, at which at least one other media source is to be shown;

138. In my opinion, Lewis or Lewis-Barraclough teaches Element 13[a].

139. Lewis teaches receiving data regarding the requested media source at the client device. This data is what is ultimately included in the manifest file, discussed above. Lewis teaches that dynamic manifest file server 110 forwards "parameters received from the request originating from client device 150 to rule resolution server 120" to "create a dynamically tailored manifest file accordingly, which may then be passed back to client device 150" as a "manifest file." Ex-1005, 3:30-41. The manifest file references "live video segments … on servers hosted in

Petitioner Disney Ex-1002, 0097

content delivery network 135." Ex-1005, 3:44-46; *see also id.*, 3:46-48 (manifest file "may comprise a playlist file such as a M3U8 playlist file"). The manifest file, which is received by the client device, includes the recited data received "regarding the requested media source."

140. To the extent PO argues that the "receiving" step refers to data received ***at the server*** regarding the requested media source, this is also taught by Lewis. Lewis teaches that producer 180 records "live footage … to be streamed to users," which is sent "to live video encoder 170, which may then encode the live video footage in real-time into live video segments 175." Ex-1005, 2:59-64. "As new live footage is captured and approved, new video segments may be encoded and stored within live video segments 175, which may then be distributed over content delivery network 135." Ex-1005, 3:1-4. Content delivery network 135 comprises "a network of servers for storage and distribution of media content." Ex-1005, 3:4-6. Lewis further teaches that live video encoder 170 provides "rule resolution server 120 with an updated list of references for accessing live video segments 175 through content delivery network 135, which may then be passed to dynamic manifest file server 110 for generating manifest file 157." Ex-1005, 3:6-11. Therefore, the servers receive data (e.g., rules for access and generation of the manifest file) regarding the requested media source (i.e., live video segments).

Petitioner Disney Ex-1002, 0098

141. Lewis also teaches that data in the manifest file includes information indicative of at least one point during playback of the requested media source at which at least one other media source is to be shown. Lewis teaches that "[i]n addition to preparing video content for optimal presentation quality, ***substantive content such as advertising content may also be targeted and prepared*** based on various parameterized rules evaluated by rule resolution server 320." Ex-1005, 7:10-13. The user's tracking profile for client device 350a, for example, "may be analyzed to formulate a targeted advertising campaign for manifest file 357a, ***adding appropriate pre-roll, post-roll, and mid-roll advertising content*** estimated to be most relevant and interesting for the user of client device 350a." Ex-1005, 7:13-19. Lewis also teaches that "[p]roducer 180 may provide specific rules to rule resolution server 120 to customize the generation of manifest file 157, such ***a rule to insert advertising segments to overwrite a particular time block*** in the live stream." Ex-1005, 3:11-15. The "advertising content" corresponds to the recited "other media source" and the associated "time blocks" correspond to the claimed "at least one point, during playback of the requested media source, at which at least one other media source is to be shown." Data in the manifest file received at the client device and/or in the rules provided to the server thus includes information indicative of at least one point during playback of the requested media source (i.e., live video segments) at which the other media source (i.e., advertising content) is to be shown.

Petitioner Disney Ex-1002, 0099

Ex-1005, 3:44-46. Lewis expressly provides concrete time-point examples and mechanisms: if ad insertion is requested at 0:00:13, the system "may generate processed video segments … so that the generated manifest file can insert the content at any playback offset," and the manifest is generated to reference the ad segments at that specific playback offset. Lewis also describes scheduled global insertions where administrators "modify … manifest files … to reference [a] high priority video segment" at a set clock time (e.g., 8:00 PM PST) for all clients—another instance of data indicating a point during playback when another media source is to be shown. During live streams, new rules may be "continually generate[d] … to queue advertising blocks at specific times during the live stream," and updated manifests are issued accordingly, reinforcing that the received data includes explicit time points for showing other media.

c. Element 13[b]: receiving information regarding the at least one other media source; and

142. In my opinion, Lewis or Lewis-Barraclough teaches Element 13[b].

143. Lewis teaches receiving information regarding the other media source (i.e., ad video segments) at the client device. The manifest file received by the client references "live video segments and ad video segments on servers." Ex-1005, 3:44-46; *see also id.*, 3:46-48. This information is also received at the server. Lewis teaches that producer 180 dynamically adds "new rules to rule resolution server 120, for example to schedule advertising blocks" and rule resolution server 120 then

Petitioner Disney Ex-1002, 0100

directs "dynamic manifest file server 110 to ***switch to advertising content at specific time blocks*** specified by the new rules added by producer 180." Ex-1005, 3:55-60. For example, "producer 180 may provide a rule to rule resolution server 120 specifying a time block to ***switch to a commercial***, such as ***thirteen (13) seconds into the live broadcast***." Ex-1005, 4:52-55. The timing information for switching to ad video segments received by the client device in the manifest file and at the server from producer 180 corresponds to the recited "information" received "regarding the at least one other media source." Lewis further teaches that the ad-insertion rules themselves identify which advertisements to use ("specific advertisements") and when to show them ("time blocks"), and that these rules are evaluated by the rule resolution server when generating the manifest. Ex-1005, 8:52-56. In practical terms, the dynamic manifest file server cannot reference ad video segments in the manifest unless it has received (from the rule resolution server/producer) information about those ad assets, e.g., identifiers/URIs, campaign selection rules, and scheduled offsets, thereby satisfying "receiving information regarding the at least one other media source." See Ex-1005, 3:55-60, 4:52-55, 8:52-56. Lewis also discloses administrator- or producer-initiated updates that cause the manifest to reference a high-priority or substitute segment at a specified clock time (e.g., a scheduled insertion across clients), which is likewise information received about another media source to be shown during playback. *See, e.g.*, Ex-1005, 3:55-60, 8:65-9:15.

Petitioner Disney Ex-1002, 0101

d. Element 13[c]: including, in either or both of the first manifest and the second manifest, information for streaming one or more segments of the at least one other media source at the at least one point during the playback of the requested media source.

144. In my opinion, Lewis or Lewis-Barraclough teaches Element 13[c].

145. Lewis teaches including in the manifest, information for streaming the one or more segments of the at least one other media source (i.e., ad video segments) at the at least one point during the playback of the requested media source (i.e., live video segments). For example, as shown in Figure 2 below, "[a]ssuming a rule specifying ***commercial insertion at thirteen seconds for ad video segments 245 running at thirty seconds***, [processed video] segments 216b and 216e" may be "created from [live video] segments 276b and 276e, respectively." Ex-1005, 5:2-5, Fig. 2. Specifically, "segment 216b may comprise the first three seconds of [live video] segment 276b, and segment 216e may comprise the last seven seconds of [live video] segment 276e … allowing advertisement or other content insertion at any desired playback offset or position without dead time." Ex-1005, 5:5-11. In Figure 2's example, dynamic manifest file server 210 thus generates "manifest file 257, with entry 258a referencing original [live video] segment 276a, entry 258b referencing processed segment 216b, entries 258c through 258e referencing [ad video] segments 246a through 246c respectively, and entry 258f referencing processed segment 216e." Ex-1005, 5:16-21. Entries 258a through 258f are therefore

Petitioner Disney Ex-1002, 0102

"played back successively without gaps by a media player application," resulting in "an advertisement smoothly integrated into a live stream with no interruptions or dead time." Ex-1005, 5:21-24. Lewis further teaches that the manifest file "may be provided as a M3U8 file, and each of the referenced [live and ad] video segments may comprise a MPEG-2 transport stream (.TS) file," such that "media player application 156 can download the referenced segments" and output them "to display 160 for viewing by user 185." Ex-1005, 9:46-62. Lewis's manifest thus includes information for streaming the ad video segments at the at least one point during the playback of the live video segments. In Lewis, the inclusion of entries 258c–258e that reference the ad video segments (e.g., 246a–246c) is itself the "information for streaming" those other-media segments, the manifest lists the segment identifiers/URIs and their order at the precise playback point created by the cut segments 216b and 216e. Ex-1005, 5:2-24. Because Lewis generates different manifests for different device types (e.g., F4M/F4F for Flash and M3U8/TS for HLS), either or both of the first and second manifests can include these ad-segment references at the specified time point, satisfying the claim's disjunctive "either or both" requirement. Ex-1005, 6:5-38, 9:46-62. Lewis also explains that ad insertion can be pre-roll, mid-roll, or post-roll, and that rule evaluation determines which ad segments appear and where they appear in the returned manifest; the player then fetches those referenced segments over the network and plays them in sequence with

Petitioner Disney Ex-1002, 0103

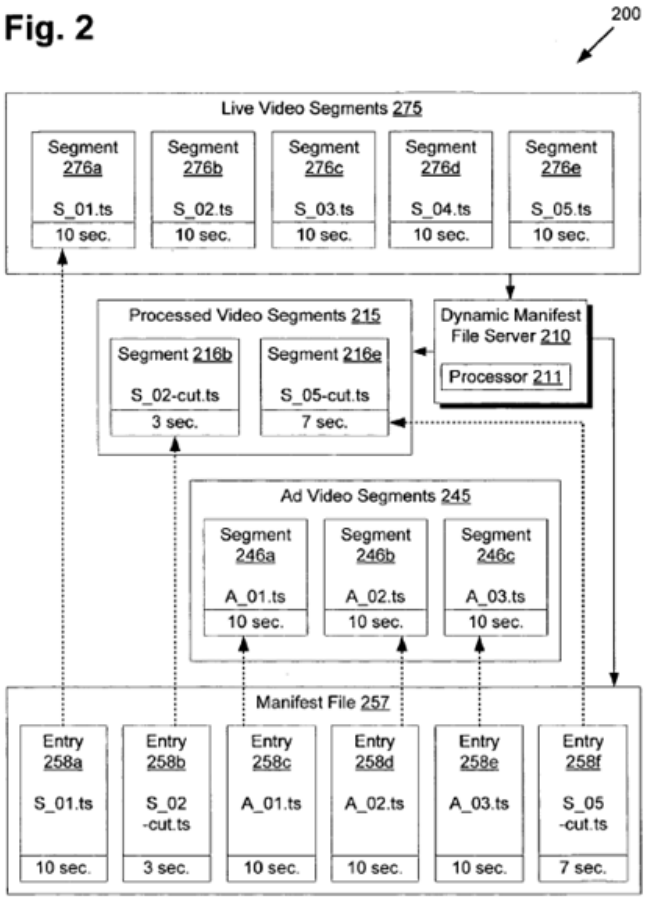
the program segments, as the claim requires. Ex-1005, 7:10-19, 8:52-56, 9:46-62. Lewis further teaches that when an insertion requires a specific playback offset, the system cuts or re-encodes the program segments so the generated manifest can reference the ad segments at that offset, again confirming that the manifest "includes information for streaming" the other media source at "the at least one point during playback." Ex-1005, 8:65-9:15.

146. While not necessary for this limitation, which is clearly taught by Lewis alone, I note that in the combined Lewis-Barraclough system, Barraclough's host server maintains and serves content items from network-based storage and processes playlist/media requests from different clients; Lewis's dynamic manifest server uses the ad-insertion rules and available ad assets to emit per-client manifests that explicitly reference the ad segments at the specified time points. Thus, the resulting first and/or second manifest includes the streaming information (segment references and ordering) for the "other media source" within the playback of the requested program. Ex-1007, 16:20-33; Ex-1005, 5:2-24, 7:10-19, 8:52-9:15, 9:46-62.

147. As I also explained in the Technological Background section (*see* Section VIII.G), this was a known concept by 2010. *See, e.g.*, Ex-1034 (Sood), 10:28-39; Ex-1028 (Mao) ¶22; Ex-1035 (Smooth Streaming White Paper), 23.

Petitioner Disney Ex-1002, 0104

Fig. 2



B. Grounds 3-4: Claims 14-15 are obvious over Lewis in view of Jain or Lewis-Barraclough in view of Jain.

1. A POSITA would have been motivated to combine Lewis's or Lewis-Barraclough's teachings with Jain's teachings and would have had a reasonable expectation of success.

148. A POSITA would have been motivated to combine the teachings of Lewis and Jain or Lewis-Barraclough and Jain and had a reasonable expectation of success in doing so because each relates to methods for streaming media content in response to a request and the control/signaling of time-based events during playback. *See* Section XI.A.1; Ex-1008 ¶¶22, 53, 59.

Petitioner Disney Ex-1002, 0105

149. Lewis teaches that "producer 180 may provide a rule to rule resolution server 120 specifying a time block to ***switch to a commercial***, such as ***thirteen (13) seconds into the live broadcast***." Ex-1005, 4:52-55. Moreover, Lewis teaches that producer 180 generates the "insertion rules in real-time for rule resolution server 120, allowing the dynamic insertion of advertisement time blocks or other content while a live event is in progress." Ex-1005, 8:57-60. Additionally, Lewis teaches that "if any of the rules require the insertion of content ***at a time offset starting or ending within a video segment***, then a cutting or cropping processing procedure may be carried out with the affected video segments so that the generated manifest file can insert the content at any playback offset." Ex-1005, 8:65-9:3. Lewis leaves the implementation details for how to indicate that an advertisement will be played or that a certain point has been reached in playback of the requested media source in the manifest file to a POSITA.

150. Jain teaches "a combination of audio beaconing and audio matching and its application to track a video on a content site, such as Hulu.com." Ex-1008 ¶32. As shown in Figure 2 below, Jain teaches activities 201, which "shows actions taken in user system 150 where a video is loaded in the user device 130, and played." Ex-1008 ¶32. "At the 5 minute mark (301 sec.), ***a 15 second advertisement is served***" and at the end "of the advertisement (316 sec.), the video continues to play until its conclusion (600 sec.)." Ex-1008 ¶32. Jain also teaches "additional beacon

Petitioner Disney Ex-1002, 0106

events" 203, where "additional data may be transmitted in the beacon to include URLs and video ID's when a video is loaded and played." Ex-1008 ¶33. Jain teaches that "[a]s the advertisement is served, an event beacon, which may include advertisement URL data, is transmitted." Ex-1008 ¶33. Jain also teaches that the beacon "transmits a ***timestamp***" and "[a]t the conclusion of the video, ***a video end beacon is transmitted to indicate the user/panelist is no longer viewing specific media***." Ex-1008 ¶33.

200	Time →	0-3 sec	4 sec ---------- 180 ---------- 300 sec				301-315 sec	316—599 sec	600 sec
201	Activities→	Video Loaded	Video Played				Ad served	Video continues	Video End
202	Audio Beacon detected by portable device →	Continuous Audio representing the website (e.g., Hulu.com)							
203	Additional beacon events →	URLs (Hulu.com & Video ID)					AD Event Beacon (co-relation URL)		Video End Beacon
204	Audio matching events detected by portable device →		Audio Signature Sample 1	Audio Signature Sample 2	Audio Signature Sample 3 (Content Match Achieved)	Same Signature Sample 4 (No Change)	Audio Signature Sample 5 (Different Content couldn't match)	Audio Signature Sample 5 (Same Video again)	No Signature (But Can't assume video is completed
205	Overlay portable device data	X			X		X	X	X
206	End- User Experience	URL of site where the video was played			Content Matched		Ad Served		Video Played till the End

FIG. 2

151. A POSITA would have been motivated to indicate in Lewis's manifest file that an advertisement will be played or that a certain point has been reached in playback of the requested media source, using a beaconing URL, as taught by Jain, because beacons were a well-known and cost-effective customer engagement tool

Petitioner Disney Ex-1002, 0107

for signaling or tracking the playback of an advertisement or requested media source. For example, using beacons in a media stream are particularly beneficial for advertising because they allow businesses to target consumers nearby in real-time and create hyper-personalized experiences based on the data from the beacons. Additionally, given that Lewis already teaches that the manifest file comprises URLs (*see, e.g.*, Ex-1005, 7:31-35), it would have been obvious to implement a beacon for signaling or tracking the playback of an advertisement or requested media source in the form of a URL in Lewis's manifest, as taught by Jain.

152. In the Lewis-Jain (Ground 3) or Lewis-Barraclough-Jain (Ground 4) combination, either or both of Lewis's first and second manifest would include a beaconing URL which, when used, indicates playback of an advertisement or that a certain point has been reached in playback of the requested media source, such as the start or end of the video, as taught by Jain. A POSITA would have understood that, in the Lewis-and-Jain combination, the straightforward implementation is to include one or more beaconing URL entries in the manifest aligned with the rule-defined cut/crop offsets; when the player reaches those offsets and dereferences the beacon URL, that network request (which Jain teaches may include an advertisement URL and a timestamp) is itself indicative that the specified playback point has been reached. *See* Ex-1005, 7:31–35, 8:65–9:3; Ex-1008 ¶¶26, 32–33. Likewise, in the Lewis-and-Barraclough-and-Jain combination, the host/playlist server would

Petitioner Disney Ex-1002, 0108

provide the same beaconing URL entries alongside the content/ad entries so the client request occurs precisely at the interval boundaries (e.g., ad start/end or program end), evidencing arrival at that point without altering media formats or standard HTTP streaming. See Ex-1008 ¶¶26, 33. A POSITA would also recognize that Jain's periodic transmissions ("every x seconds") supply additional intermediate playback waypoints, if desired, that can be mapped to manifest/playlist entries. *See* Ex-1008 ¶26.

153. In addition to the reasons provided above, a POSITA would have been motivated to combine Lewis-Jain because they have complementary aims and relate to the same stage of the pipeline. Specifically, Lewis's server builds manifests that pick which segments to deliver (and when), including ad segments; Jain teaches server-side insertion of measurement beacons into those very audio streams, including on-the-fly during transcoding. Integrating beaconing as taught by Jain into Lewis's packaging/transcoding chain is therefore a straightforward, same-stage insertion.

154. A POSITA would also have had a strong commercial/technical motivation to combine Lewis-Jain. Specifically, Lewis enables targeted/ad-supported streaming (rules for ad insertion, per-device targeting, CDN steering). A natural next step is measuring whether those targeted ads/content were actually played, precisely the cross-platform measurement problem Jain identifies

Petitioner Disney Ex-1002, 0109

and solves. Thus, Lewis's delivery features create the need, and Jain supplies the tool to meet that need.

155. Additionally, Lewis and Jain are compatible and the combination achieves predictable results: Lewis relies on "widely supported" streaming protocols and standard segment formats (HLS/F4M), and Jain teaches that beaconing can be integrated with transcoding and synchronized with video, i.e., nothing about manifests prevents adding an audio watermark. The expected result of the combination (i.e., instrumented playback measurable by external devices) follows predictably.

156. Moreover, implementing Jain's teachings in Lewis's system would have been routine for a POSITA. Lewis already cuts/switches segments server-side (e.g., frame-accurate ad insertion). A POSITA would have been able to apply Jain's beacon encoder to (a) the audio of ad segments and/or (b) program segments selected by the manifest, either at ingest or on-the-fly(both paths are expressly contemplated in Jain. Therefore, the Lewis-Jain combination simply adds known, server-side, stream-compatible measurement beacons to a known, server-side manifest/segment pipeline to satisfy an acknowledged industry need; the result is predictable and achievable with routine engineering.

157. Likewise, a POSITA would have understood that the Lewis-Barraclough-Jain combination would provide a unified architecture.

Petitioner Disney Ex-1002, 0110

Specifically,playlist-driven cloud delivery and manifest-driven adaptation: Barraclough's host/server model authenticates playlist requests and streams from network storage, with the system selecting formats/bitrates for the remote device. Lewis's rule-based manifests likewise pick the wire format (HLS/F4M) and bitrate per device/network. A POSITA would have viewed these as complementary mechanisms within the same OTT pipeline (playlist selection → server packaging/manifest → segment delivery).

158. Moreover, the combination provides an obvious place to add measurement: Jain's beacons are designed specifically for cross-platform streaming and are taught to be inserted server-side during transcoding, the very operation Barraclough uses when configuring bitrate/format for a remote device. Adding Jain's encoding step into Barraclough's (and/or Lewis's) packaging/transcoding workflow meets the common industry need to measure remote, playlist-initiated playback across devices.

159. Furthermore, the combination allows for predictable operation across device/bitrate variations: Barraclough anticipates delivering different formats/bitrates for the same title; Jain anticipates this too and explains how to encode in a way that fits into transcoding workflows and remains properly synchronized with video, yielding predictable measurement regardless of the ultimate bitrate/format delivered.

Petitioner Disney Ex-1002, 0111

160. Finally, a POSITA would have understood that the combination would provide flexible deployment: Barraclough expressly allows functions (server, cache, playlist manager) to be implemented in different arrangements. That flexibility invites adding a beacon-encoding module at the host or content site, as Jain proposes via API and server-side integration. In other words, the playlist/cloud-delivery system that configures format/bitrate for remote devices (Barraclough) and the rule-driven manifest/segment pipeline (Lewis) are natural hosts for Jain's server-side beaconing. Integrating them fulfills a recognized cross-platform measurement need with routine, predictable modifications.

161. As I also explained in the Technological Background section (*see* Sections VIII.H and VIII.I), these were known concepts by 2010. *See, e.g.*, Ex-1034 (Sood), 4:4-8, Cl. 7, 10:30-39; Ex-1031 (Luby), 18:45-65, 37:64-38:4, 41:49-60; Ex-1038 (Krishnan) ¶¶35-37, 41, 46, 48; Ex-1037 (Bailey), 1:56-2:8, 4:65-5:17; Ex-1036 (Christian) ¶¶20, 25-27; Ex-1030 (Guzman), 33; Ex-1029 (Strasman) ¶122, Fig. 7.

162. A POSITA would have had a reasonable expectation of success in combining the teachings of Lewis or Lewis-Barraclough and Jain. All three describe methods for streaming media content in response to a request. *See* Section XI.A.1; Ex-1008 ¶¶22, 53, 59. Additionally, both Lewis and Jain teach methods for inserting advertisements into a media stream. Ex-1005, 8:52-56; Ex-1008 ¶¶32-33. A

Petitioner Disney Ex-1002, 0112

POSITA would thus have understood that these references disclose interrelated teachings based on well-understood technologies that would have been amenable to various well-understood and predictable combinations.

2. Dependent Claims 14-15

a. Claim 14: The method of streaming media via the data communications network as recited in claim 13, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of playback of at least one advertisement.

163. In my opinion, Lewis in view of Jain ("Lewis-Jain") or Lewis-Barraclough in view of Jain ("Lewis-Barraclough-Jain") teaches Claim 14.

164. As explained in Section XI.A.3, Lewis teaches that "producer 180 may provide a rule to rule resolution server 120 specifying a time block to ***switch to a commercial***, such as ***thirteen (13) seconds into the live broadcast***." Ex-1005, 4:52-55. Moreover, Lewis teaches that producer 180 generates the "insertion rules in real-time for rule resolution server 120, allowing the dynamic insertion of advertisement time blocks or other content while a live event is in progress." Ex-1005, 8:57-60. Lewis leaves the implementation details for how to indicate that an advertisement will be played in the manifest file to a POSITA. It would have been an obvious design choice to use a beaconing URL to indicate playback of the advertisement, as

Petitioner Disney Ex-1002, 0113

taught by Jain.[4] A POSITA would have implemented the "beaconing URL" as an additional manifest (or playlist) entry aligned to the cut/crop points that Lewis generates for ad insertions, so that the client issues a standard HTTP request exactly at ad start or ad end; because Lewis expressly teaches URL-based manifests and frame-accurate ad placement, inserting a URL whose retrieval signals ad playback is a routine, implementation-neutral choice. (*See, e.g.*, manifest "comprises URLs" and cutting/cropping so inserted content lands at any playback offset.). Jain then supplies the ad-specific signaling content, an "event beacon" transmitted as the advertisement is served, so the manifest's added beacon URL call is "indicative of playback of at least one advertisement" as required. This is consistent with standard adaptive-streaming practice in which manifests enumerate URL requests that drive time-aligned actions during playback.

165. As shown in Figure 2 below, Jain teaches "a combination of audio beaconing and audio matching and its application to track a video on a content site, such as Hulu.com." Ex-1008 ¶32. Timeline 200 shows "a scenario where a user/panelist plays a ten minute video on Hulu.com." Ex-1008 ¶32. Activities 201 shows actions taken by the "user system 150 where a video is loaded in the user

[4] The term "beaconing URL" only appears in the claims of the '639 Patent. The patent states that "[a]dvertising breaks can be determined by the media provider(s) 130 and indicated by 'cue tones,' which are analog or digital signals which indicate in a first form that a break is about to begin" and "in a second form that the break is about to end." Ex-1001, 9:63-67.

Petitioner Disney Ex-1002, 0114

device 130, and played," "[a]t the 5 minute mark (301 sec.), a 15 second advertisement is served," and at the end "of the advertisement (316 sec.), the video continues to play until its conclusion (600 sec.)." Ex-1008 ¶32. Jain also teaches "additional beacon events" 203, where "additional data may be transmitted in the beacon to include URLs and video ID's when a video is loaded and played." Ex-1008 ¶33, Fig. 2 (yellow). Jain teaches that "[a]s the advertisement is served, an event beacon, which may include advertisement URL data, is transmitted." Ex-1008 ¶33.

200	Time →	0-3 sec	4 sec ---------- 180 ---------- 300 sec				301-315 sec	316—599 sec	600 sec
201	Activities→	Video Loaded	Video Played				Ad served	Video continues	Video End
202	Audio Beacon detected by portable device →	Continuous Audio representing the website (e.g., Hulu.com)							
203	Additional beacon events →	URLs (Hulu.com & Video ID)					AD Event Beacon (co-relation URL)		Video End Beacon
204	Audio matching events detected by portable device →		Audio Signature Sample 1	Audio Signature Sample 2	Audio Signature Sample 3 (Content Match Achieved)	Same Signature Sample 4 (No Change)	Audio Signature Sample 5 (Different Content couldn't match)	Audio Signature Sample 5 (Same Video again)	No Signature (But Can't assume video is completed
205	Overlay portable device data	X			X		X	X	X
206	End- User Experience	URL of site where the video was played			Content Matched		Ad Served		Video Played till the End

FIG. 2

166. In the Lewis-Jain (Ground 3) or Lewis-Barraclough-Jain (Ground 4) combination, either or both of Lewis's first and second manifest would include a

Petitioner Disney Ex-1002, 0115

beaconing URL which, when used, indicates playback of an advertisement, as taught by Jain.

167. As I also explained in the Technological Background section (*see* Section VIII.H), this was a known concept by 2010. *See, e.g.*, Ex-1034 (Sood), 10:30-39; Ex-1031 (Luby), 37:64-38:4, 41:49-60; Ex-1038 (Krishnan) ¶¶35-37, 41, 46, 48; Ex-1037 (Bailey), 1:56-2:8; Ex-1036 (Christian) ¶¶20, 25-27.

b. Claim 15: The method of streaming media via the data communications network as recited in claim 8, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of a certain point has been reached in playback of the requested media source.

168. In my opinion, Lewis-Jain or Lewis-Barraclough-Jain teaches Claim 15.

169. Lewis teaches this claim for the reasons explained in Section XI.B.2.a. Additionally, Lewis teaches that "if any of the rules require the insertion of content ***at a time offset starting or ending within a video segment***, then a cutting or cropping processing procedure may be carried out with the affected video segments so that the generated manifest file can insert the content at any playback offset." Ex-1005, 8:65-9:3. Lewis leaves the implementation details for how to indicate that a certain point has been reached in playback of the requested media source in the manifest file to a POSITA. It would have been an obvious design choice to use a beaconing URL for this purpose, as taught by Jain. In the Lewis-Jain or Lewis-Barraclough-

Petitioner Disney Ex-1002, 0116

Jain combination, a POSITA would have aligned the beaconing URL with the cut/crop boundaries that Lewis creates specifically so the manifest can insert content at "any playback offset" (Ex-1005, 8:65–9:3), guaranteeing that the beacon fires exactly at the offset. Because Lewis's manifests "comprise URLs" (Ex-1005, 7:31–35), issuing a lightweight HTTP GET to a beacon endpoint at that offset is a routine, implementation-neutral choice requiring no change to media formats. Jain further teaches beacons at load, during playback (including periodic transmissions "every x seconds"), and at end-of-video, each with timestamps (Ex-1008 ¶¶26, 29, 33), providing canonical playback points (start, intermediate milestones, and end) readily mapped to manifest/playlist entries in Lewis-Jain or Lewis-Barraclough-Jain.

170. In addition to the teachings described in Section XI.B.2.a, Jain teaches that "video beacons can be embedded within an action script that will be running within the video Flash Player's run time environment on web page 110," such that when the page is accessed and the script is activated, the video beacon is triggered "which extracts and store[s] URL information on a server (e.g., content server 100), and launches the video Flash Player." Ex-1008 ¶26. When the beacon is triggered, it "may access AM server 120 to load a pre-recorded audio file containing a special embedded compatible code (e.g., CBET)," which is "utilized for beacon 131 ***to transmit for a given period of time (e.g., every x seconds)***." Ex-1008 ¶26. Jain further teaches that "[t]he web activity monitoring application collects web activities

Petitioner Disney Ex-1002, 0117

data from the user device 130 (e.g., site ID, video page URL, video file URL, ***start and end timestamp*** and any additional metadata about videosite information, URL information, time, etc.)." Ex-1008 ¶29. Moreover, Jain teaches that the "beacon also transmits a ***timestamp***" and "[a]t the conclusion of the video, ***a video end beacon is transmitted to indicate the user/panelist is no longer viewing specific media***." Ex-1008 ¶33; *see also id.*, ¶¶20, 23 (API communicates "timestamp"), Fig. 2 (yellow).

200	Time →	0-3 sec	4 sec ---------- 180 ---------- 300 sec				301-315 sec	316—599 sec	600 sec
201	Activities→	Video Loaded	Video Played				Ad served	Video continues	Video End
202	Audio Beacon detected by portable device →	Continuous Audio representing the website (e.g., Hulu.com)							
203	Additional beacon events →	URLs (Hulu.com & Video ID)					AD Event Beacon (co-relation URL)		Video End Beacon
204	Audio matching events detected by portable device →		Audio Signature Sample 1	Audio Signature Sample 2	Audio Signature Sample 3 (Content Match Achieved)	Same Signature Sample 4 (No Change)	Audio Signature Sample 5 (Different Content couldn't match)	Audio Signature Sample 5 (Same Video again)	No Signature (But Can't assume video is completed
205	Overlay portable device data	X			X		X	X	X
206	End- User Experience	URL of site where the video was played			Content Matched		Ad Served		Video Played till the End

FIG. 2

171. In the Lewis-Jain (Ground 3) or Lewis-Barraclough-Jain (Ground 4) combination, either or both of Lewis's first and second manifest would include a beaconing URL which, when used, indicates that a certain point has been reached in playback of the requested media source, such as the start or end of the video, as taught by Jain.

Petitioner Disney Ex-1002, 0118

172. As I also explained in the Technological Background section (*see* Section VIII.I), this was a known concept by 2010. *See, e.g.*, Ex-1030 (Guzman), 33; Ex-1029 (Strasman) ¶122, Fig. 7; Ex-1034 (Sood), 4:4-8, Cl. 7; Ex-1031 (Luby), 18:45-65; Ex-1037 (Bailey), 4:65-5:17.

C. Grounds 5-6: Claim 8 is obvious over Pyle alone or Pyle in view of Barraclough.

1. A POSITA would have been motivated to combine Pyle's teachings with Barraclough's teachings and would have had a reasonable expectation of success.

173. A POSITA would have been motivated to combine the teachings of Pyle and Barraclough and had a reasonable expectation of success in doing so because each relates to methods for streaming media content in response to a request based on, for example, characteristics and/or capabilities of the requesting device. *See* Section XI.A.1; Ex-1006, 10:22-56.

174. Pyle teaches that if the requested content is a movie, a requesting device can be "***a HD television*** with very large form factor," which "can further be configured to only present media that is rated PG-13 (unless parent codes are entered) and is coupled to a reliable network with high bandwidth in an English-speaking household." Ex-1006, 10:27-33. As a second example, a requesting device "can be ***a smart phone*** with a more modest form factor and resolution support, and is currently in transit at high speeds resulting in a less reliable network connection" where "the default language … is set to German," and "user preference indicates

Petitioner Disney Ex-1002, 0119

hearing impairment." Ex-1006, 10:33-39, 10:53-56. Pyle teaches that a "composition component" examines data included in the request and determines "a suitable or optimal manifest" to "select and transmit … to the requesting device." Ex-1006, 10:53-56. Pyle further teaches that "individual manifests 204 can be optimized for … ***client preferences***, and so forth." Ex-1006, 8:23-27. A POSITA would have understood that Pyle determines the identity of the client because Pyle customizes the manifest to the particular user's preferences. A POSITA would also have understood that Pyle determines the first device type (i.e., HD television) and second device type (i.e., smartphone) based on the identity of the first and second client, respectively, because Pyle teaches that the particular representation is selected based upon the "***capabilities, limitations, or a current state of the presentation device*** or an associated network, ***as well as settings or preferences input by users*** of the presentation device." Ex-1006, 11:63-12:3. Additionally, a POSITA would have understood that Pyle's "local configuration 602" and request "606," which can include "all or a portion" of that configuration, necessarily associate the device's class, capabilities, network state, and user-controlled settings with the particular client session, so that the selector first attributes those characteristics to that client and then determines the applicable device type before generating the manifest. Ex-1006, 12:10–23. A POSITA would further have understood from Pyle's disclosure that "individual manifests 204 can be optimized for … endpoint profiles or

Petitioner Disney Ex-1002, 0120

configurations" that different endpoint classes (e.g., television versus smartphone) will cause the system to output different manifests reflecting different track sets/bitrates and option flags, confirming that the device type for each identified client drives manifest generation. Ex-1006, 8:17–27, 12:63–13:2. And because Pyle expressly contemplates simultaneous requests from "multiple clients," a POSITA would have recognized that the system must distinguish one client from another and bind each request to its own profile/device state to apply the correct device-type-based rules. Ex-1006, 34:20–34, 34:41–43.

175. In addition to the teachings described in Section XI.A.1, Barraclough teaches that the user "provides authentication information, such as the user name and password … and in response the host ***authenticates the user as an authorized user***, and facilitates media playlist access." Ex-1007, 7:56-62.

176. Thus, in view of both Pyle and Barraclough, a POSITA would have been motivated to determine the device type based on the identity of the client. For example, Pyle's method teaches to use device classification based on the identity of the device, and Barraclough teaches complimentary information stored for the particular user of a particular device for the reasons described in Section XI.A.1. Barraclough would further Pyle's goal of optimizing manifests based on "a particular setting or reference or a particular set of settings or preferences" of the user. Ex-1006, 10:65-11:9. In the Pyle-Barraclough combination, Pyle's

Petitioner Disney Ex-1002, 0121

components determine the device type based on data stored for the particular client, as taught by Barraclough. Additionally, in the Pyle–Barraclough combination, Barraclough's authentication step provides the per-user identity that Pyle then uses to retrieve or infer the corresponding device class (e.g., TV vs. smartphone) and generate the client-specific manifest. Ex-1007, 7:56–62; Ex-1006, 10:53–56, 12:10–23. Moreover, in the Pyle–Barraclough combination, the information Pyle uses to select a manifest (device capabilities, language/rating/accessibility preferences, bandwidth conditions) is supplied and anchored by Barraclough's authenticated session and playlist context. Barraclough teaches that the user provides credentials; the host authenticates the user and then facilitates playlist access and delivery from network storage, classifying the requesting device and mapping it to appropriate encodings/bitrates for streaming. Feeding those authenticated, per user (and per device class) parameters into Pyle's selection logic is a straightforward integration that a POSITA would find natural.

177. In addition to the reasons described above, a POSITA would have been motivated to combine Pyle-Barraclough because they provide complementary solutions to address the same problem. Barraclough gives the service model and storage topology: a host server authenticates remote users, manages playlists, and delivers media from a network cache populated from a user's source; the cache stores multiple representations (e.g., original, "format A/B," low-bandwidth versions), and

Petitioner Disney Ex-1002, 0122

selection can depend on device class and link conditions. It separates playlist metadata from content, uses a cross-referencing file structure so the same title can appear under multiple playlists and multiple formats, and even describes mapping device classes to high/low bitrate encodings for delivery. Pyle provides the delivery logic and standard control plane: multiple manifests, track sets (selectable vs seamlessly switchable), independent track storage, and dynamic composition so clients and servers pick the best combination for device capability, DRM, language, bitrate, etc., without the combinatorial explosion of pre-muxed versions. It uses standard HTTP adaptive streaming concepts (e.g., MPD-like manifests, PIFF/ISO BMFF compatibility), time-based segment addressing, and a selection component/presentation component to retrieve the right representation and switch tracks seamlessly at playback. Barraclough's playlists and multi-format cache supply the what/where of content; Pyle supplies the how to choose and switch the right tracks/bitrates for each remote device at playback time.

178. Moreover, both references address an acknowledged need (heterogeneous devices & variable networks): Barraclough explicitly targets remote playback across diverse devices and links and therefore stores alternative formats/bitrates and even class-based high/low bitrate variants for each title; but it still relies on populating the cache with separate versions and then picking among them. A POSITA would have looked to Pyle's dynamic composition and track-set

Petitioner Disney Ex-1002, 0123

model to standardize and automate that selection at playback time, reducing duplication and enabling seamless switching when conditions change.

179. Moreover, the combination uses a known, standardized technique to improve a similar system. Specifically, Pyle teaches industry-standard HTTP streaming with manifest-described selectable/switchable track sets and independent track addressing (e.g., PIFF/ISO BMFF), specifically to avoid "combinatorial complexity." Swapping in Pyle's manifest/track-set selection for Barraclough's ad-hoc playlist-to-format mapping is a simple substitution of known elements to achieve predictable scalability and interoperability.

180. The combination also makes content selection "encoding-agnostic," matching Barraclough's architecture. Specifically, Barraclough already separates playlist data from content and proposes encoding-agnostic playlist references and cross-references (same title under multiple playlists and formats). Pyle's composition layer and object model of attributes (codec, resolution, language, DRM, etc.) drops directly into that separation, letting the host publish multiple manifests per playlist and the client pick an optimal representation automatically.

181. The combination also improves user experience with seamless adaptation. Specifically, Barraclough supports serving low-bandwidth versions and high/low bitrate selection; Pyle adds "switchable" track sets so the client can seamlessly switch bitrate/angle/subtitles mid-stream (e.g., when bandwidth dips)

Petitioner Disney Ex-1002, 0124

while keeping the playlist context. That is a straightforward enhancement of the exact capability Barraclough seeks (remote streaming under changing conditions).

182. Finally, the combination reduces server and storage costs while increasing catalog flexibility. Specifically, Pyle's dynamic composition lets publishers add or remove tracks (e.g., new audio languages or captions) simply by updating composition XML/manifests "without changing any audio or video files," which directly addresses Barraclough's scenario of cross-referencing titles across multiple playlists and formats in a shared cache. The combination thus yields fewer pre-muxed artifacts to store and manage while supporting more variants.

183. As I also explained in the Technological Background section (*see* Sections VIII.C and VIII.D), these were known concepts by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 12:15-21; Ex-1021 (Mehrotra) ¶29.

184. A POSITA would have had a reasonable expectation of success in combining the teachings of Pyle and Barraclough. Both describe methods for streaming media content in response to a request based on characteristics and/or capabilities of the requesting device in a similar manner such as by storing the media content in different formats corresponding to different device types and selecting a suitable format based on the requesting device type. *See* Section XI.A.1; Ex-1006, 10:22-56. A POSITA would thus have understood that these references disclose interrelated teachings based on well-understood technologies that would have been

Petitioner Disney Ex-1002, 0125

amenable to various well-understood and predictable combinations. Pyle provides the manifest/track set composition and request driven selection; Barraclough provides the authenticated playlist flow and device class–to–encoding mapping used at delivery. Integrating the two merely routes Barraclough's authenticated user/device parameters into Pyle's already taught selection step, an application of known components for their intended purpose with predictable results.

185. Moreover, both references provide a common client–server paradigm over packet networks because they both implement media delivery via a host/server, network storage, and remote packet-based devices over the Internet. Integrating Pyle's manifest/selection into Barraclough's host-and-cache pipeline is a modular software change that stays within the same architecture. They also both provide interoperable, well-documented formats and protocols. Specifically, Pyle expressly targets compatibility with MPEG-2 TS/fragmented MP4 (ISO BMFF/PIFF) and HTTP adaptive streaming manifests, while Barraclough streams common media types and already contemplates transcoding to device-appropriate formats/bitrates. A POSITA would have expected Pyle's manifest-driven addressing to reference the very cache files Barraclough stores. Additionally, aligned selection logic already present in both references. Specifically, Barraclough performs per-device/per-bandwidth selection (e.g., device classes mapped to high/low bitrates); Pyle formalizes that decision into track-set selection and seamless

Petitioner Disney Ex-1002, 0126

switching. Migrating to Pyle's mechanism is an incremental, predictable refactor of logic the POSITA already practices. Finally, the combination's playlist < > composition mapping is straightforward. Each Barraclough playlist would be treated as a Pyle composition; multiple manifests for that playlist (e.g., mobile, TV, family/uncut) would be published, and the host (Barraclough) would distribute the right manifest while the client (per Pyle) would fetch the segments and switch the tracks as needed in the combination. All of the required pieces (e.g., server, manifests, cache, client) are well known and would have been routine to integrate for a POSITA.

2. Independent Claim 8

a. Element 8[pre]: A method of streaming media via a data communications network, the method comprising:

186. In my opinion, Pyle or Pyle in view of Barraclough ("Pyle-Barraclough") teaches the preamble.

187. Pyle teaches "dynamic composition of media for streaming to consuming devices." Ex-1006, 1:14-15. Pyle also teaches "a network environment in which the communications network/bus 1240 is the Internet" and "the computing objects 1210, 1212, etc. can be Web servers with which the client computing objects or devices 1220, 1222, 1224, 1226, 1228, etc. communicate via any of a number of known protocols, such as the hypertext transfer protocol (HTTP)." Ex-1006, 34:47-

Petitioner Disney Ex-1002, 0127

55. As I also explained in the Technological Background section (*see* Section VIII.A), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), Abstract; Ex-1028 (Mao) at Figs. 1, 3; Ex-1023 (Gopalakrishnan), Abstract.

b. Element 8[a]: receiving a universal source locator (URL) in a first request from a first client, wherein the URL includes information indicative of a requested media source;

188. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[a].

189. Pyle teaches receiving a URL in a first request from a first client that indicates a requested media source. Specifically, Pyle teaches a method for "facilitating HTTP delivery of streaming content" (Ex-1006, 32:43-45), which allows "multiple clients to take advantage of the information-gathering capabilities of the server." Ex-1006, 34:41-43, Figs. 11-12. Pyle teaches "***computing objects or devices 1220***, 1222, 1224, 1226, 1228, etc." which "***can be thought of as clients***." Ex-1006. 34:21-22. Moreover, Pyle teaches that "a request for the content can be transmitted (e.g., by a client) or received (e.g., by a server)," which includes "a set of characteristics or attributes defining a suitable representation of the content." Ex-1006, 32:45-49. Pyle further teaches that requests include a URL with information indicative of a requested media source. For example, Pyle teaches that Media Presentation Description ("MPD") "information is constantly updated for the client to formulate the correct uniform resource locator (URL) to obtain the latest segment of content" such that "***the client can formulate time-based segment request URLs***."

Petitioner Disney Ex-1002, 0128

Ex-1006, 5:4-7, 5:21-22. The URLs specify or describe the "***locations***" of the content 206 "included in one or more manifest 204" (Ex-1006, 11:33-36) and are "determined as a function of at least one of a track set; ***an attribute associated with the content***, a particular content segment, or a particular track; or a time associated with the content, ***a particular content segment***, or a particular track." Ex-1006, Cl. 12. The request URL received from a first client (e.g., computing device 1220 (blue)) corresponds to the recited "URL in a first request from a first client." In particular, Pyle explains that the manifest "describes at least one location" of the content and that the client issues "segment request URLs," so the server necessarily receives a URL in the client's initial HTTP request that identifies the media/segment being asked for, i.e., a URL "indicative of a requested media source." Pyle also discloses URL templates for those client requests (e.g., with Time/Index and optional Bandwidth/SegReqType substitutions), and an addressing scheme in which the request URL encodes the track set, attributes (bitrate/language/DRM), and time, further confirming that the URL itself carries information that identifies the requested media source.

190. While not necessary for this limitation, which is clearly taught by Pyle alone, I note that in the Pyle-Barraclough combination, Barraclough supplies the authenticated "first client" context, an Internet-connected, remote packet-based device that reaches the host's web page, provides credentials, and "in response the

Petitioner Disney Ex-1002, 0129

host authenticates the user … and facilitates media playlist access" after which the client (per Pyle) makes the first HTTP GET using the manifest/segment URL for the selected title; the host then serves the requested item from network storage. This is the receipt of a URL in a first request from a first client, as recited in Element 8[a].

191. As I also explained in the Technological Background section (*see* Section VIII.B), this was a known concept by 2010. *See, e.g.*, Ex-1030 (Guzman), 46-47; Ex-1023 (Gopalakrishnan), 7:26-28; Ex-1029 (Strasman) ¶¶77, 107; Ex-1021 (Barraclough'545), 12:5-8; Ex-1039 (Gabin), 135; Ex-1031 (Luby), 13:56-59.

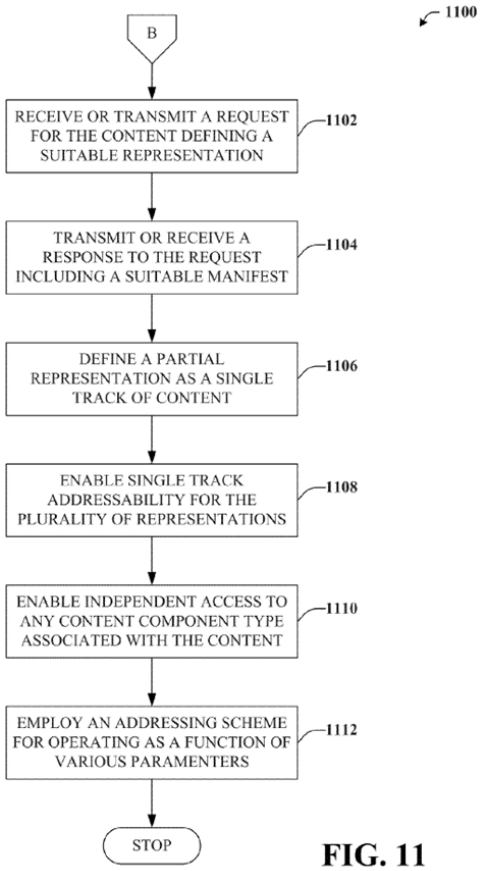


FIG. 11

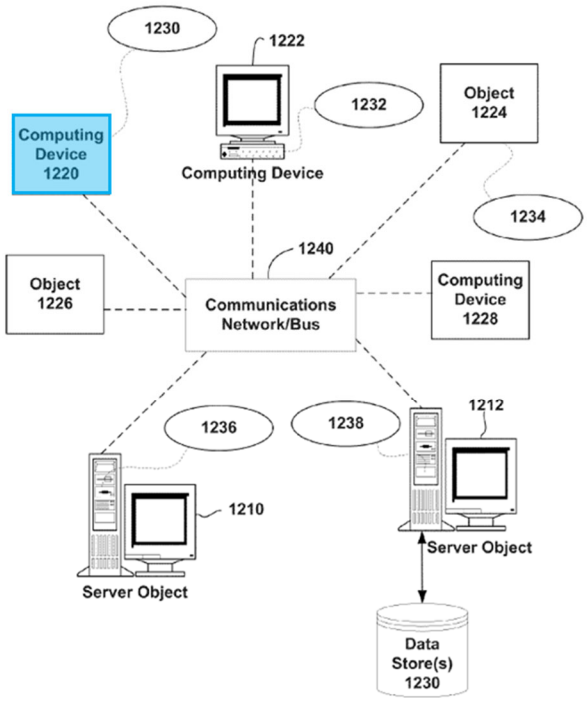


FIG. 12

Petitioner Disney Ex-1002, 0130

c. Element 8[b]: determining an identity of the first client;

192. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[b].

193. Ground 5: Pyle teaches determining the identity of the first client. As shown in Figure 12 below, Pyle teaches that "***computing objects or devices 1220*** … ***can be thought of as clients*** and computing objects 1210, 1212, etc. can be thought of as servers" that "provide data services, such as ***receiving data from client computing objects or devices 1220***, … storing of data, processing of data, ***transmitting data to client computing objects or devices 1220***." Ex-1006, 34:20-34, Fig. 12 (blue). Pyle further teaches that "individual manifests 204 can be optimized for … ***client preferences***, and so forth." Ex-1006, 8:23-27; *see also id.*, 11:63-12:3 ("even though many representations 208 of content 206 can exist, particular representation 508 can be a more suitable representation based upon … settings or ***preferences input by users of the presentation device***"). A POSITA would thus have understood that Pyle determines the identity of the first client during generation of the manifest because Pyle customizes the manifest to the particular user/client preferences. While Pyle leaves the details for ***how*** to determine the user/client's identity to a POSITA, this action would have been understood by a POSITA because associating unique identities, such as IP address, with a client was a standard computing feature at the time. Regardless, this feature is also expressly taught by Barraclough. Pyle's system 600 expressly states that the selection component

Petitioner Disney Ex-1002, 0131

determines a selected manifest based on "local configuration 602" (device/network state, settings, preferences) and/or a recommendation returned in response to a client request, and that request 606 "can include … all or a portion of [the] local configuration 602," which necessarily ties the request (and ensuing selection) to the particular client whose configuration and preferences are being applied. Pyle's claims likewise recite selection "based upon" device/network configuration or "settings or preferences," or a recommendation responsive to a request that includes the local configuration. Pyle further teaches that "individual manifests … can be optimized for … client preferences" and endpoint profiles, confirming per-client tailoring within the selection flow. Pyle's composition schema even carries user/device-tied attributes such as rating system/value and an "isAdultContent" flag, as well as text categories like "description for hearing impaired," reinforcing that the server must associate requests with a particular client profile to apply such preferences. In practice, a POSITA would have implemented this by identifying the specific client session (e.g., via standard HTTP/session tokens or network identifiers) within Pyle's HTTP server environment so that the correct per-client preferences drive manifest generation and delivery.

Petitioner Disney Ex-1002, 0132

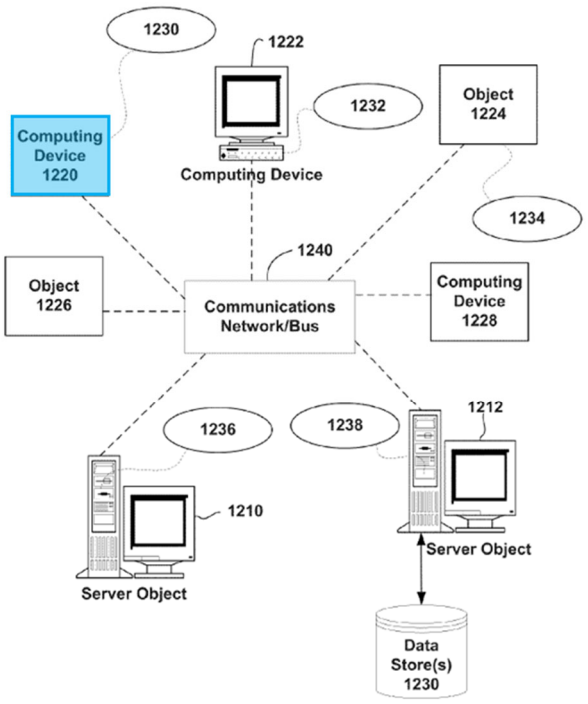


FIG. 12

194. Ground 6: Barraclough teaches determining the identity of the first client by authenticating the user on the server. Barraclough teaches that "[w]hen the registered user or a remote guest user wishes to access a playlist at a remote playback device, the user accesses a web page hosted by the host server, by way of a remote Internet access device such as a hand-held wireless telephone-type device." Ex-1007, 7:52-56. The user then "***provides authentication information***, such as the user name and password … and in response ***the host authenticates the user as an authorized user***, and facilitates media playlist access." Ex-1007, 7:56-62. Barraclough further teaches that "the host server 320 serves the populated content to ***authorized users*** at remote packet-based devices." Ex-1007, 16:20-22. In the Pyle–

Petitioner Disney Ex-1002, 0133

Barraclough combination, this authenticated identity is explicitly established at login (Barraclough), after which the host serves only "authorized users" and classifies the device making the request, by way of the request or by retrieving classification stored "for a particular user," to map delivery to an appropriate encoding; those per-user/per-device parameters then feed Pyle's manifest-selection logic.

195. In the Pyle-Barraclough combination, Pyle's method determines an identity of the first client by authenticating the user on the server, as taught by Barraclough. A POSITA would have been motivated to combine Pyle-Barraclough and had a reasonable expectation of success in doing so for the reasons described in Section XI.C.1. Operationally, once the host authenticates the user (Barraclough), the session is bound to that client's identity and associated device class/preferences; Pyle then selects a manifest "based upon data included in a request for the content" and/or the "local configuration" supplied with the request, precisely the per-client information tied to that authenticated identity. This straightforward integration yields the claimed determination of the first client's identity while leveraging Pyle's selection machinery to tailor delivery to that identified client.

196. As I also explained in the Technological Background section (*see* Section VIII.C), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 12:15-21; Ex-1021 (Mehrotra) ¶29.

Petitioner Disney Ex-1002, 0134

d. Element 8[c]: determining a first device type based on the identity of the first client;

197. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[c].

198. Ground 5: Pyle teaches determining the first device type based on the first client's identity. Pyle teaches that the client request includes device-specific details. Specifically, "***[r]equest 606 can include*** a request for specific content 206, as well as all or a portion of ***local configuration 602***, which can be employed for determining a suitable representation (e.g., particular representation 508) of content 206." Ex-1006, 12:19-23. Moreover, selection component 502 determines "selected manifest 504 ***based upon local configuration 602***," which relates "to at least one of ***device or network configuration, device or network state, device or network settings or preferences***, an indication of options, ***features***, or ***attributes*** in connection with content 206, or the like." Ex-1006, 12:10-17, Fig. 6. For example, if the content is a movie, "[t]he requesting device (not shown) can be, e.g., a ***HD television*** with very large form factor," which "can further be configured to only present media that is rated PG-13 (unless parent codes are entered) and is coupled to a reliable network with high bandwidth in an English-speaking household." Ex-1006, 10:27-33. A "composition component" examines data included in the client's request and determines "a suitable or optimal manifest" to "select and transmit … to the requesting device." Ex-1006, 10:53-56. The selection component is a "client-side selection" and the composition component is a "server-side selection." Ex-1006,

Petitioner Disney Ex-1002, 0135

12:32-36. Both may be used together to select the appropriate manifest based upon local configuration 602. Ex-1006, 12:32-40. A HD television corresponds to the recited "first device type." Pyle also teaches that the computing device "can be, e.g., a television, a desktop computer, a laptop, netbook, tablet, gaming console, smart phone or the like." Ex-1006, 12:63-13:2, Fig. 12 (blue). Pyle thus determines the first device type (i.e., HD television) based on the identity of the first client because, for example, a "client-side selection" component is used, which considers the device's "local configuration." Ex-1006, 12:32-36; *see also id.*, Ex-1006, 11:63-12:3 (manifest is selected based upon the "***capabilities, limitations, or a current state of the presentation device*** or an associated network, ***as well as settings or preferences input by users*** of the presentation device"). To the extent PO argues that Pyle's device type determination is not based on the client's identity, this feature is further taught by Barraclough. Additionally, Pyle explains that "individual manifests 204 can be optimized for … endpoint profiles or configurations," which a POSITA would have recognized as device-type categories (e.g., TV-class versus handheld-class) tied to the requesting client's session; accordingly, before any manifest is produced, the system must determine that the identified first client is a TV-class device. Ex-1006, 8:17-27, 12:63-13:2. And because request 606 may carry "all or a portion" of the client's local configuration, a POSITA would have understood that

Petitioner Disney Ex-1002, 0136

the device-type determination is made for that identified client using the configuration bound to that client's request. Ex-1006, 12:19-23

199. Ground 6: Barraclough teaches determining a first device type based on the identity of the first client for the reasons described in Section XI.A.2.d, namely, because Barraclough's device type classification is specifically based on the identity of a particular user that requested the media. In the Pyle-Barraclough combination, Pyle's selection and composition components determine the device type (i.e., HD television) based on the particular client (i.e., computing device 1220), as taught both by Pyle's receipt of user input/preferences and Barraclough's "particular user" data storage. More specifically, Barraclough authenticates the user and then serves content to "authorized users at remote packet-based devices," using per-user/device information to map requests to encodings appropriate for that device class; a POSITA would therefore have expected the host to resolve the device type from the authenticated client's identity and pass that to Pyle's selector, which relies on device configuration to drive manifest choice. Ex-1007, 7:56-62, 16:20-22; Ex-1006, 12:10-17, 12:32-40.

200. As I also explained in the Technological Background section (*see* Section VIII.D), this was a known concept by 2010. *See, e.g.*, Ex-1022 (Zapata) ¶¶192-193, 199; Ex-1021 (Barraclough'545), 6:11-22; Ex-1024 (Mehrotra) ¶2; Ex-1023 (Gopalakrishnan), 7:28-30, 14:14-19.

Petitioner Disney Ex-1002, 0137

e. Element 8[d]: based at least in part on the determination of the first device type, generating, with a processor, a first manifest having information for streaming one or more segments of the requested media source in accordance with a first chunking protocol via the data communications network;

201. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[d].

(1) Generating a first manifest with a processor based on determination of first device type

202. Pyle teaches selection component 502, a client-side component, which determines a "selected manifest 504 based upon local configuration 602" Ex-1006, 12:11-17. Moreover, Pyle teaches that composition component 210, a server-side component, "facilitate[s] composition of new manifest 422, which can be stored amongst other manifests 204." Ex-1006, 10:57-60. For example, "[t]he requesting device (not shown) can be, e.g., a HD television with very large form factor," which "can further be configured to only present media that is rated PG-13 (unless parent codes are entered) and is coupled to a reliable network with high bandwidth in an English-speaking household." Ex-1006, 10:27-33. A "composition component" examines data in the request and determines "a suitable or optimal manifest" to "select and transmit … to the requesting device." Ex-1006, 10:53-56; *see also id.*, 12:32-36 (selection component and composition component may be used together to select an appropriate manifest). Pyle uses a processor to generate the manifest file based on the first device type. Ex-1006, 37:5-8 ("a component may be … a processor

Petitioner Disney Ex-1002, 0138

running on a processor" or 'a processor"), Cls. 13 ("the following ***processor-executable components***: a selection component that ***receives a selected manifest*** … each manifest ***associated with a particular configuration*** of the particular representation of content"), 1. The manifest composed for the first device type (i.e., HD television) corresponds to the recited "first manifest." Additionally, Pyle states that the composed/selected manifest is "optimized" for endpoint profiles and device capabilities (e.g., television vs. handheld), directly tying generation of the first manifest to the earlier device-type determination. Ex-1006, 10:66-11:9, 8:17-27. A POSITA would have implemented this by mapping the determined device type to a manifest template or rule set (e.g., 1080p/5.1 tracks for TV-class devices) and invoking processor-executable code to emit the corresponding manifest.

203. While not necessary for this limitation, which is clearly taught by Pyle alone, I note that in the Pyle-Barraclough combination, the authenticated identity/device-class obtained at login (Barraclough) is supplied to Pyle's selection/composition components, which then generate the device-specific first manifest. Ex-1007, 7:56-62.

(2) Manifest has information for streaming one or more segments of the requested media source via the data communications network

204. Pyle's system facilitates HTTP "delivery of streaming media" by using a manifest component that maintains "multiple manifests 204_1-204_N for a single item

Petitioner Disney Ex-1002, 0139

of content 206 … suitable for delivery to or presentation at endpoints of a streaming network." Ex-1006, 7:45-53. Pyle teaches that "content 206 can be associated with various representations 208_1-208_M of that content 206, such as, e.g., different representations based upon different bitrates, resolutions, languages, or even an original theatrical version versus a PG-13 version, or substantially any other suitable attribute." Ex-1006, 7:53-58; *see also id.*, 8:17-27 ("individual manifests 204 can be optimized for particular delivery formats, wire formats, endpoint profiles or configurations, client preferences, and so forth"). Pyle also teaches a communications component that retrieves a particular representation of the content "according to a streaming protocol based upon information included in" the selected manifest. Ex-1006, 11:60-63. Pyle further explains that MPD information enables the client to formulate time-based segment request URLs "to obtain the latest segment of content," confirming that the manifest includes the addressing/timing information needed to stream "one or more segments" over the network. Ex-1006, 5:4-7, 5:21-22, 11:33-36, 32:43-49. Thus, once the first device type is known, the generated first manifest necessarily carries the URLs and segment descriptors for streaming the requested media source via HTTP.

(3) In accordance with a first chunking protocol

205. Pyle teaches that the first manifest has information for streaming the requested media source segments in accordance with a first chunking protocol. Pyle

Petitioner Disney Ex-1002, 0140

teaches that a "chunk" is a "contiguous set of samples for one track." Ex-1006, 6:17. Pyle further teaches that the manifest describes "the locations of various content segments, which can appear to a presentation device as a large set of small-interval content files, say 5 seconds or so." Ex-1006, 8:17-20. Additionally, Pyle teaches that "[e]vent information can also be delivered in … Smooth Streaming Transport Protocol (SSTP)." Ex-1006, 2:65-67; *see also id.*, 3:63-65 ("the disclosed subject matter can describe a next generation smooth streaming transport protocol with a client manifest"). SSTP was a well-known "chunking protocol" implemented by Microsoft by 2010. *See, e.g.*, Ex-1043 (Begen), 62-63. The SSTP chunking protocol, similar to the HLS chunking protocol (*see* Section XI.A.2.j), breaks video into small fragments (typically 2-4 seconds), and the client player downloads a manifest file (such as a .ism or .ismc file) that tells it which fragments to request. Beyond defining a "chunk" as "a contiguous set of samples for one track," Pyle teaches that segments appear as small-interval files (~5 seconds) and that content can be cross-indexed into chunks, establishing the chunked organization used for delivery. Pyle further discloses time-based segment addressing and carriage of segment start/duration in the wire format so that the client can formulate time-based segment request URLs, i.e., a concrete chunking protocol the session follows. And because switchable track sets rely on segment alignment/bitstream switching, the manifest necessarily defines

Petitioner Disney Ex-1002, 0141

the segment cadence/timeline per device profile, reinforcing that the first manifest operates "in accordance with a first chunking protocol."

206. Pyle's SSTP chunking protocol, when applied to the first manifest generated based on the determination of the first device type (i.e., HD television), corresponds to the recited "first chunking protocol." Pyle thus teaches the "chunking protocol" limitation under either parties' construction because it teaches both a "protocol that dynamically creates segments of media based on a [user's] request" and more broadly teaches a pre-determined protocol that allows data to be sent in a series of smaller, manageable pieces. Consequently, when the device type is an HD television, Pyle teaches generating, and, if needed, composing, a manifest that selects appropriate track sets and exposes time-based segment URLs with small-interval files (e.g., ~5-second segments), which constitutes the "first chunking protocol" for that session.

207. While not necessary for this limitation, which is clearly taught by Pyle alone, I note that in the Pyle-Barraclough combination, Barraclough's authenticated playlist/session provides the device-class mapping that informs those manifest rules (e.g., high-bitrate television profile), after which Pyle's processor-executable components deliver segments over HTTP in that chunked form. (Ex-1007, 7:56-62; Ex-1006, 11:60-63, 8:17-21, 10:66-11:9.) Moreover, in the Pyle–Barraclough combination, once the authenticated client is resolved as a TV-class device,

Petitioner Disney Ex-1002, 0142

Barraclough's server-side mapping of device classes to encoding parameter sets (and storage of versions for each device class) provides the concrete profile that Pyle's composition/selection uses to emit the first manifest exposing the appropriate representations and segment addressing for that class, i.e., the "first chunking protocol" for that session. In parallel, Barraclough's encoding-agnostic playlists keep title selection separate from encodings, so that Pyle's manifest generation can specialize delivery details (bitrates/segments/attributes) based on device type without changing playlist metadata.

208. As I also explained in the Technological Background section (*see* Section VIII.E), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 4:31-35; Ex-1032 (Johnson) ¶¶36, 38, 57; Ex-1030 (Guzman) at 41, 33; Ex-1023 (Gopalakrishnan), Abstract, 27:46-48, 23:15-27; Ex-1029 (Strasman) ¶¶9, 77, Cl. 19; Ex-1043 (Begen), 59-60, 62-63; Ex-1033 (MSDN Magazine) at 35; Ex-1034 (Sood), 3:40-43, 4:7-8; Ex-1039 (Gabin), 134-137; Ex-1040 (Carmel), 2:4-6, 2:22-28, 7:23-25; Ex-1041 (Brueck), 3:36-39, 7:62-8:5; Ex-1042 (Bocharov), Fig. 5, ¶14.

f. Element 8[e]: sending the first manifest to the first client;

209. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[e].

210. Pyle teaches that "a request for the content can be transmitted (e.g., by a client) or received (e.g., by a server)" and "***a response to the request can be***

Petitioner Disney Ex-1002, 0143

transmitted or received," which includes "***a suitable manifest*** associated with the suitable representation of the content." Ex-1006, 32:43-53.

211. As I also explained in the Technological Background section (*see* Section VIII.F), this was a known concept by 2010. *See, e.g.*, Ex-1023 (Gopalakrishnan), 2:15-19; Ex-1029 (Strasman) ¶14.

g. Element 8[f]: receiving the URL in a second request from a second client;

212. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[f] for the reasons described in Section XI.C.2.b. Specifically, Pyle teaches "***computing objects or devices*** 1220, 1222, 1224, 1226, ***1228***, etc." which "***can be thought of as clients***." Ex-1006. 34:21-22, Fig. 12 (green). Moreover, Pyle teaches that "a request for the content can be transmitted (e.g., by a client) or received (e.g., by a server)," which includes "a set of characteristics or attributes defining a suitable representation of the content." Ex-1006, 32:45-49. The client request includes a URL. Ex-1006, 5:21-22, 11:33-36, Cl. 12; *see also id.*, 10:40-41 ("both requesting devices can request the same movie (e.g., same content 206)"). The URL request received from a second client (e.g., computing device 1228) corresponds to the recited "URL in a second request from a second client." Further, Pyle expressly contemplates concurrent requests from multiple distinct clients: its server architecture is designed so that "multiple clients" utilize the server's information-gathering and delivery functions, and Figure 12 depicts several different client

Petitioner Disney Ex-1002, 0144

devices issuing HTTP requests to the server. Ex-1006, 34:41-43, 34:20-34, Fig. 12. In the two-device example (HD television vs. smartphone), each client uses MPD information to formulate time-based segment request URLs for the same content 206; thus, after receipt of the first client's URL, the server also receives the second client's segment/manifest URL(s) for that same requested media source, satisfying receipt of "the URL in a second request from a second client." Ex-1006, 5:4-7, 5:21-22, 10:27-41, 11:33-36, 32:43-49.

213. While not necessary for this limitation, which is clearly taught by Pyle alone, I note that in the Pyle-Barraclough combination, Barraclough confirms that different remote users/devices separately access the host, provide credentials, and, once authenticated, are served content at their respective "remote packet-based devices," which necessarily includes each device issuing its own HTTP request(s) containing the content-identifying URL. Ex-1007, 7:52-62, 16:20-22.

214. As I also explained in the Technological Background section (*see* Section VIII.C), this was a known concept by 2010. *See, e.g.*, Ex-1029 (Strasman) ¶107; Ex-1031 (Luby), 13:56-59.

Petitioner Disney Ex-1002, 0145

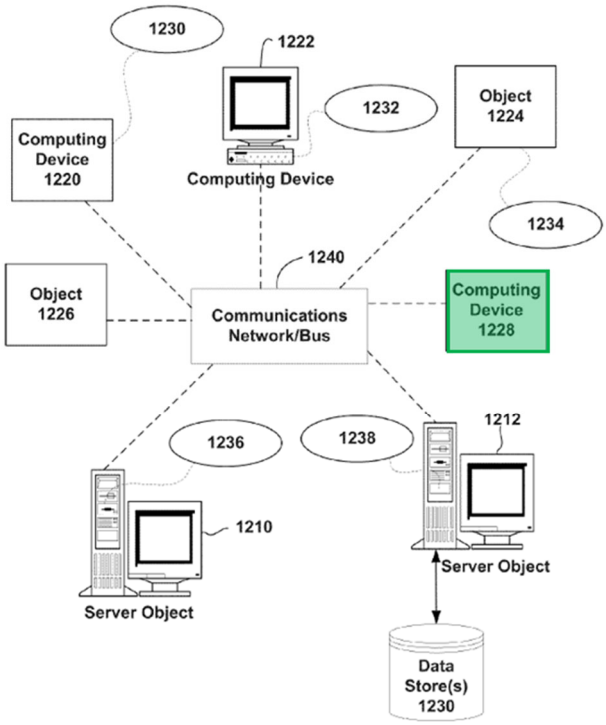


FIG. 12

h. Element 8[g]: determining an identity of the second client, the identity of the second client different from the identity of the first client;

215. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[g].

216. Ground 5: As explained in Section XI.C.2.c, Pyle teaches that "individual manifests 204 can be optimized for … ***client preferences***." Ex-1006, 8:23-27; *see also id.*, 11:63-12:3 ("even though many representations 208 of content 206 can exist, particular representation 508 can be a more suitable representation based upon … settings or ***preferences input by users of the presentation device***"). A POSITA would thus have understood that Pyle determines the identity of the second client (i.e., computing device 1228 (green)), which is different from the

Petitioner Disney Ex-1002, 0146

identity of the first client during generation of the manifest because Pyle customizes the manifest to the particular user/client's preferences. Ex-1006, Fig. 12. While Pyle leaves the details for ***how*** to determine the user/client's identity to a POSITA, this action would have been understood by a POSITA because associating unique identities, such as IP address, with a client was a standard computing feature at the time. Regardless, this feature is expressly taught by Barraclough. Additionally, Pyle explains that the incoming request can include "all or a portion of local configuration 602," and that the selector determines a "selected manifest 504 based upon local configuration 602," which encompasses device state/capabilities and user settings/preferences. Ex-1006, 12:10-23. Pyle's two-device example further illustrates that different clients (e.g., an HD television vs. a smartphone in transit) request the same content but with different configurations and preference sets, so the system must associate each request with the corresponding client in order to apply the correct per-client parameters, i.e., determine that the second client's identity is different from the first. Ex-1006, 10:27-41, 10:53-56. Pyle also states that "multiple clients" utilize the server's information-gathering/delivery functions, reinforcing that per-client identification occurs so that distinct optimizations can be applied. Ex-1006, 34:41-43.

Petitioner Disney Ex-1002, 0147

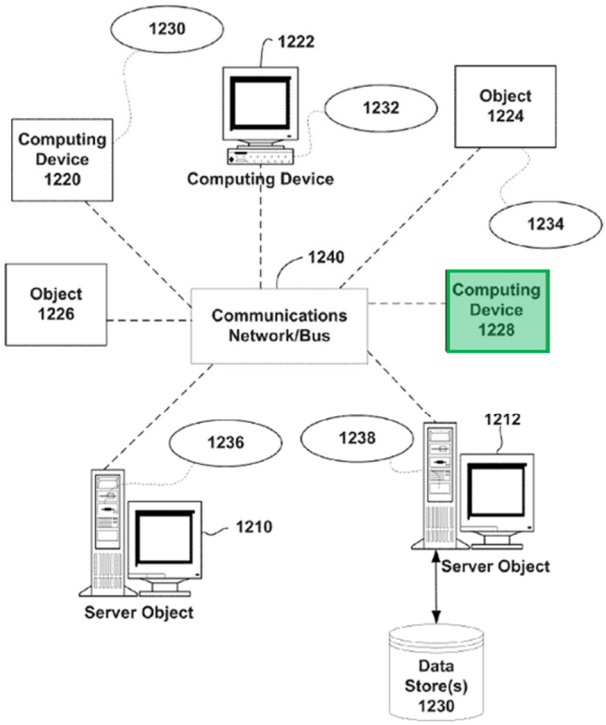


FIG. 12

217. Ground 6: Barraclough teaches this limitation for the reasons described in Section XI.C.2.c. Ex-1007, 7:56-62 (user "provides authentication information, such as the user name and password … and in response the host authenticates the user as an authorized user, and facilitates media playlist access").

218. In the Pyle-Barraclough combination, Pyle's method determines an identity of the second client by authenticating the user on the server, as taught by Barraclough.

219. As I also explained in the Technological Background section (*see* Section VIII.C), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 12:15-21; Ex-1021 (Mehrotra) ¶29.

i. Element 8[h]: determining a second device type based on the identity of the second client, the second device type different from the first device type;

220. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[h].

221. Ground 5: As explained in Section XI.C.2.d, Pyle teaches that the client request includes device-specific details. Specifically, the request from the client includes the local configuration information. Ex-1006, 12:19-23. Moreover, selection component 502 determines "selected manifest 504 ***based upon local configuration 602***," which relates "to at least one of ***device or network configuration, device or network state, device or network settings or preferences***, an indication of options, ***features***, or ***attributes*** in connection with content 206, or the like." Ex-1006, 12:10-17, Fig. 6. For example, if the content is a movie, "the requesting device can be ***a smart phone*** with a more modest form factor and resolution support, and is currently in transit at high speeds resulting in a less reliable network connection" where "the default language … is set to German," and "user preference indicates hearing impairment." Ex-1006, 10:33-39, 10:53-56. A smartphone corresponds to the recited "second device type." A "composition component" examines data included in the client's request and determines "a suitable or optimal manifest" to "select and transmit … to the requesting device." Ex-1006, 10:53-56. The selection component is a "client-side selection" and the composition component is a "server-side selection." Ex-1006, 12:32-36. Both may

Petitioner Disney Ex-1002, 0149

be used together to select the appropriate manifest based upon local configuration 602. Ex-1006, 12:32-40. Pyle thus determines the second device type (i.e., smart phone) based on the identity of the second client because, for example, a "client-side selection" component is used, which considers the device's "local configuration." Ex-1006, 12:32-36; *see also id.*, Ex-1006, 11:63-12:3. To the extent PO argues that Pyle's device type determination is not based on the client's identity, this feature is further taught by Barraclough. Accordingly, when the smartphone client submits request 606 including "all or a portion of" its local configuration 602 (e.g., reduced resolution support, less-reliable network state, different language/accessibility settings), the selection/composition logic attributes those characteristics to that identified client and determines that the device type for that client is a smartphone, which is different from the previously determined television device type. Ex-1006, 12:10–23, 12:32–40. Pyle also enumerates discrete endpoint categories, "television … smart phone or the like", confirming that device type is a class the system resolves per client before selecting or composing the manifest. Ex-1006, 12:63–13:2. And because Pyle teaches that "multiple clients" utilize the server concurrently, a POSITA would have understood that the system distinguishes at least a first and a second client and resolves their different device types prior to delivery. Ex-1006, 34:41–43.

Petitioner Disney Ex-1002, 0150

222. Ground 6: Barraclough teaches determining a second device type based on the identity of the second client for the reasons described in Section XI.A.2.i. In the Pyle-Barraclough combination, Pyle's selection and composition components determine the device type (i.e., smartphone) based on the particular client (i.e., computing device 1228), as taught both by Pyle's receipt of user input/preferences and Barraclough's "particular user" data storage.

223. As I also explained in the Technological Background section (*see* Section VIII.D), this was a known concept by 2010. *See, e.g.*, Ex-1022 (Zapata) ¶¶193, 199; Ex-1023 (Gopalakrishnan), 7:28-30, 14:14-19.

j. Element 8[i]: based at least in part on the determination of the second device type, generating, with the processor, a second manifest having information for streaming the one or more segments of the requested media source in accordance with a second chunking protocol via the data communications network, wherein content of the second manifest is different from content of the first manifest; and

224. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[i] for the reasons described in Section XI.C.2.e.

(1) Generating a second manifest with the processor based on determination of second device type, wherein content of the second manifest is different from content of the first manifest

225. Pyle teaches selection component 502, a client-side component, which determines a "selected manifest 504 based upon local configuration 602." Ex-1006,

Petitioner Disney Ex-1002, 0151

12:11-17. Moreover, Pyle teaches that composition component 210, a server-side component, "facilitate[s] composition of new manifest 422." Ex-1006, 10:57-60. For example, "the requesting device can be ***a smart phone*** with a more modest form factor and resolution support, and is currently in transit at high speeds resulting in a less reliable network connection" where "the default language … is set to German," and "user preference indicates hearing impairment." Ex-1006, 10:33-39. A "composition component" examines data in the request and determines "a suitable or optimal manifest" to "select and transmit … to the requesting device." Ex-1006, 10:53-56; *see also id.*, 12:32-36 (selection component and composition component may be used together to select an appropriate manifest), 37:5-8 ("a component may be … a processor running on a processor" or 'a processor"), Cl. 13. The manifest composed for the second device type (i.e., smartphone) corresponds to the recited "second manifest," which has different content from that of the first manifest file generated for a HD television. In other words, Pyle teaches that the client-side selector and server-side composer may be used together; in either case, processor-executable components generate the manifest from the second client's local configuration, producing a smartphone-specific second manifest (e.g., modest resolution, less-reliable bandwidth, German language, hearing-impairment) that differs from the TV-oriented first manifest.

Petitioner Disney Ex-1002, 0152

(2) Manifest has information for streaming one or more segments of the requested media source via the data communications network

226. Pyle's communications component retrieves a particular representation of the content "according to a streaming protocol based upon information included in" the selected manifest. Ex-1006, 11:60-63.

(3) In accordance with a second chunking protocol

227. Pyle teaches that its second manifest has information for streaming the requested media source segments in accordance with a second chunking protocol for the reasons described in Section XI.C.2.e. Specifically, Pyle teaches using Microsoft's well-known SSTP chunking protocol. Ex-1006, 2:65-67, 3:63-65. Pyle's SSTP chunking protocol, when applied to the second manifest generated based on the determination of the second device type (i.e., smartphone), corresponds to the recited "second chunking protocol." Moreover, as explained in Section XI.C.2.e, Pyle teaches the "chunking protocol" limitation under either parties' construction. In other words, because Pyle's manifests drive time-based segment addressing and present segments that "appear … as a large set of small-interval content files," a POSITA would have understood that the second manifest governs delivery "in accordance with a second chunking protocol" (e.g., the smartphone-appropriate segment addressing/track-set exposure reflected in that manifest). And, as Pyle notes, the disclosed system supports Microsoft Smooth Streaming Transport

Petitioner Disney Ex-1002, 0153

Protocol (SSTP) as well as other HTTP adaptive approaches, so the second manifest can operate under the chunking rules appropriate to the second device type.

228. To the extent PO argues that the second chunking protocol has to be different than the first chunking protocol, which is not a requirement of the claims, Pyle also teaches enabling "flexible selection between multiple wire formats for compatibility with more devices (e.g., some devices are limited to MPEG-2 Transport Streams, MP4 multiplex files, or protected interoperable file format (PIFF) Fragmented international organization for standardization (ISO), Base Media files, and so on)." Ex-1006, 2:52-57; *see also id.*, 5:8-10 (describing 3GPP live streaming method of MPD file), 5:23-26 (describing HTTP Adaptive Streaming). A "wire format" is the binary format or structure that the streaming protocol uses to send video chunks, metadata, and timing information between the client and server. Any of these streaming protocols, when applied to the second manifest generated based on the determination of the second device type (i.e., smartphone), would correspond to the recited "second chunking protocol." Thus, for a smartphone client, the processor can generate a second manifest that both differs in content from the TV-oriented first manifest and, where appropriate, operates under a second chunking protocol (or wire format) suitable for that device type.

229. While not necessary for this limitation, which is clearly taught by Pyle alone, I note that in the Pyle-Barraclough combination, Barraclough's authenticated,

Petitioner Disney Ex-1002, 0154

per-user/per-device context supplies the device-class determination for the second client, and Pyle's processor-executable selection/composition then emits the corresponding second manifest that governs chunked delivery for that session.

230. As I also explained in the Technological Background section (*see* Section VIII.E), this was a known concept by 2010. *See, e.g.*, Ex-1021 (Barraclough'545), 4:31-35; Ex-1032 (Johnson) ¶¶36, 38, 57; Ex-1030 (Guzman) at 41, 33; Ex-1023 (Gopalakrishnan), Abstract, 27:46-48, 23:15-27; Ex-1029 (Strasman) ¶¶9, 77, Cl. 19; Ex-1043 (Begen), 59-60, 62-63; Ex-1033 (MSDN Magazine) at 35; Ex-1034 (Sood), 3:40-43, 4:7-8; Ex-1039 (Gabin), 134-137; Ex-1040 (Carmel), 2:4-6, 2:22-28, 7:23-25; Ex-1041 (Brueck), 3:36-39, 7:62-8:5; Ex-1042 (Bocharov), Fig. 5, ¶14.

k. Element 8[j]: sending the second manifest to the second client.

231. In my opinion, Pyle or Pyle-Barraclough teaches Element 8[j] for the reasons described in Section XI.C.2.f. Ex-1006, 32:43-53 ("***a response to the request can be transmitted*** or received," which includes "***a suitable manifest*** associated with the suitable representation of the content").

232. As I also explained in the Technological Background section (*see* Section VIII.F), this was a known concept by 2010. *See, e.g.*, Ex-1023 (Gopalakrishnan), 2:15-19; Ex-1029 (Strasman) ¶14.

Petitioner Disney Ex-1002, 0155

233. Thus, in my opinion, Pyle or Pyle-Barraclough renders obvious Claim 8.

D. Grounds 7-8: Claims 13-15 are obvious over Pyle in view of Jain or Pyle-Barraclough in view of Jain.

1. A POSITA would have been motivated to combine Pyle's or Pyle-Barraclough's teachings with Jain's teachings and would have had a reasonable expectation of success.

234. A POSITA would have been motivated to combine the teachings of Pyle and Jain or Pyle-Barraclough and Jain and had a reasonable expectation of success in doing so because each relates to methods for streaming media content in response to a request. *See* Sections XI.B.1 and XI.C.1.

235. Pyle teaches that "manifests 204 can be described based upon a set of attributes associated with one or more representation 208 and/or content 206." Ex-1006, 8:33-35. An exemplary attribute includes "***free version with no encryption but with advertisements***." Ex-1006, 8:55-67. Pyle further teaches that a URL included in manifest 204 "can be determined as a function of at least one of ***a track set 404 (e.g., the next contiguous track set), an attribute associated with the content 206***, a particular content segment, or ***a particular track (e.g., attribute naming convention), or a time associated with content 206***, a particular content segment, or a particular track." Ex-1006, 11:35-47. Thus, a POSITA would have understood that Pyle's first and/or second manifest includes a URL which, when used, is indicative of playback of an advertisement and that a certain point has been reached in playback

Petitioner Disney Ex-1002, 0156

of the requested media source. Pyle leaves the implementation details as to how the manifest indicates that advertisements are to be shown during playback to a POSITA. While Pyle does not use the term "beaconing URL," a POSITA would have understood that Pyle's URL is used to signal or track playback of the advertisement or requested media content, which would meet the plain and ordinary meaning of this term (if it is not indefinite).

236. Jain teaches inserting advertisements into a media stream and beaconing URLs indicating playback of an advertisement and that a certain point has been reached in playback of the requested media source for the reasons described in Section XI.B.1.

237. A POSITA would have been motivated to indicate in Pyle's manifest file that an advertisement will be played or that a certain point has been reached in playback of the requested media source, using a beaconing URL, as taught by Jain, for the same reasons described in Section XI.B.1. Additionally, given that Pyle already teaches that the manifest file comprises URLs, it would have been obvious to implement a beacon for signaling or tracking the playback of an advertisement or requested media source in the form of a URL in Pyle's manifest, as taught by Jain. In the Pyle-Jain (Ground 7) or Pyle-Barraclough-Jain (Ground 8) combination, Pyle's first and/or second manifest would include a beaconing URL which, when used, indicates playback of an advertisement or that a certain point has been reached

Petitioner Disney Ex-1002, 0157

in playback of the requested media source, such as the start or end of the video, as taught by Jain.

238. In addition to the reasons described above, a POSITA would have been motivated to combine Pyle-Jain or Pyle-Barraclough-Jain because the combination solves a recognized problem (i.e., audience measurement) in a streaming system already tailored to heterogeneous devices. Specifically, Pyle and Barraclough build the delivery scaffolding (compositions, track sets, playlists, formats/bitrates to many devices) in the combination; and Jain supplies the measurement capability that the art identifies as lacking across platforms. Jain expressly states the need to "integrate exposure to digital media across platforms" and provides a solution that piggybacks on the audio channel used by those very streams. A POSITA would naturally have added Jain's beaconing to the Pyle-Barraclough streaming workflows to capture exposure without altering user-perceived playback.

239. Additionally, Pyle invites adding auxiliary tracks or time-aligned data; and Jain is exactly such an addition. Because Pyle lets publishers add tracks "without changing any of the other tracks," classify them as selectable/switchable, and even supply time-aligned "sparse" information, a POSITA would have recognized that beaconed audio (or a lightweight auxiliary track carrying beacon events) is a textbook use of Pyle's composition/track-set model. This would have been a

Petitioner Disney Ex-1002, 0158

straight-line, predictable integration of known elements to achieve a known benefit (i.e., exposure analytics).

240. Moreover, Barraclough already performs the step where Jain's encoding lives. Specifically, Barraclough's host/server formats or transcodes media "to a bitrate" or "to a format … different than" the source before streaming to remote devices. Jain teaches performing beacon encoding in the same server-side transcoding pipeline (pre-encoded or on-the-fly with a transcoding object/encoding engine). The references therefore align on where the work happens and with what: the server's encode/transcode path. A POSITA would have seen this as an obvious place to drop Jain's encoding library.

241. Furthermore, the standards and formats used by these references match the ordinary skill set. Specifically, Pyle targets ISO-base media (MP4/PIFF), adaptive streaming manifests and track-level switching; Jain targets mainstream streaming containers/codecs (H.264/MP3, Flash delivery) and provides both server- and client-side implementations; and while Barraclough is format-agnostic, it explicitly adapts to multiple formats and bitrates. All three rely on standard, interoperable media plumbing. A POSITA would thus have viewed the combination as routine engineering, not research.

242. Finally, the combination provides both a commercial and common-sense incentive. In practice, publishers want high-quality streaming that

Petitioner Disney Ex-1002, 0159

works everywhere and reliable analytics/attribution (who watched what, when, and where). Pyle and Pyle-Barraclough provide the former; Jain provides the latter. Therefore, combining them increases utility without changing the fundamental operation of either system, resulting in a predictable use of prior-art elements.

243. As I also explained in the Technological Background section (*see* Sections VIII.G, VIII.H, and VIII.I), these were known concepts by 2010. *See, e.g.*, Ex-1028 (Mao) ¶22; Ex-1035 (Smooth Streaming White Paper), 23; Ex-1034 (Sood), 4:4-8, Cl. 7, 10:28-39; Ex-1031 (Luby), 18:45-65, 37:64-38:4, 41:49-60; Ex-1038 (Krishnan) ¶¶35-37, 41, 46, 48; Ex-1037 (Bailey), 1:56-2:8, 4:65-5:17; Ex-1036 (Christian) ¶¶20, 25-27; Ex-1030 (Guzman), 33; Ex-1029 (Strasman) ¶122, Fig. 7.

244. A POSITA would have had a reasonable expectation of success in combining the teachings of Pyle or Pyle-Barraclough and Jain. All three describe methods for streaming media content in response to a request. *See* Sections XI.B.1 and XI.C.1. Additionally, both Pyle and Jain teach inserting advertisements into a media stream. Ex-1006, 8:55-67; Ex-1008 ¶¶32-33. A POSITA would thus have understood that these references disclose interrelated teachings based on well-understood technologies that would have been amenable to various well-understood and predictable combinations. For example, detailed integration paths are already given in the combination. Specifically, Jain walks through server-side (pre-encode

Petitioner Disney Ex-1002, 0160

or on-the-fly) and client-side beacon insertion, including synchronization, ActionScript triggers, and the presence of an encoding library at the content site's encoding/transcoding engine; Barraclough's host/server side is already built to handle that step; and Pyle's compositions are already built to accept new tracks or event data without disturbing others. A POSITA implementing beams in the server pipeline and referencing them in Pyle's composition would have expected it to work. Moreover, the file formats and streaming behaviors of the references are compatible and routine. Specifically, Jain's examples (H.264/MP3; Flash delivery) map onto the MP4/HTTP adaptive ecosystems Pyle and Barraclough contemplate, and Barraclough explicitly converts among formats/bitrates for target devices, exactly the scenario where Jain teaches inserting the beacon during transcoding. No specialized hardware changes would have been required, and Jain even contemplates client-side insertion when preferred. Finally, architectural decoupling in Pyle reduces risk. Specifically, because Pyle stores partial representations as individually addressable single tracks and supports independent access to each content component, adding a beaconed-audio track (or replacing an audio track with a beaconed version) would not have disturbed video/text tracks or DRM. Pyle's architecture thus would have given a POSITA high confidence in successful integration of the combination.

Petitioner Disney Ex-1002, 0161

2. Dependent Claims 13-15

a. Claim 13

(1) Element 13[pre]: The method of streaming media via the data communications network as recited in claim 8, further comprising:

245. In my opinion, Pyle in view of Jain ("Pyle-Jain") or Pyle-Barraclough in view of Jain ("Pyle-Barraclough-Jain") teaches the preamble. *See* Section XI.C.2.a.

(2) Element 13[a]: receiving data regarding the requested media source, wherein the data includes information indicative of at least one point, during playback of the requested media source, at which at least one other media source is to be shown;

246. In my opinion, Pyle-Jain or Pyle-Barraclough-Jain teaches Element 13[a].

247. Pyle teaches receiving data regarding the requested media source at the client device. Pyle teaches that "***a request for the content*** can be transmitted (e.g., by a client) or received (e.g., by a server)" and "a response to the request can be transmitted or received," which includes "***a suitable manifest associated with the suitable representation of the content***." Ex-1006, 32:43-53. The manifest, which is received by the client device, includes the recited data received "regarding the requested media source." In the Pyle-Jain or Pyle-Jain-Barraclough combination, a POSITA would have populated that received manifest (or a companion

Petitioner Disney Ex-1002, 0162

configuration fetched with it) with Jain's beacon endpoints and timing parameters (e.g., ad-start/ad-end markers), such that the very data the client receives identifies the points in playback when the other media will be inserted.

248. To the extent PO argues that the "receiving" step refers to data received ***at the server*** regarding the requested media source, this is also taught by Pyle. Pyle teaches that "request 606 can be transmitted to component composition 210" (i.e., a server-side component), which "select[s] a particular manifest 418 from the set of available manifests 204," and transmits it to the client. Ex-1006, 10:22-25, 12:24-36; *see also id.*, 9:29-32 (manifest component 202 maintains "manifests 204 in data store 212," which is a "repository of all or portions of data, data sets, or information," and composition component 210 organizes "these manifests 204 based upon logic that resides in data store 212"). The manifests may also be received by the server components from content providers. Ex-1006, 10:63-65 ("a content provider can compose new manifest 422 specifically tailored to, e.g., track sets that receive the most requests, deliveries, and/or presentations"). Thus, Pyle's server components receive data (i.e., the manifest) regarding the requested media source (i.e., requested content) from content providers via data store 212. In the Pyle-Jain or Pyle-Barraclough-Jain combination, a POSITA would have integrated Jain's ad-event definitions (including the beacon URLs that confirm those events) into the very

Petitioner Disney Ex-1002, 0163

manifest that the server receives and selects before responding, so the server itself "receives" data that is indicative of the ad-insertion point(s).

249. Pyle further teaches that data in the manifest includes information indicative of at least one point during playback of the requested media source at which at least one other media source is to be shown. Pyle teaches that "manifests 204 can be described based upon a set of attributes associated with one or more representation 208 and/or content 206." Ex-1006, 8:33-35. An exemplary attribute includes "version 322 (e.g., a live version, a remixed version, theatrical version, ***free version with no encryption but with advertisements***, etc.), as well as other suitable attributes 324." Ex-1006, 8:55-67. Data in the manifest received at the client device and/or by the server components from a content provider or the data store thus includes information indicative of at least one point during playback of the requested media source (i.e., content) at which the other media source (i.e., advertisements) is to be shown. Pyle leaves the implementation details as to how the manifest indicates when advertisements are to be shown during playback to a POSITA. These details are taught by Jain. In the Pyle-Jain or Pyle-Jain combination, a POSITA would have used Pyle's attribute/time selectors to define the ad window(s) on the shared timeline and associate those windows with Jain's beacon endpoints, so that the presence and later invocation of those endpoints reflect the very points in playback when the ad is to be displayed.

Petitioner Disney Ex-1002, 0164

250. As shown in Figure 2 below, Jain teaches "additional beacon events" 203, where "additional data may be transmitted in the beacon to include URLs and video ID's when a video is loaded and played." Ex-1008 ¶33, Fig. 2 (yellow). Jain teaches that "[a]s the advertisement is served, an event beacon, which may include advertisement URL data, is transmitted." Ex-1008 ¶33; *see also id.*, ¶32 ("***[a]t the 5 minute mark (301 sec.), a 15 second advertisement is served***," and at the end "of the advertisement (316 sec.), the video continues to play until its conclusion (600 sec.)"). Jain thus teaches indicating at least one point during playback of the requested media source (i.e., video) at which the other media source (i.e., advertisements) is to be shown using a URL beacon. In Pyle-Jain or Pyle-Barraclough-Jain combination, those same beacon definitions are established (or referenced) by the manifest received at playback start, so the data received "regarding the requested media source" is what specifies the ad-insertion point(s).

251. Moreover, in the Pyle-Jain (Ground 7) or Pyle-Barraclough-Jain (Ground 8) combination, Pyle's manifest would include information indicative of at least one point during playback of the requested media content at which an advertisement is to be shown using a URL beacon, as taught by Jain.

Petitioner Disney Ex-1002, 0165

200	Time →	0-3 sec	4 sec ---------- 180 ---------- 300 sec				301-315 sec	316—599 sec	600 sec
201	Activities→	Video Loaded	Video Played				Ad served	Video continues	Video End
202	Audio Beacon detected by portable device →	Continuous Audio representing the website (e.g., Hulu.com)							
203	Additional beacon events →	URLs (Hulu.com & Video ID)					AD Event Beacon (co-relation URL)		Video End Beacon
204	Audio matching events detected by portable device →		Audio Signature Sample 1	Audio Signature Sample 2	Audio Signature Sample 3 (Content Match Achieved)	Same Signature Sample 4 (No Change)	Audio Signature Sample 5 (Different Content couldn't match)	Audio Signature Sample 5 (Same Video again)	No Signature (But Can't assume video is completed
205	Overlay portable device data	X			X		X	X	X
206	End- User Experience	URL of site where the video was played			Content Matched		Ad Served		Video Played till the End

FIG. 2

(3) Element 13[b]: receiving information regarding the at least one other media source; and

252. In my opinion, Pyle-Jain or Pyle-Barraclough-Jain teaches Element 13[b] for the reasons described in the previous section. Specifically, Pyle teaches receiving information regarding the other media source (i.e., advertisement) because the manifest received at the client device and/or the server components includes attributes such as "free version … ***with advertisements***." Ex-1006, 8:55-67. Moreover, Jain teaches providing timing information for switching to advertisements in a URL beacon. For example, Jain teaches that "***[a]t the 5 minute mark (301 sec.), a 15 second advertisement is served***," and at the end "of the advertisement (316 sec.), the video continues to play until its conclusion (600 sec.)." Ex-1008 ¶32. In the Pyle-Jain or Pyle-Barraclough-Jain combination, Jain's timing

Petitioner Disney Ex-1002, 0166

information for switching to advertisements would be included in Pyle's manifest, which corresponds to the recited "information" received "regarding the at least one other media source."

(4) Element 13[c]: including, in either or both of the first manifest and the second manifest, information for streaming one or more segments of the at least one other media source at the at least one point during the playback of the requested media source.

253. In my opinion, Pyle-Jain or Pyle-Barraclough-Jain teaches Element 13[c] for the reasons described in Section XI.D.2.a.2. Specifically, Pyle teaches including in the manifest, information for streaming the one or more segments of the at least one other media source (i.e., advertisements) (Ex-1006, 8:55-67) and the Pyle-Jain or Pyle-Barraclough-Jain combination teaches including timing information for streaming the advertisements at the at least one point during the playback of the requested media source (i.e., video). Jain teaches that "***[a]t the 5 minute mark (301 sec.), a 15 second advertisement is served***," and at the end "of the advertisement (316 sec.), the video continues to play until its conclusion (600 sec.)." Ex-1008 ¶32. In the Pyle-Jain (Ground 7) or Pyle-Barraclough-Jain (Ground 8) combination, Pyle's first and/or second manifest includes information for streaming an advertisement at the at least one point during playback of the requested media content, as taught by Jain.

Petitioner Disney Ex-1002, 0167

254. As I also explained in the Technological Background section (*see* Section VIII.G), this was a known concept by 2010. *See, e.g.*, Ex-1034 (Sood), 10:28-39; Ex-1028 (Mao) ¶22; Ex-1035 (Smooth Streaming White Paper), 23.

b. Claim 14: The method of streaming media via the data communications network as recited in claim 13, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of playback of at least one advertisement.

255. In my opinion, Pyle-Jain or Pyle-Barraclough-Jain teaches Claim 14 for the reasons described in Section XI.D.2.a.4.

256. Pyle teaches that an exemplary attribute of a manifest includes "***free version … with advertisements***." Ex-1006, 8:55-67. Pyle further teaches that a URL included in manifest 204 "can be determined as a function of at least one of a track set 404 (e.g., the next contiguous track set), ***an attribute associated with the content 206, a particular content segment, or a particular track (e.g., attribute naming convention),*** or a time associated with content 206, a particular content segment, or a particular track." Ex-1006, 11:35-47. Thus, a POSITA would have understood that Pyle's first and/or second manifest includes a URL which, when used, is indicative of playback of at least one advertisement. While Pyle does not use the term "beaconing URL," a POSITA would have understood that Pyle's URL is used to signal or track playback of the advertisement, which would meet the plain and ordinary meaning of this term (if it is not indefinite). In the Pyle-Jain or Pyle-

Petitioner Disney Ex-1002, 0168

Barraclough-Jain combination, a POSITA would for example implement these manifest URLs as beacon endpoints (e.g., HTTP GET/POST) parameterized with the media/ad identifiers that Jain teaches, and trigger them at the ad start/end boundaries that Jain's ad-event timeline defines, so use of the URL is itself evidence that an ad is playing. In the Pyle-Barraclough-Jain combination, Barraclough's host would for example generate or modify the returned manifest(s) to include the appropriate beaconing URL(s) for the target device/bitrate while Jain supplies the precise ad-playback events that cause those URLs to be invoked.

257. Jain also teaches Claim 14 for the reasons described in Section XI.B.2.a. In the Pyle-Jain (Ground 7) or Pyle-Barraclough-Jain (Ground 8) combination, Pyle's first and/or second manifest would include a beaconing URL which, when used, indicates playback of an advertisement, as taught by Jain.

258. As I also explained in the Technological Background section (*see* Section VIII.H), this was a known concept by 2010. *See, e.g.*, Ex-1034 (Sood), 10:30-39; Ex-1031 (Luby), 37:64-38:4, 41:49-60; Ex-1038 (Krishnan) ¶¶35-37, 41, 46, 48; Ex-1037 (Bailey), 1:56-2:8; Ex-1036 (Christian) ¶¶20, 25-27.

c. Claim 15: The method of streaming media via the data communications network as recited in claim 8, wherein either or both of the first manifest and the second manifest include at least one beaconing URL which, when used, is indicative of a certain point has been reached in playback of the requested media source.

259. In my opinion, Pyle-Jain or Pyle-Barraclough-Jain teaches Claim 15.

Petitioner Disney Ex-1002, 0169

260. Pyle teaches that "[t]ime based Requests can use time information in the video stream that is shared by all alternate tracks in order to push information utilized for the next universal resource locator (URL) requests for an entire Switch Group or Groups sharing a common timeline." Ex-1006, 2:61-67. Pyle further teaches that a URL included in manifest 204 "can be determined as a function of at least one of ***a track set 404 (e.g., the next contiguous track set)***, an attribute associated with the content 206, a particular content segment, or a particular track (e.g., attribute naming convention), or ***a time associated with content 206***, a particular content segment, or a particular track." Ex-1006, 11:35-47. Thus, a POSITA would have understood that Pyle's first and/or second manifest includes a URL which, when used, is indicative of a certain point has been reached in playback of the requested media source. While Pyle does not use the term "beaconing URL," a POSITA would have understood that Pyle's URL is used to signal or track playback of the requested media content, which would meet the plain and ordinary meaning of this term (if it is not indefinite). In the Pyle-Jain or Pyle-Barraclough-Jain combination, a POSITA for example would have implemented same time-dependent URL mechanisms as beacon endpoints that the player calls at manifest-defined positions (e.g., start, mid-roll boundary, end); Jain's disclosure of "video loaded/played" data, an "event beacon" when an ad is served, and a "video end beacon" provides the canonical "certain points" in playback that the beaconing URL

Petitioner Disney Ex-1002, 0170

denotes when used. In addition, Pyle's schema allows time-based segment request URLs (including templates where $Time$ is replaced by the segment start time), reinforcing that URLs in the manifest are expressly tied to timeline positions, a natural place to include beaconing URLs indicative that a specific playback time has been reached.

261. Jain also teaches Claim 15 for the reasons described in Section XI.B.2.b. In the Pyle-Jain (Ground 7) or Pyle-Barraclough-Jain (Ground 8) combination, Pyle's first and/or second manifest would include a beaconing URL which, when used, indicates that a certain point has been reached in playback of the requested media source, such as the start or end of the video, as taught by Jain.

262. As I also explained in the Technological Background section (*see* Section VIII.I), this was a known concept by 2010. *See, e.g.*, Ex-1030 (Guzman), 33; Ex-1029 (Strasman) ¶122, Fig. 7; Ex-1034 (Sood), 4:4-8, Cl. 7; Ex-1031 (Luby), 18:45-65; Ex-1037 (Bailey), 4:65-5:17.

XII. CONCLUSION

263. For the reasons above, it is my opinion that Claims 8 and 13-15 of the '639 Patent are obvious based on each of the grounds specified in my Declaration.

264. I declare that all statements made in my Declaration of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false

Petitioner Disney Ex-1002, 0171

statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: November 5, 2025

Dr. Brian W. Anthony, Ph.D.

Petitioner Disney Ex-1002, 0172