

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

THE WALT DISNEY COMPANY, DISNEY STREAMING
SERVICES LLC, HULU LLC, and NETFLIX INC.,
Petitioner,

v.

WAG ACQUISITION, L.L.C.,
Patent Owner.

IPR2022-01228
Patent 9,742,824 B2

Before HUBERT C. LORIN, JOHN A. HUDALLA, and
STEVEN M. AMUNDSON, *Administrative Patent Judges*.

AMUNDSON, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)

I. INTRODUCTION

The Walt Disney Company, Disney Streaming Services LLC, Hulu LLC, and Netflix Inc. (collectively “Petitioner”) filed a Petition requesting an *inter partes* review of claims 1–12 in U.S. Patent No. 9,742,824 B2 (Exhibit 1001, “the ’824 patent”) under 35 U.S.C. §§ 311–319. Paper 5 (“Pet.”). WAG Acquisition, L.L.C. (“Patent Owner”) filed a Preliminary Response. Paper 11 (“Prelim. Resp.”).

In the Institution Decision, we instituted review based on all challenged claims and all challenges included in the Petition. Paper 12 (“Inst. Dec.”). We have jurisdiction under 35 U.S.C. § 6. We issue this Final Written Decision under 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73. For the reasons explained below, Petitioner has shown by a preponderance of the evidence that claims 1–12 in the ’824 patent are unpatentable. *See* 35 U.S.C. § 316(e) (2018).

II. BACKGROUND

A. Procedural History

After we instituted review, Patent Owner filed a Response (Paper 16, “Resp.”), Petitioner filed a Reply (Paper 18, “Reply”), and Patent Owner filed a Sur-reply (Paper 20, “Sur-reply”). On November 13, 2023, we held an oral hearing, and the record includes the hearing transcript. Paper 27 (“Tr.”).

B. Real Parties in Interest

Petitioner identifies the following real parties in interest: The Walt Disney Company, Disney Streaming Services LLC, Disney Streaming Technology LLC, Disney Platform Distribution, Inc., BAMTECH, LLC, ESPN, Inc., Hulu LLC, and Netflix Inc. Pet. 1; Paper 9, 1; Paper 10, 1.

Patent Owner identifies itself as the sole real party in interest. Paper 6, 2.
The parties do not raise any issue about real parties in interest.

C. Related Matters

Petitioner and Patent Owner identify the following civil actions where Patent Owner has asserted the '824 patent and related patents against Petitioner and other alleged infringers:

- *WAG Acquisition, L.L.C. v. Amazon.com, Inc. et al.*, No. 6:21-cv-00815 (W.D. Tex. filed Aug. 6, 2021);
- *WAG Acquisition, L.L.C. v. Google LLC et al.*, No. 6:21-cv-00816 (W.D. Tex. filed Aug. 6, 2021);
- *WAG Acquisition, L.L.C. v. Netflix, Inc.*, No. 6:21-cv-01083 (W.D. Tex. filed Oct. 18, 2021);
- *WAG Acquisition, L.L.C. v. The Walt Disney Company et al.*, No. 2:21-cv-08230 (C.D. Cal. filed Oct. 18, 2021);
and
- *WAG Acquisition, L.L.C. v. Hulu LLC*, No. 2:21-cv-08242 (C.D. Cal. filed Oct. 18, 2021) (now consolidated with the immediately preceding civil action).

Pet. 2; Paper 6, 2–3; Paper 9, 1–2; Paper 10, 1.

Petitioner and Patent Owner identify the following Board proceedings as related matters involving the '824 patent or a related patent asserted against Petitioner in a civil action:

- *The Walt Disney Company et al. v. WAG Acquisition, L.L.C.*, IPR2022-01227 (U.S. Patent No. 9,762,636 B2);
- *The Walt Disney Company et al. v. WAG Acquisition, L.L.C.*, IPR2022-01346 (U.S. Patent No. 9,729,594 B2);
- *Google LLC v. WAG Acquisition, L.L.C.*, IPR2022-01411 (U.S. Patent No. 9,729,594 B2);

- *Google LLC v. WAG Acquisition, L.L.C.*, IPR2022-01412 (U.S. Patent No. 9,742,824 B2);
- *Google LLC v. WAG Acquisition, L.L.C.*, IPR2022-01413 (U.S. Patent No. 9,762,636 B2);
- *Amazon.com, Inc. et al. v. WAG Acquisition, L.L.C.*, IPR2022-01429 (U.S. Patent No. 9,729,594 B2);
- *Amazon.com, Inc. et al. v. WAG Acquisition, L.L.C.*, IPR2022-01430 (U.S. Patent No. 9,742,824 B2); and
- *Amazon.com, Inc. et al. v. WAG Acquisition, L.L.C.*, IPR2022-01433 (U.S. Patent No. 9,762,636 B2).

Paper 9, 4; Paper 10, 3–4; Prelim. Resp. 3.

Additionally, Petitioner and Patent Owner identify numerous civil actions and Office proceedings involving patents related to the '824 patent, e.g., U.S. Patent No. 8,122,141 B2 (Exhibit 1005). Pet. 1–4; Paper 6, 3–7; Paper 9, 1–4; Paper 10, 1–3.

D. The '824 Patent (Exhibit 1001)

The '824 patent, titled “Streaming Media Delivery System,” issued on August 22, 2017, from an application filed on October 3, 2016. Ex. 1001, codes (22), (45), (54). The patent identifies that application as the latest in a series of continuation and continuation-in-part applications that started with an application filed on March 28, 2001. *Id.* at 1:6–22, code (63). The patent claims priority to a provisional application filed on September 12, 2000. *Id.* at 1:22–28, code (60). The patent states that the invention relates to “systems and methods for delivering streaming media, such as audio and video, on the Internet.” *Id.* at 1:52–55; *see id.* at code (57).

The '824 patent describes problems with conventional streaming technologies. *See* Ex. 1001, 2:34–3:41. As an example, “users viewing or

listening to streaming content over Internet connections often encounter interruptions,” called “dropouts,” due to “unanticipated transmission delays and losses that are inherent in many Internet protocols.” *Id.* at 2:34–40; *see id.* at 5:25–32. Conventional streaming technologies employ “a pre-buffering technique to store up enough audio or video data in the user’s computer so that it can play the audio or video with a minimum of dropouts.” *Id.* at 2:42–45. But this “process requires the user to wait until enough of the media file is buffered in memory before listening or viewing can begin,” e.g., to wait “from ten to twenty seconds or more.” *Id.* at 2:45–47, 2:53–54.

As another example, the “audio or video data is delivered from the source at the rate it is to be played out.” Ex. 1001, 2:63–65; *see id.* at 5:60–65, 6:8–12, 8:66–9:2. Because “transmission of audio/video media data to the user takes place at the rate it is played out, the user’s buffer level can never be increased or replenished while it is playing” if Internet slowdowns or gaps cause the user’s buffer level to decrease from its initial level. *Id.* at 3:5–11; *see id.* at 10:37–38. “In time, extended or repeated occurrences of these gaps empty the user’s buffer.” *Id.* at 3:11–13; *see id.* at 3:34–35. When that occurs, the “audio/video material stops playing, and the buffer must be refilled to” its initial level before playing resumes. *Id.* at 3:13–15; *see id.* at 3:35–40.

The ’824 patent identifies a need for “improved systems and methods for delivering streaming content over the Internet” that:

- (1) “facilitate continuous transmission of streaming content”;
- (2) “respond on demand without objectionable buffering delay”; and

(3) “perform without disruption or dropouts.”

Ex. 1001, 3:45–50.

Figure 1 in the '824 patent (reproduced below) depicts a system for delivering streaming content over the Internet:

Fig. 1

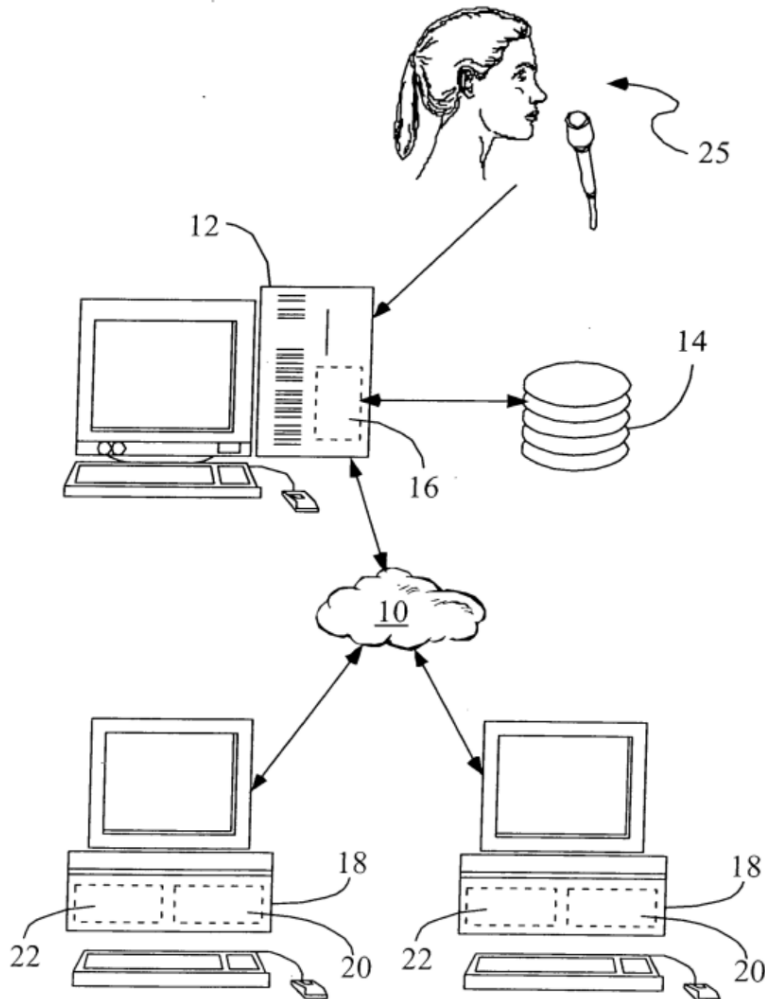


Figure 1 illustrates a streaming system including server 12 with server buffer 14 and buffer manager 16 and at least one user computer 18 with user buffer 20 and buffer manager 22. Ex. 1001, 4:23–25, 6:32–37, 6:48–59, Fig. 1. Server 12 and user computer 18 communicate “via the Internet 10 or other data communications medium.” *Id.* at 6:48–51.

Server buffer 14 stores time-sequenced data elements. Ex. 1001, 6:30–36. Server buffer 14 “is filled the first time the media source connection is established or a disk file is read.” *Id.* at 8:3–4. “Once server buffer 14 is full, for each new data element received into the buffer the oldest data element is deleted (or displaced) from the buffer.” *Id.* at 8:9–11.

After user computer 18 connects to server 12, the server “sends the media data to the user computer” at “a rate faster than the playback rate, which may be the highest rate that the data connection between the server and the user computer will support, or any lower rate that is a higher rate than the playback rate.” Ex. 1001, 8:15–22; *see id.* at 8:61–65, 9:38–41, 14:66–15:1. Server 12 provides data elements at that rate until “the predetermined amount of data that had been stored in the server buffer has been transferred to” the user computer buffer. *Id.* at 8:22–24. After transferring the contents of the server buffer to the user computer buffer and reaching a steady-state condition, each data element “is immediately sent out to the user computer” when it arrives at the server. *Id.* at 8:25–28.

If, however, the user computer buffer “begins to deplete or becomes depleted due to networking interruptions, the server will attempt to send as much data as is necessary to rebuild” the user computer buffer “to the proper level, again at higher than a playback rate.” Ex. 1001, 10:24–29. This permits rebuilding of the user computer buffer “under circumstances wherein Internet interruptions have blocked the normal flow of data.” *Id.* at 10:29–31.

A “data communications transport mechanism,” such as the Transmission Control Protocol (TCP), “may be used for the reliable delivery of data in an ordered sequence from the source of the media data to the

server, or from the server to the media player software of the user computer.” Ex. 1001, 8:38–42. “Resending missing data is the responsibility of the reliable transport mechanism.” *Id.* at 8:42–43. “The server buffer 14 ‘sends’ data by delivering it to the transport mechanism.” *Id.* at 8:43–45. “The transport mechanism actually manages transmission of the data across the communications medium, and has processes to determine if all the data that has been sent has been received by the destination.” *Id.* at 8:45–48.

“All media data to be delivered to a user computer may be sent at a higher than playback rate, either by the server buffer 14 passing media data to the transport mechanism, or by the transport mechanism delivering or redelivering the media data to the user computer.” Ex. 1001, 8:61–65.

The ’824 patent describes an embodiment where the “server buffer manager, or the media source, provides for sequentially numbering the media data elements.” Ex. 1001, 14:48–51. The “server buffer manager does not maintain a pointer into the server buffer for each user.” *Id.* at 14:51–53. Rather, “the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received.” *Id.* at 14:53–55.

To initiate streaming, the “user computer transmits a request to the server to send one or more data elements, specifying the serial numbers of the data elements.” Ex. 1001, 14:57–59. The “server responds by sending the requested data elements, and depends upon the reliable transmission protocol to assure delivery.” *Id.* at 14:59–62. The “user computer then continues with additional data requests for the duration of playing the audio/video material.” *Id.* at 14:62–64. “In this manner, the user computer, not

the server, maintains the record of the highest data element number stored in the user computer buffer.” *Id.* at 14:64–66. The “media data will be transmitted to the user computer as fast as the data connection between the user computer and the server will allow.” *Id.* at 14:66–15:1.

In the embodiment where the user computer transmits requests to the server for data elements and specifies the serial numbers of the requested data elements, the server provides “a buffer manager and a [first-in, first-out] FIFO buffer.” Ex. 1001, 15:1–3. The server also provides means for (1) “receiving the sequentially numbered media data elements from a broadcast media source or a file based media source” and (2) “storing those data elements in the FIFO buffer.” *Id.* at 15:1–6.

In the embodiment where the user computer transmits requests to the server for data elements and specifies the serial numbers of the requested data elements, the server buffer manager comprises means for the following:

- (1) “receiving the media data”;
- (2) “supplying media data in order to the FIFO buffer”;
- (3) “supplying the FIFO buffer with a predetermined number of data elements”; and
- (4) “deleting the oldest data element in the buffer as each new data element is received” once “the FIFO buffer is full.”

Ex. 1001, 15:6–11. The server buffer manager attempts to maintain the predetermined number of data elements in the FIFO buffer. *Id.* at 15:12–13.

In the embodiment where the user computer transmits requests to the server for data elements and specifies the serial numbers of the requested data elements, the user computer is “associated with a media player software

incorporating a user buffer.” Ex. 1001, 15:16–17. Further, the user computer comprises means for the following:

- (1) “receiving and storing a predetermined number of media data elements which are received sequentially by the media player”;
- (2) “playing the data out sequentially as audio and/or video”;
and
- (3) “deleting media data elements from the buffer as they are played out.”

Id. at 15:16–22. “As data is played out, the next sequential data elements are requested from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user’s buffer.” *Id.* at 15:22–25.

E. The Challenged Claims

Petitioner challenges independent method claim 1, claims 2–4 that depend directly or indirectly from claim 1, independent system claim 5, claims 6–8 that depend directly from claim 5, independent computer-program-product claim 9, and claims 10–12 that depend directly or indirectly from claim 9. Pet. 6, 10–68. Claims 1 and 9 exemplify the challenged claims and read as follows (with formatting added for clarity and with bracketed numbers and letters added for reference purposes):¹

1. [1.a] A method for distributing over the Internet, from a server system to one or more user systems, a pre-recorded audio

¹ We use the same numbers and letters that Petitioner uses to identify the claim language. *See* Pet. 14–50.

or video program stored in digitally encoded form on computer-readable media, the method comprising:

[1.b] reading, by at least one computer of the server system, the pre-recorded audio or video program from the computer-readable media;

[1.c] supplying, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate;

[1.d] serially identifying the media data elements, said serial identification indicating a time sequence of the media data elements;

[1.e] storing the media data elements in a data structure under the control of the server system;

[1.f] receiving requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial identifiers of the requested one or more media data elements, each received request originating from a requesting user system of the one or more user systems; and

[1.g] responsive to the requests, sending, by the server system, the one or more media data elements having the one or more specified serial identifiers, to the requesting user systems corresponding to the requests;

wherein

[1.h] the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection;

[1.i] each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow[s];

[1.j] the one or more media data element[s] sent are selected without depending on the server system maintaining

a record of the last media data element sent to the requesting user systems;

[1.k] all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests; and

[1.l] all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein.

9. [9.a] A computer program product for distributing over the Internet from a server system comprising at least one computer to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media, the computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions comprising:

[9.b] instructions executable to cause one of the at least one computers

to read the pre-recorded audio or video program from the computer-readable media,

[9.c] to supply, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate,

[9.d] to serially identify the media data elements, said serial identification indicating a time sequence of the media data elements, and

[9.e] to store the media data elements in a data structure under the control of the server system;

[9.f] instructions executable to cause one of the at least one computers to receive requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial identifiers of

the one or more media data elements, each received request originating from a requesting user system of the one or more user systems; and

[9.g] instructions executable to cause one of the at least one computers to send, responsive to the requests, the one or more media data elements having the one or more specified serial identifiers to the requesting user systems corresponding to the requests;

wherein,

[9.h] the data connection between the server system and each requesting user systems [sic] has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection;

[9.i] each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allows;

[9.j] the one or more media data elements sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems;

[9.k] all of the media data elements that are sent by the server to the one or more user systems are sent in response to the requests; and

[9.l] all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein.

Ex. 1001, 16:36–17:15, 18:13–65.

F. The Asserted References

For its challenges, Petitioner relies on the following references:

Name	Reference	Exhibit
Carmel	US 6,389,473 B1, issued May 14, 2002 (based on an application filed March 24, 1999)	1004

Name	Reference	Exhibit
Shteyn	US 7,529,806 B1, issued May 5, 2009 (based on an application filed November 4, 1999)	1008

Pet. 6. Petitioner asserts that each reference qualifies as prior art “under at least 35 U.S.C. § 102(e).” *Id.* at 6, 54; *see* 35 U.S.C. § 102(e) (2006).²

Patent Owner does not dispute that each reference qualifies as prior art. *See, e.g.*, Resp. 21–61; Sur-reply 8–26.

G. The Asserted Challenges to Patentability

Petitioner asserts the following challenges to patentability:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
1–12	103(a)	Carmel
1–12	103(a)	Carmel, Shteyn

Pet. 6, 10–68.

H. Testimonial Evidence

To support its challenges, Petitioner relies on the Declaration of Henry Houh, Ph.D. (Exhibit 1002)³ and the Supplemental Declaration of Henry Houh, Ph.D. (Exhibit 1030). Dr. Houh states, “I have been retained by counsel for” Petitioner to analyze the ’824 patent and “provide my opinions regarding the patentability of claims 1–12 of the ’824 patent.”

² The Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112-29, 125 Stat. 284 (2011), amended 35 U.S.C. § 102 and § 103 effective March 16, 2013. Because the effective filing date of the challenged claims predates the AIA’s amendments to § 102 and § 103, this decision refers to the pre-AIA versions of § 102 and § 103.

³ Dr. Houh’s initial declaration (Exhibit 1002) includes two different sets of paragraphs numbered 46–55. To the extent necessary to distinguish between these two sets in our analysis below, we additionally reference the page number in the bottom-right corner of the affected pages.

Ex. 1002 ¶ 1; *see* Ex. 1030 ¶ 3. Dr. Houh also states, “I was awarded a Ph.D. degree in Electrical Engineering and Computer Science in February 1998 from Massachusetts Institute of Technology (MIT)” and “also received a Master of Science (M.S.) in Electrical Engineering and Computer Science (February 1991), a Bachelor of Science in Electrical Engineering and Computer Science (June 1989) and a Bachelor of Science in Physics (February 1990) from MIT.” Ex. 1002 ¶ 4.

To support its positions, Patent Owner relies on the Declaration of W. Leo Hoarty (Exhibit 2006). Mr. Hoarty states, “I have been retained by counsel for” Patent Owner to “provide opinions on certain issues in connection with” the ’824 patent. Ex. 2006 ¶ 2. Mr. Hoarty also states, “I have approximately 40 years of experience in the field of” the ’824 patent, including “multimedia computer communications systems, and more particularly delivering streaming media, such as audio and video, on the internet (which started to become a widespread practice beginning nearly 30 years ago).” *Id.* ¶ 5.

Additionally, Petitioner submits Mr. Hoarty’s deposition testimony, and Patent Owner submits Dr. Houh’s deposition testimony. *See* Ex. 1029 (July 6, 2023, Hoarty Dep. Tr.); Ex. 2005 (Apr. 10, 2023, Houh Dep. Tr.);⁴ Ex. 2010 (Aug. 25, 2023, Houh Dep. Tr.).

I. Burden

In an *inter partes* review, a petitioner bears the burden of persuasion to prove “unpatentability by a preponderance of the evidence.” *Dynamic*

⁴ Patent Owner submits a duplicate of Exhibit 2005 as Exhibit 2009.

Drinkware, LLC v. Nat'l Graphics, Inc., 800 F.3d 1375, 1378 (Fed. Cir. 2015) (quoting 35 U.S.C. § 316(e)); *see* 37 C.F.R. § 42.1(d) (2023).

III. PATENTABILITY ANALYSIS

A. Legal Principles: Obviousness

A patent may not be obtained “if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” 35 U.S.C. § 103(a) (2006). An obviousness analysis involves underlying factual inquiries including (1) the scope and content of the prior art; (2) differences between the claimed invention and the prior art; (3) the level of ordinary skill in the art; and (4) where in evidence, objective indicia of nonobviousness, such as commercial success, long-felt but unsolved needs, and failure of others.⁵ *Graham v. John Deere Co.*, 383 U.S. 1, 17–18, 35–36 (1966); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1047–48 (Fed. Cir. 2016) (en banc). When evaluating a combination of references, an obviousness analysis should address “whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007).

We analyze the obviousness issues according to these principles.

B. Level of Ordinary Skill in the Art

Factors pertinent to determining the level of ordinary skill in the art include (1) the educational level of the inventor; (2) the type of problems encountered in the art; (3) prior-art solutions to those problems; (4) the

⁵ Patent Owner does not address objective indicia of nonobviousness. *See, e.g.*, Resp. 21–61; Sur-reply 8–26.

rapidity with which innovations are made; (5) the sophistication of the technology; and (6) the educational level of workers active in the field. *Envtl. Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 696–97 (Fed. Cir. 1983). Not all factors may exist in every case, and one or more of these or other factors may predominate in a particular case. *Id.* These factors are not exhaustive, but merely a guide to determining the level of ordinary skill in the art. *Daiichi Sankyo Co. v. Apotex, Inc.*, 501 F.3d 1254, 1256 (Fed. Cir. 2007). Moreover, the prior art itself may reflect an appropriate skill level. *Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001).

Petitioner asserts that a person of ordinary skill in the art would have had “a B.S. degree in computer science or electrical engineering (or comparable degree) and two years of experience in networking or streaming media, or a M.S. in computer science or electrical engineering (or comparable degree).” Pet. 7–8. Petitioner also asserts that a “higher level of education or specific skill might make up for less experience, and vice-versa.” *Id.* at 8. Dr. Houh’s testimony supports Petitioner’s assertions. *See* Ex. 1002 ¶ 54;⁶ Ex. 1030 ¶ 12.

Patent Owner “largely agrees” with Petitioner’s description of a person of ordinary skill in the art. Resp. 9. But Patent Owner adds the “clarification” that a skilled artisan would also have had “some theoretical understanding as well as some familiarity with basic internet protocols and tools for working with dynamic content and creating interactive web sites to handle such content.” *Id.*; *see* Sur-reply 3. Mr. Hoarty’s testimony supports Patent Owner’s position. *See* Ex. 2006 ¶ 13.

⁶ This citation references paragraph 54 on bottom-right pages 31 and 32 in Exhibit 1002.

Petitioner urges the Board to reject Patent Owner’s “clarification” because Patent Owner “cites no intrinsic evidence support, and such additions render” the definition “unbounded and confusing with ‘some theoretical understanding’ and ‘tools for working with dynamic content,’ as confirmed during the deposition of WAG’s expert W. Leo Hoarty.” Reply 2 (citing Ex. 1029, 20:17–25:16; Ex. 1030 ¶¶ 13–15).

We adopt Petitioner’s description of a person of ordinary skill in the art because it comports with the technology and claims in the ’824 patent as well as the asserted references. *See, e.g.*, Ex. 1001, 3:43–4:15, 14:48–15:25, 16:36–19:4; Ex. 1002 ¶ 54;⁷ Ex. 1004, 2:7–15, 6:28–31, 6:50–54; Ex. 1030 ¶ 12; Ex. 2005, 38:11–39:8, 43:3–44:5.

Patent Owner cites no intrinsic evidence to support its “clarification” to Petitioner’s description of a person of ordinary skill in the art. *See* Resp. 9; Sur-reply 3; Ex. 2006 ¶ 13.

We note that our analysis would not change if we adopted Patent Owner’s “clarification” to Petitioner’s description of a person of ordinary skill in the art. Patent Owner’s “clarification” increases the level of ordinary skill. *See* Resp. 9; Sur-reply 3; Ex. 2006 ¶ 13. “[I]t is generally easier to establish obviousness under a higher level of ordinary skill.” *Kinetic Concepts, Inc. v. Smith & Nephew, Inc.*, 688 F.3d 1342, 1366 (Fed. Cir. 2012); *see* Tr. 10:12–22.

C. Claim Construction

We construe claim terms “using the same claim construction standard” that district courts use to construe claim terms in civil actions

⁷ This citation references paragraph 54 on bottom-right pages 31 and 32 in Exhibit 1002.

under 35 U.S.C. § 282(b). *See* 37 C.F.R. § 42.100(b). Under that standard, claim terms “are given their ordinary and customary meaning, which is the meaning the term would have to a person of ordinary skill in the art at the time of the invention.” *Power Integrations, Inc. v. Fairchild Semiconductor Int’l, Inc.*, 904 F.3d 965, 971 (Fed. Cir. 2018) (citing *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc)). The meaning of claim terms may be determined by “look[ing] principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17).

Petitioner asserts that the challenged claims “should be interpreted according to their plain and ordinary meaning” and does not propose an explicit construction for any claim language. Pet. 9; *see* Reply 3–6; Tr. 11:2–4.

Patent Owner requests that the Board construe the following claim language:

- the preambles in independent claims 1, 5, and 9;
- “each received request specifying one or more serial identifiers of the requested one or more media data elements” in limitations 1.f, 5.f, and 9.f;
- “the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection,” i.e., limitations 1.h, 5.h, and 9.h;
- “the one or more media data elements sent are selected without depending on the server system maintaining a record of the last media data element sent to the

requesting user systems,” i.e., limitations 1.j, 5.j, and 9.j;
and

- “all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests,” i.e., limitations 1.k, 5.k, and 9.k.

Resp. 10–20; *see* Sur-reply 3–8.

In each independent claim’s preamble, Patent Owner proposes that the Board construe the term “program” to mean “an entire program.”

Resp. 10–11 (emphasis omitted); *see id.* at 14, 19. For limitations 1.f, 1.h, 1.j, 1.k, and the corresponding limitations in claims 5 and 9, Patent Owner does not propose an explicit construction for any claim term or phrase. *See* Resp. 9–10, 12–20; Sur-reply 5–8. Rather, Patent Owner argues about claim scope, i.e., what the limitations do and do not encompass. *See* Resp. 12–20; Sur-reply 5–8.

Petitioner disputes that the term “program” in each independent claim’s preamble means “an entire program.” Reply 3–4. Petitioner also disputes Patent Owner’s contentions concerning claim scope. *Id.* at 4–6.

“[O]nly those terms need be construed that are in controversy, and only to the extent necessary to resolve the controversy.” *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999); *see Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). We determine that no claim language requires an explicit construction to decide whether Petitioner satisfies the “preponderance of the evidence” standard for proving unpatentability.

In the patentability analysis below, we address the preambles in independent claims 1, 5, and 9. *See infra* §§ III.D.2(a), III.D.3. In the

patentability analysis below, we also address claim scope, e.g., whether limitations 1.h, 1.j, and 1.k encompass the prior art and whether collateral estoppel (issue preclusion) precludes Patent Owner from disputing that Carmel teaches certain limitations. *See, e.g., infra* §§ III.D.2(h)(iii), III.D.2(j)(iii), III.D.2(k)(v), III.E.2(a)(iii).

D. Alleged Obviousness over Carmel: Claims 1–12

Petitioner contends that claims 1–12 are unpatentable under § 103(a) as obvious over Carmel. *See* Pet. 6, 10–54. Patent Owner disputes Petitioner’s contentions, e.g., that Carmel teaches limitations 1.h, .1j, 1.k, and the corresponding limitations in claims 5 and 9. *See, e.g.,* Resp. 21–22, 29–51, 53–54; Sur-reply 8–25.

Below, we provide an overview of Carmel. Then, we consider the obviousness issues raised by the parties. For the reasons explained below, we agree with Petitioner that claims 1–12 are unpatentable under § 103(a) as obvious over Carmel.

1. OVERVIEW OF CARMEL (EXHIBIT 1004)

Carmel is a U.S. patent titled “Network Media Streaming,” filed on March 24, 1999, and issued on May 14, 2002. Ex. 1004, codes (12), (22), (45), (54). Carmel states that the invention “relates generally to network data communications, and specifically to real-time multimedia broadcasting over a network.” *Id.* at 1:10–12; *see id.* at code (57).

Carmel explains that “[i]n network broadcasting, data are transmitted over a network in real time from a single transmitting computer to a plurality of clients simultaneously.” Ex. 1004, 1:16–18. The network may include “a LAN, a WAN, an intranet or a public network such as the Internet.” *Id.* at 1:18–20. The data may include “multimedia data, typically comprising

images and sound,” such as “still images, video, graphics, animation or any combination thereof.” *Id.* at 1:20–22, 2:35–36, 4:48.

Carmel explains that the objects of the invention include the following:

- (1) “provid[ing] substantially continuous, high-bandwidth data streaming over a network using common, existing server and network infrastructure”;
- (2) “provid[ing] data broadcasting capability, particularly for multimedia data, without the need for a dedicated broadcast computer system”; and
- (3) “enabl[ing] a personal computer to remotely broadcast a multimedia program through an Internet service provider (ISP) using common, universally-supported Internet communication protocols.”

Ex. 1004, 1:51–58, 1:63–67.

Carmel attempts to achieve those objects with a system and method for “real-time broadcasting from a transmitting computer to one or more client computers over a network.” Ex. 1004, 3:25–39, 14:19–32, 15:63–16:8, code (57). The transmitting computer “generates a data stream . . . divided into a sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed.” *Id.* at 2:1–6. Preferably, “each segment or slice is contained in a separate, respective file,” and each file “includes one or more time stamps, indicating a real time at which the data in the file were recorded or an elapsed time relative to the beginning of [the] stream.” *Id.* at 2:22–23, 7:27–31. “Each slice is preferably assigned a respective slice index.” *Id.* at 2:6–7; *see id.* at 7:27–28.

The transmitting computer “uploads the sequence of slices to” a network server “substantially in real time” and “preferably using an Internet protocol, most preferably the File Transfer Protocol (FTP).” Ex. 1004, 2:7–10; *see id.* at 3:60–62, 5:19–21, 6:50–54. The client computers “download the data stream from” the network server “preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP)” or another protocol “similarly known in the art.” *Id.* at 2:11–15; *see id.* at 3:63–66, 5:25–28, 7:5–9, 10:37–39, 11:1–2, 15:18–21.

The transmitting computer and the client computers may “monitor the uploading and downloading of data to and from the server, respectively, in order to determine the amount of time required to convey each slice and to verify that the slices are conveyed at a sufficient rate.” Ex. 1004, 2:51–56; *see id.* at 3:8–11, 7:35–40, 10:55–56, 11:9–11. “When the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer.” *Id.* at 2:56–59.

The network server may provide the data slices “at multiple resolution or quality levels.” Ex. 1004, 3:5–6. If so, each level “has a different degree of data compression, and thus corresponds to a different data bandwidth requirement.” *Id.* at 3:6–8. The “compression level of the data is varied” to “adjust the data streaming rate to the available bandwidth” of the data link between the network server and a client computer. *Id.* at 7:45–49; *see id.* at 3:6–11, 4:43–47, 9:6–8. For example, “the compression ratio may be adjusted” to “match the data stream bandwidth to the available link bandwidth.” *Id.* at 11:45–48; *see id.* at 3:18–21. Hence, “steps are taken to

increase the data transmission or reception rate” if “a lag is detected.” *Id.* at 7:40–42.

Carmel’s Figure 2 (reproduced below) depicts a system for real-time broadcasting from a transmitting computer to one or more client computers over a network:

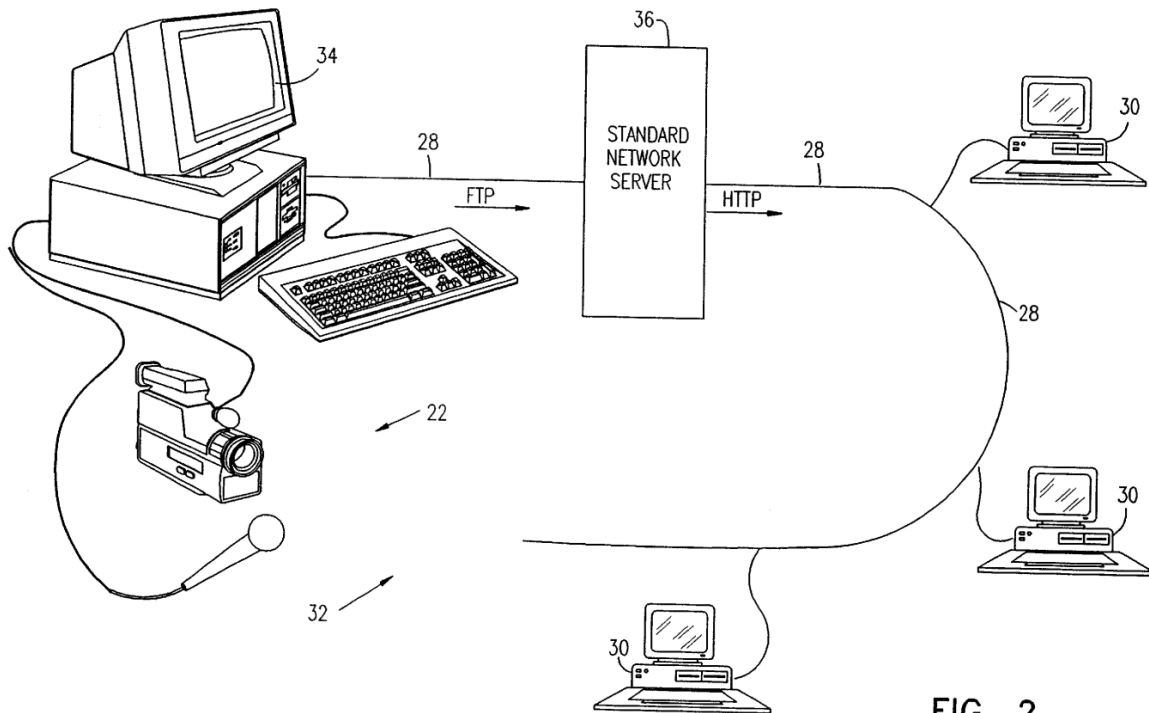


FIG. 2

Figure 2 illustrates computer system 32 including a plurality of client computers 30, transmitting computer 34, and network server 36 that “communicate over network 28, preferably using the well-known Internet Protocol (IP).” Ex. 1004, 6:24–31, Fig. 2; *see id.* at 5:43–45.

Transmitting computer 34 “receives audiovisual input from input devices 22,” e.g., “a video camera and/or microphone” that may “generate a multimedia data stream representing an entertainment or informational program to be transmitted to a plurality of clients” in real time. Ex. 1004, 1:24–28, 3:25–28, 6:32–33, 14:19–20, code (57), Fig. 2. For instance, a streamed program may constitute “an interview program or an entertainment

or sports event.” *Id.* at 6:57–59. But “data inputs of other types may be generated at or by computer 34 using any suitable means,” such as prerecorded media stored “on disk or tape.” *Id.* at 6:33–35, 6:59–60, 9:64–66.

After “preparing the multimedia sequence,” transmitting computer 34 uploads the multimedia sequence to network server 36 over network 28. Ex. 1004, 6:50–54, Fig. 2. Preferably, “the data in the sequence are compressed” and the sequence is “generated and compressed in real time.” *Id.* at 6:54–58. “Clients 30 connect to server 36 and receive the multimedia sequence, substantially in real time.” *Id.* at 7:4–5, Fig. 2.

Carmel’s Figure 3A (reproduced below) depicts an example data structure of a stream with broadcast data:

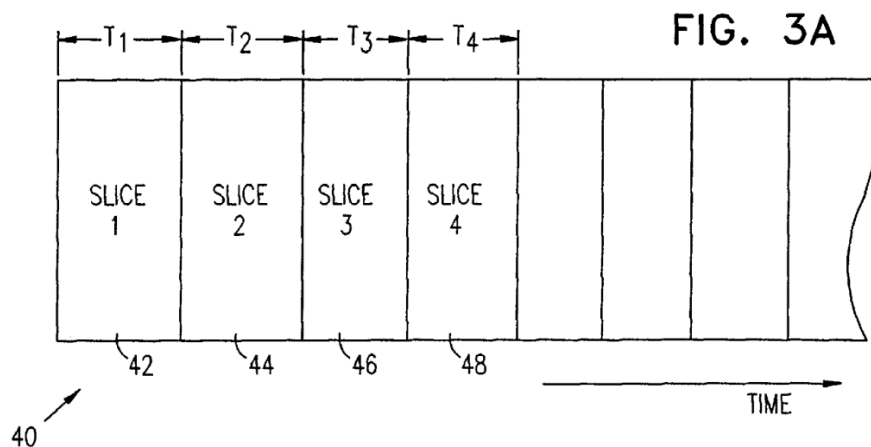


Figure 3A illustrates data stream 40 with broadcast data including “a series of data slices 42, 44, 46, 48, etc.” where “[e]ach slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T₁, T₂, T₃, etc.” Ex. 1004, 5:46–48, 7:18–25, Fig. 3A.

Transmitting computer 34 “stores each slice as a corresponding file, having a running slice index 1, 2, 3 . . . N.” Ex. 1004, 7:27–28.

Transmitting computer 34 may vary the size of each file “by adjusting slice

durations T_1 , T_2 , T_3 , etc.,” e.g., depending on the bandwidth of a data link. *Id.* at 9:31–37; *see id.* at 4:67–5:2, 11:53–55.

Carmel’s Figure 3C (reproduced below) depicts a user interface graphic for use with Figure 3A’s data stream 40:

FIG. 3C

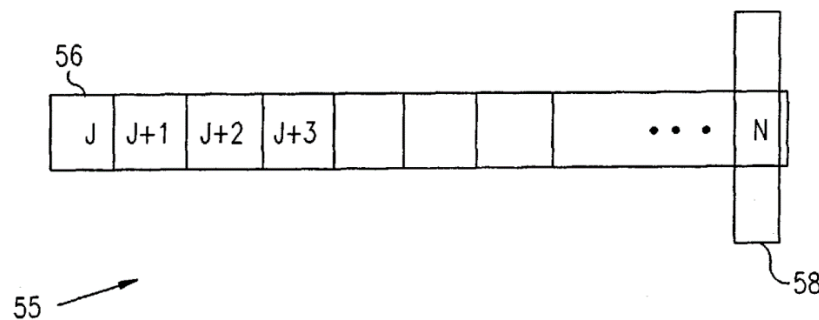


Figure 3C illustrates user interface graphic slider 55 including bar 56 and movable indicator 58, and these graphic elements preferably appear on a client computer’s display. Ex. 1004, 5:54–57, 8:18–23, Fig. 3C. “The symbols J, J+1, J+2, . . . N in the figure are the indices of the slices of stream 40 that are stored on server 36” where “N is the index of the most recent slice” and “J is the index of the earliest stored slice,” i.e., the first slice if server 36 stores all of the data slices in the data stream or the earliest not-yet-erased slice. *Id.* at 8:23–29.

For instance, graphic slider 55 “is displayed by the client” with movable indicator 58 preferably marking “the most recent slice.” Ex. 1004, 8:32–34, 10:41–43, Fig. 3C. By controlling indicator 58, “a user can decide and indicate at which slice of data stream 40 to begin downloading.” *Id.* at 10:42–44, Fig. 3C. If a user decides to begin downloading earlier than the most recent slice, the user may move indicator 58 “to the left along bar 56 to that point, preferably using a mouse or other pointing device, as is known in

the art.” *Id.* at 8:36–39; *see id.* at 4:7–11. Then, “[r]esponsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.” *Id.* at 10:44–47. Moreover, a user may move indicator 58 “back and forth along bar 56 to jump back and forth along stream 40.” *Id.* at 8:39–41.

For “a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user.” Ex. 1004, 10:48–50. Client 30 may use time stamps in the data stream “to synchronize the data” so that “the multimedia sequence is played back just as it was input at computer 34.” *Id.* at 10:50–52; *see id.* at 2:15–17.

Carmel's Figure 6A (reproduced below) depicts steps in a method that a client computer performs when downloading Figure 3A's data stream from a server:

FIG. 6A

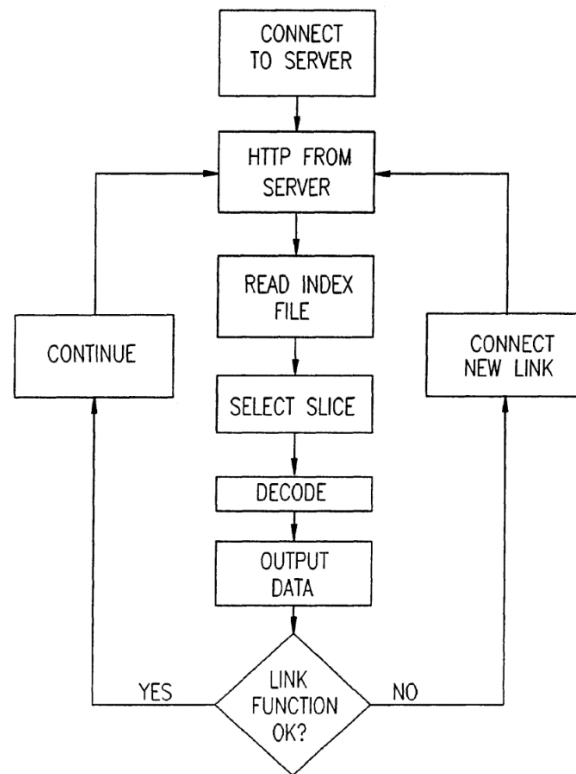


Figure 6A depicts “a method of downloading broadcast data from a server to a client” according to an embodiment of Carmel’s invention and shows steps that a client performs “in downloading and playing back” data stream 40 in Figure 3A. Ex. 1004, 6:3–6, 10:24–27, Figs. 3A, 6A.

Among other things, Figure 6A shows the following steps: “Connect to Server,” “HTTP from Server,” “Read Index File,” “Select Slice,” “Decode,” and “Output Data.” Ex. 1004, Fig. 6A; *see id.* at 10:35–47.

Carmel's Figure 3D (reproduced below) depicts another example data structure of a stream with broadcast data:

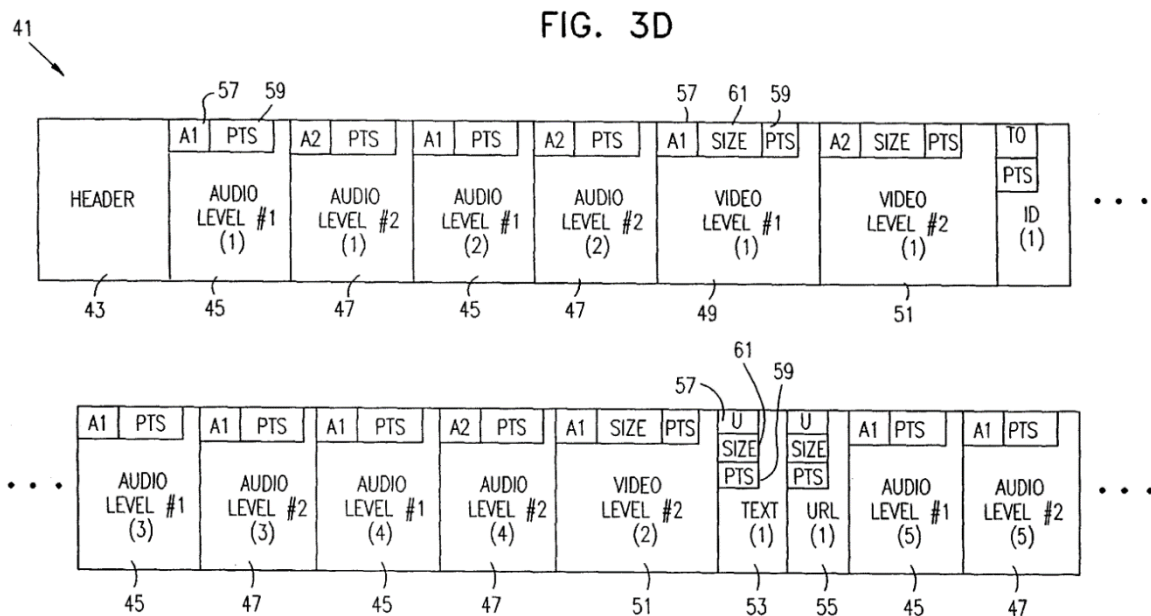


Figure 3D illustrates multilevel data stream 41 with broadcast data including audio slices 45 and 47, video slices 49 and 51, text slice 53, and URL slice 55. Ex. 1004, 5:58–60, 8:42–47, Fig. 3D. “Each slice is preferably identified by a level identifier 57, a presentation time stamp (PTS) index 59 and, as appropriate, a size identifier 61.” *Id.* at 8:47–50, Fig. 3D. Multilevel data stream 41 also includes header 43 containing information indicating, among other things, “formats of the data in the stream,” “the duration of the multimedia sequence represented by the stream,” and “a description of the available stream levels and associated data sizes.” *Id.* at 8:51–55, Fig. 3D.

As Figure 3D shows, each slice in multilevel data stream 41 includes multimedia data at two quality levels “identified as level #1 and level #2,” although “a larger number of levels may also be used.” Ex. 1004, 8:56–59, Fig. 3D. Typically, “the audio and video data in level #1” (slices 45 and 49) are “more highly compressed” and have a smaller “data volume” than the

audio and video data in level #2 (slices 47 and 51). *Id.* at 8:59–63; *see id.* at 9:3–5. The level #1 slices “can therefore be transmitted over a lower-bandwidth data link” than the level #2 slices “while maintaining the required slice timing indicated by time stamps 59.” *Id.* at 8:62–66.

To download multilevel data stream 41, a client computer (1) “connects to the server, generally using a single HTTP link,” (2) reads header 43, and (3) preferably makes “an initial assessment of the link bandwidth.” Ex. 1004, 10:67–11:4, Fig. 6B; *see id.* at 9:49–52. Then, the client computer “selects one of the available quality levels in the stream.” *Id.* at 11:4–5, Fig. 6B. “Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level.” *Id.* at 11:5–7, Fig. 6B. “The slices are received, decoded and output by the client.” *Id.* at 11:7–8, Fig. 6B.

Carmel's Figure 6B (reproduced below) depicts steps in a method that a client computer performs when downloading Figure 3D's multilevel data stream from a server:

FIG. 6B

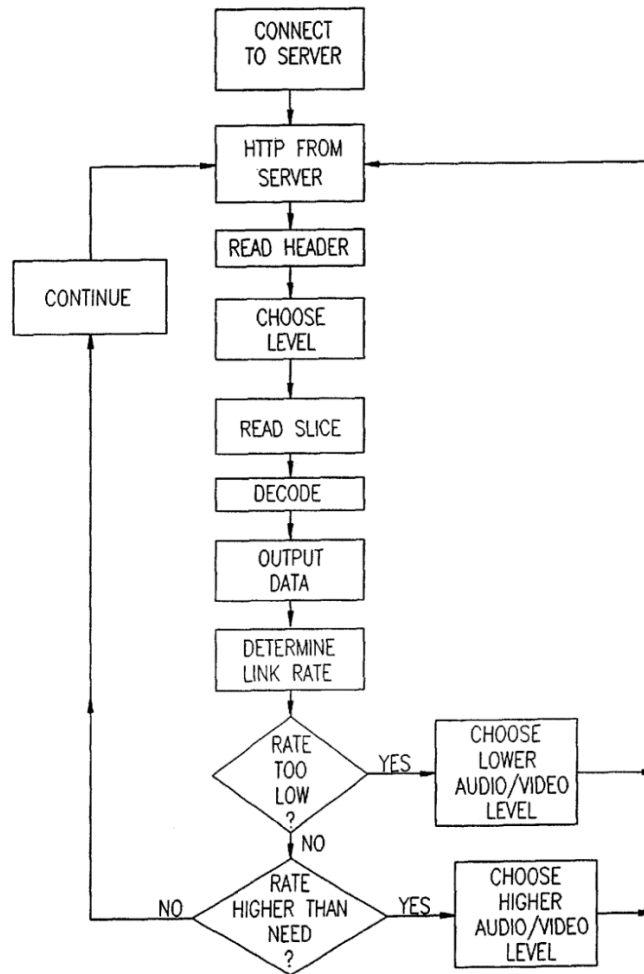


Figure 6B depicts “a method of downloading broadcast data from a server to a client” according to another embodiment of Carmel’s invention and shows steps that a client performs “in downloading and playing back” multilevel data stream 41 in Figure 3D. Ex. 1004, 6:7–10, 10:64–67, Figs. 3D, 6B; *see id.* at 9:8–9.

Among other things, Figure 6B shows the following steps: “Connect to Server,” “HTTP from Server,” “Read Header,” “Choose Level,” “Read Slice,” “Decode,” and “Output Data.” Ex. 1004, Fig. 6B; *see id.* at 10:67–11:8.

For producing a data stream with data slices and then downloading the data stream, Carmel discloses two preferred embodiments as follows: first, data stream 40 in Figure 3A with each data slice contained in a separate file where a client downloads the data stream according to Figure 6A; and second, multilevel data stream 41 in Figure 3D with all data slices contained in a single file where a client downloads the data stream according to Figure 6B. Ex. 1004, 5:46–48, 5:58–60, 6:3–10, 7:18–22, 7:27–28, 8:42–44, 10:24–27, 10:64–67, Figs. 3A, 3D, 6A–6B. This decision refers to the first preferred embodiment as the 3A-6A “separate files” embodiment and the second preferred embodiment as the 3D-6B “single file” embodiment. *See, e.g., infra* § III.D.2(k)(v).c.

2. INDEPENDENT CLAIM 1

(a) Preamble 1.a

Claim 1 recites “[a] method for distributing over the Internet, from a server system to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media.” Ex. 1001, 16:36–39.

Petitioner contends that Carmel teaches claim 1’s preamble because Carmel contains the following disclosures:

- (1) “distributing a data stream from a server system (server 36) to one or more user systems (clients 30) over a network (network 28)”;
- (2) using the Internet when distributing the data stream;

- (3) the “data stream includes an audio or video program”;
and
- (4) the “audio or video program” may be “prerecorded and stored on a disk or a tape,” e.g., “a prerecorded sequence . . . [to] be broadcast[ed].”

Pet. 14–16 (alterations by Petitioner) (emphasis omitted) (quoting Ex. 1004, 6:59–60).

As discussed above, Patent Owner proposes that the Board construe the term “program” in claim 1’s preamble to mean “an entire program.” Resp. 10–11 (emphasis omitted); *see id.* at 14, 19; *supra* § III.C. But Patent Owner does not argue that Carmel fails to teach “an entire program.” *See, e.g.,* Resp. 21–51; Sur-reply 8–25. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378.

Generally, a preamble does not limit a claim. *Allen Eng’g Corp. v. Bartell Indus., Inc.*, 299 F.3d 1336, 1346 (Fed. Cir. 2002). We need not decide whether claim 1’s preamble limits the claim because we agree with Petitioner that Carmel teaches claim 1’s preamble. *See* Pet. 14–17; Ex. 1002 ¶¶ 53–54, 62–67.⁸

Specifically, Carmel discloses computer system 32 including a plurality of client computers 30, transmitting computer 34, and network server 36 that “communicate over network 28, preferably using the well-known Internet Protocol (IP).” Ex. 1004, 6:24–31, Fig. 2; *see id.* at 5:43–45;

⁸ This citation references paragraphs 53 and 54 on bottom-right pages 34 and 35 in Exhibit 1002.

Ex. 1002 ¶¶ 53–54, 63–65.⁹ Transmitting computer 34 “receives audiovisual input from input devices 22,” e.g., “a video camera and/or microphone” that may “generate a multimedia data stream representing an entertainment or informational program to be transmitted to a plurality of clients” in real time. Ex. 1004, 1:24–28, 3:25–28, 6:32–33, 14:19–20, code (57), Fig. 2; *see* Ex. 1002 ¶¶ 66–67. Transmitting computer 34 may also receive audiovisual input in the form of “a prerecorded sequence” stored “on disk or tape.” Ex. 1004, 6:33–35, 6:59–60, 9:64–66; *see* Ex. 1002 ¶ 67.

For instance, a streamed program may constitute “an interview program or an entertainment or sports event.” Ex. 1004, 6:57–59. A client computer downloads and reconstructs a streamed program “for the appreciation of a user.” *Id.* at 10:44–50; *see* Ex. 1002 ¶ 98.

Because Carmel discloses that a streamed program may constitute “an interview program” obtained from “a prerecorded sequence” and then downloaded and reconstructed by a client computer “for the appreciation of a user,” Carmel teaches distributing “an entire program” according to Patent Owner’s proposed construction of the term “program” in claim 1’s preamble. Ex. 1004, 6:32–35, 6:57–60, 9:64–66, 10:44–50; *see id.* at 1:24–28; Ex. 1002 ¶¶ 66–67, 98; Resp. 11.

(b) Limitation 1.b

Claim 1 recites “reading, by at least one computer of the server system, the pre-recorded audio or video program from the computer-readable media.” Ex. 1001, 16:40–42 (limitation 1.b).

⁹ This citation references paragraphs 53 and 54 on bottom-right pages 34 and 35 in Exhibit 1002.

Petitioner contends that Carmel teaches limitation 1.b because Carmel contains the following disclosures:

- (1) “[b]roadcast data [is] input to the computer . . . from a video, audio or animation sequence stored on disk or tape,” and “[the] [c]omputer 34 [then] conveys file 40 [i.e., broadcast data] to server 36 . . . using FTP”;
- (2) “[t]he transmitting computer uploads the sequence of slices to the server substantially in real time, preferably using an Internet protocol”; and
- (3) “Computer 34 continues to upload files 42, 44, 46, etc., until data stream 40 is finished or terminated by a user of computer 34. All of the files in the data stream may be saved on server 36 for any desired period of time.”

Pet. 18–19 (alterations by Petitioner) (emphases omitted) (quoting Ex. 1004, 2:7–10, 7:50–53, 9:64–10:3).

Further, Petitioner asserts that an ordinarily skilled artisan would have understood that “the server system would necessarily ‘read’ the data files 40 that are uploaded to the server via an FTP or other protocols from computer 34.” Pet. 20 (citing Ex. 1002 ¶ 73).

Patent Owner makes no arguments specific to limitation 1.b. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.b. *See* Pet. 18–20; Ex. 1002 ¶¶ 69, 71–73.

(c) Limitation 1.c

Claim 1 recites “supplying, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate.” Ex. 1001, 16:43–46 (limitation 1.c).

Petitioner contends that Carmel teaches limitation 1.c because Carmel’s prerecorded audio or video program includes “a series of data slices 42, 44, 46, 48, etc.” where “[e]ach slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T1, T2, T3, etc.” Pet. 21 (emphases omitted) (quoting Ex. 1004, 7:22–25). Petitioner maps the claimed “media data elements” to Carmel’s data slices. *Id.* Petitioner contends that Carmel’s data slices “are preferably compressed” and that “compression is a form of encoding.” *Id.* (citing Ex. 1002 ¶¶ 76–77; Ex. 1004, 2:1–14).

Further, Petitioner asserts that Carmel’s transmitting computer 34 uploads the data slices to network server 36 “such that while any given slice (other than first slice 42) is being created, one or more preceding slices are in the process of being uploaded.” Pet. 22 (emphasis omitted) (quoting Ex. 1004, 7:32–35). Petitioner also asserts that network server 36 stores the data slices “for any desired period of time.” *Id.* at 23 (emphasis omitted) (quoting Ex. 1004, 7:52–53).

Additionally, Petitioner contends that Carmel’s data slices “have a playback rate that can differ based on the server’s encoding or compression rate.” Pet. 23. Petitioner also contends that (1) the network server provides the data slices “at multiple resolution or quality levels” and (2) each level “has a different degree of data compression, and thus corresponds to a different data bandwidth requirement.” *Id.* (emphases omitted) (quoting Ex. 1004, 3:5–8).

Patent Owner makes no arguments specific to limitation 1.c. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.c. *See* Pet. 21–25; Ex. 1002 ¶¶ 74–82.

(d) Limitation 1.d

Claim 1 recites “serially identifying the media data elements, said serial identification indicating a time sequence of the media data elements.” Ex. 1001, 16:47–49 (limitation 1.d).

Petitioner contends that Carmel teaches limitation 1.d because Carmel’s data slices are “preferably time slices” and “[e]ach slice is preferably assigned a respective slice index,” thereby providing a serial identification indicating a time sequence of the slices.” Pet. 25–26 (alteration by Petitioner) (emphasis omitted) (quoting Ex. 1004, 2:5–7).

Patent Owner makes no arguments specific to limitation 1.d. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.d. *See* Pet. 25–27; Ex. 1002 ¶¶ 83–87.

(e) Limitation 1.e

Claim 1 recites “storing the media data elements in a data structure under the control of the server system.” Ex. 1001, 16:50–51 (limitation 1.e).

Petitioner contends that Carmel teaches limitation 1.e because Carmel’s network server 36 stores the data slices in the data stream in “a container (data structure) or memory” within the server for “any desired period of time, as long as the server has sufficient free memory that is accessible to” transmitting computer 34. Pet. 28 (emphasis omitted) (quoting Ex. 1004, 7:52–55). Petitioner contends that Carmel discloses

indexing the data slices stored in network server 36 using, for example, the symbols “J, J+1, J+2, . . . N” where “N is the index of the most recent slice” and “J is the index of the earliest stored slice,” i.e., the first slice if network server 36 stores all of the data slices in the data stream or the earliest not-yet-erased slice. *Id.* (quoting Ex. 1004, 8:23–29).

Patent Owner makes no arguments specific to limitation 1.e. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.e. *See* Pet. 28–29; Ex. 1002 ¶¶ 88–92.

(f) Limitation 1.f

Claim 1 recites as follows:

receiving requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial identifiers of the requested one or more media data elements, each received request originating from a requesting user system of the one or more user systems.

Ex. 1001, 16:52–58 (limitation 1.f).

Petitioner contends that Carmel teaches limitation 1.f because Carmel’s network server 36 receives requests from a client computer that reads an index file for the numbered data slices and requests “such numbered slices.” Pet. 30. As support, Petitioner quotes Carmel’s disclosure as follows:

The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an

appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.

Id. at 30–31 (emphasis omitted) (quoting Ex. 1004, 10:41–47).

According to Petitioner, a user of a client computer may select where in the data stream a download will begin, for example by moving a slider to select a particular data slice based on its index number, e.g., “J, J+1, J+2, . . . N.” Pet. 31–32 (emphasis omitted) (quoting Ex. 1004, 8:21–29, 8:32–38).

Patent Owner contends that the phrase “each received request specifying one or more serial identifiers of the requested one or more media data elements” in limitation 1.f means that “even if the ‘received request’ is for more than one element, each such requested element is identified in the request by the respective serial identifier of that media data element.”

Resp. 12. According to Patent Owner, “[t]he ‘one’ term in the claim language is simply directed to the case of a request for a single element, while the ‘or more’ term contemplates that a request may identify more than one element, by the elements’ respective serial identifiers.” *Id.* at 13.

But Patent Owner does not argue that Carmel fails to teach limitation 1.f. *See, e.g.*, Resp. 22–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.f. *See* Pet. 30–32; Ex. 1002 ¶¶ 93–96. Further, for the reasons explained below for limitation 1.k, Carmel teaches “each received request specifying one or more serial identifiers of the requested one or more media data elements” according Patent Owner’s contention concerning claim scope. *See* Ex. 1002 ¶¶ 95, 98, 128, 130; Ex. 1004, 10:24–63, Fig. 6A; Ex. 1029, 66:17–67:1; Ex. 1030 ¶¶ 32–33, 37; Ex. 2005, 61:18–62:12, 65:10–66:6, 69:10–18,

78:7–10, 79:9–12, 82:6–83:12, 93:23–94:22; *infra* § III.D.2(k)(v).c;
Resp. 12.

(g) Limitation 1.g

Claim 1 recites “responsive to the requests, sending, by the server system, the one or more media data elements having the one or more specified serial identifiers, to the requesting user systems corresponding to the requests.” Ex. 1001, 16:60–63 (limitation 1.g).

Petitioner contends that Carmel teaches limitation 1.g because Carmel’s network server 36 sends data slices to a client computer responsive to a user input and the client computer “reconstructs and outputs the multimedia data for the appreciation of a user.” Pet. 33 (emphasis omitted) (quoting Ex. 1004, 10:44–50). Petitioner contends that Carmel also teaches limitation 1.g because “responsive to the client selection,” network server 36 sends “data slices at the chosen quality level,” and the data slices are “received, decoded and output by the client.” *Id.* at 33–34 (emphasis omitted) (quoting Ex. 1004, 11:4–8). Further, Petitioner asserts that “the server only transmits what the client requests.” *Id.* at 34 (citing Ex. 1002 ¶ 99; Ex. 1004, Fig. 6B).

Patent Owner makes no arguments specific to limitation 1.g. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.g. *See* Pet. 33–35; Ex. 1002 ¶¶ 97–99.

(h) Limitation 1.h

Claim 1 recites “the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate

of the one or more media data elements sent via that connection.” Ex. 1001, 16:64–67 (limitation 1.h).

(i) Petitioner’s Contentions

Petitioner contends that Carmel teaches limitation 1.h for two independent reasons. *See* Pet. 35–38; Reply 7–11. First, Petitioner asserts that Carmel discloses that “[w]hen the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer.” Pet. 35 (alteration by Petitioner) (emphases omitted) (quoting Ex. 1004, 2:56–59). Petitioner asserts that this disclosure describes “the ‘sufficient rate’ during normal streaming operation (uploading and downloading of data to and from the server).” *Id.* at 35–36 (quoting Ex. 1007 (Final Written Decision on Remand in IPR2016-01238), 18).

Second, Petitioner asserts that Carmel discloses “periodically assessing the data connection between the server system and each requesting user system, and based on this assessment, changing the quality level of” the data slices “so that [the] data rate from the server is more rapid than the playback rate.” Pet. 36; *see* Reply 9. As support, Petitioner quotes the following disclosures in Carmel:

- (1) “downloading the sequence includes determining a data bandwidth of the network between the server and the client computer and selecting one of the quality levels responsive to the determined bandwidth”;
- (2) “[e]ach of clients 30 chooses . . . the quality level appropriate to the bandwidth of its link on network 28 to server 36”;

- (3) “[p]eriodically, client 30 makes an assessment of the rate of data transfer over the link from the server and, if necessary, changes the quality level accordingly”; and
- (4) “the compression level of the data is varied, as is likewise described below, so as to adjust the data streaming rate to the available bandwidth over one or more channels between computer 34 and server 36, and/or between server 36 and client 30.”

Pet. 36–37 (alterations by Petitioner) (emphases omitted) (quoting Ex. 1004, 4:43–47, 7:45–49, 9:6–8, 11:9–11); *see* Reply 9 (citing Ex. 1002 ¶ 109; Ex. 1004, 1:50–53, 4:43–47, 9:6–9).

Further, Petitioner asserts that in addition to “the express disclosure in Carmel, it was well known in the art that the data rate or bandwidth capacity can be more rapid than the encoding or playback rate.” Pet. 35 (footnote omitted) (citing Ex. 1002 ¶¶ 102–107); Reply 10 (quoting Pet. 35).

Additionally, based on the Board’s decision in IPR2016-01238 involving Patent Owner and related U.S. Patent No. 8,122,141 B2 (Exhibit 1005, “the ’141 patent”), Petitioner contends that collateral estoppel (issue preclusion) precludes Patent Owner from disputing that Carmel teaches limitation 1.h. Reply 7 (citing Ex. 1007 (Final Written Decision on Remand in IPR2016-01238), 6, 16–23); *see* Pet. 37–38. Specifically, Petitioner references the Board’s finding relative to the following limitation in the ’141 patent’s claim 10: “instructions to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user” (the “server to send” limitation). Reply 7; *see* Ex. 1005, 14:24–28. Petitioner asserts that the Board’s decision in IPR2016-01238 finding that Carmel teaches the “server to send” limitation precludes Patent Owner from

disputing that Carmel teaches limitation 1.h because the issues are “the same.” Reply 7–8. According to Petitioner, both limitations address “transmitting media data elements more rapid than a playback rate.” *Id.* at 7.

(ii) Patent Owner’s Contentions

Patent Owner disputes that Carmel teaches limitation 1.h. *See* Resp. 4–6, 13–17, 29–34; Sur-reply 8–12. In particular, Patent Owner contends that limitation 1.h’s “data rate” concerns “a physical characteristic of a data connection” and references “physical hardware.” Sur-reply 6, 9; *see* Tr. 32:13–14. As support, Patent Owner cites the ’824 patent’s disclosure about “the user’s connection with a 56K modem,” i.e., a modem having “a maximum receive data rate of 56,000 bits per second.” Sur-reply 6; *see* Ex. 1001, 4:47–49, 10:1–2. Patent Owner contends that “the connection itself is expected to have this capability.” Resp. 8, 31–32 (citing Ex. 1001, 9:64–10:6). According to Patent Owner, the “physical characteristics of the ‘data connection’ are unchanging, even though the operating characteristics vary—as explicitly contemplated by the ’824 patent.” Sur-reply 6 (citing Ex. 1001, 2:34–40).

Patent Owner asserts that limitation 1.h does not “require that the instantaneous transfer rate of the connection be faster than the playback rate at all times during transmission of the requested program.” Resp. 13 (emphasis omitted); *see* Sur-reply 8. According to Patent Owner, the ’824 patent “makes clear that there will be reception interruptions with an internet connection.” Resp. 13 (citing Ex. 1001, 12:19–22).

Patent Owner contends that Carmel “does not disclose a connection data rate that will be faster than the playback rate for what is sent over the connection in the course of streaming the requested program.” Resp. 31.

According to Patent Owner, Carmel “discloses that the connection may lag, and turn out to be less rapid than the playback rate, in which case the player may compensate by requesting a stream version with greater data compression.” *Id.*

Regarding Carmel’s disclosure that “the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer,” Patent Owner asserts that limitation 1.h “is not phrased in terms of ‘equal to or faster’ than the playback rate.” Resp. 29 (emphasis omitted). According to Patent Owner, limitation 1.h “literally states that the ‘data connection’ itself ‘has a data rate more rapid than the playback rate’ – not ‘equal to’ ‘or’ ‘more rapid,’ but simply ‘more rapid’ than the playback rate.” *Id.* (emphasis omitted). Patent Owner asserts that Carmel’s “repeated use of the words ‘generally equal to’ the playback rate makes clear that Carmel does not teach to provide a connection that can be relied on to provide throughput more rapid than the playback rate for the transmission of the entire program.” *Id.* at 32; *see* Sur-reply 17 (citing Ex. 1004, 3:39, 3:59, 4:2, 4:24, 4:62, code (57)).

Further, Patent Owner asserts that “Carmel’s disclosure of a mere ‘data rate’ that should be equal to or faster than the playback rate does not disclose that ‘the data connection’ between the server and each client should have a data rate that is generally equal to or faster than the playback rate.” Sur-reply 10 (emphasis omitted). Patent Owner also asserts that Carmel does not tie the “generally equal to or faster” rate to a “single connection” and that Carmel discloses using multiple connections between the server and a client. *Id.*

Regarding Carmel’s disclosure about choosing a “quality level appropriate to the bandwidth,” Patent Owner contends that choosing a “quality level to meet a currently observed bandwidth is not the same as choosing a connection with a data rate more rapid than the playback rate.” Sur-reply 10–11. According to Patent Owner, the “concept of matching the data and transmission bandwidth pervades Carmel and does not teach a connection having a data rate in excess of the playback rate.” *Id.* at 11.

Patent Owner contends that Carmel “teaches away from the modification of specifying a higher connection data rate” by “eradicat[ing] the increase by decreasing the data compression.” Resp. 33–34 (citing *Chemours Co. v. Daikin Indus., Ltd.*, 4 F.4th 1370, 1377 (Fed. Cir. 2021)).

Regarding Petitioner’s assertion that “it was well known in the art that the data rate or bandwidth capacity can be more rapid than the encoding or playback rate,” Patent Owner argues that Petitioner “offers no rationale for why it would have been obvious to have modified Carmel in this regard and fails to address expectation of success.” Resp. 32 (quoting Pet. 36); *see id.* at 33–34; Sur-reply 12.

Further, Patent Owner asserts that “modifying Carmel to provide a connection faster than the playback rate, and sending all elements as fast as that connection will permit, risks overflowing the memory of the player.” Resp. 32. Patent Owner similarly asserts that “consistently sending faster than the playback rate in Carmel would cause Carmel to fail, since Carmel has no disclosure of a buffer.” Sur-reply 12 (citing Resp. 32; Ex. 2006 ¶¶ 18, 106).

Additionally, Patent Owner contends that collateral estoppel does not preclude it from disputing that Carmel teaches limitation 1.h because the

issue whether Carmel teaches the “server to send” limitation in ’141 patent claim 10 differs materially from the issue whether Carmel teaches limitation 1.h. *See* Resp. 4–6; Sur-reply 8–9. Patent Owner asserts that the issues differ because:

- (1) in IPR2016-01238 the Board applied the “broadest reasonable interpretation” standard for claim construction to the “server to send” limitation when deciding that Carmel teaches that limitation; and
- (2) here the Board will apply the *Phillips* standard for claim construction to limitation 1.h when deciding whether Carmel teaches that limitation.

Resp. 4–5.

Patent Owner identifies another reason that the issues differ: The limitations have “different wording” and “different scope.” Sur-reply 9. According to Patent Owner, the “server to send” limitation concerns media data elements sent “at instantaneous points in time,” whereas limitation 1.h’s “data rate” concerns “a physical characteristic of a data connection” and references “physical hardware.” *Id.* at 6, 9.

(iii) Analysis

For the reasons explained below, we (1) agree with Patent Owner that limitation 1.h’s “data rate” concerns “a physical characteristic of a data connection” and references “physical hardware” and (2) agree with Petitioner that Carmel teaches limitation 1.h. *See* Pet. 35–36; Sur-reply 6, 9; Ex. 1001, 4:40–52, 8:17–20, 10:1–2; Ex. 1002 ¶¶ 57, 94, 100–107; Inst. Dec. 34.

a. Claim Scope

Limitation 1.h recites “the data connection between the server system and each requesting user system has a data rate more rapid than the playback

rate of the one or more media data elements sent via that connection.”

Ex. 1001, 16:64–67. Regarding limitation 1.h’s “data rate,” the ’824 patent explains that “the data rate in bits per second that must be transmitted and received in order to listen to or view the material uninterrupted” corresponds to the “bandwidth requirement” of audio or video media. *Id.* at 4:40–43.

The patent describes a data connection slower than the “bandwidth requirement” as “unsatisfactory” for listening or viewing. *Id.* at 4:44–46.

Additionally, the ’824 patent discloses a data connection between a server and a modem having a “maximum receive data rate of 56,000 bits per second.” Ex. 1001, 4:47–49; *see id.* at 10:1–2. The patent explains that audio or video media “encoded for distribution over the Internet may be compressed to be listenable or viewable within such a 56,000 bits per second bandwidth,” i.e., “the highest rate that the data connection” will support. *Id.* at 4:49–52, 8:17–20. Thus, the patent equates the “maximum receive data rate” of the “physical hardware” to a bandwidth capacity.

Further, in a related litigation, Patent Owner asserted that limitation 1.h’s “data rate” does not concern the “actual rate of delivery of data.” *See* Ex. 1015, 10.¹⁰

Accordingly, we agree with Patent Owner that limitation 1.h’s “data rate” concerns “a physical characteristic of a data connection” and references “physical hardware.” Sur-reply 6, 9; *see* Ex. 1001, 4:40–52.

¹⁰ For Exhibit 1015 (Plaintiff’s Responsive Claim Construction Brief), we cite to the page numbers of the brief rather than the page numbers that Petitioner applied to the exhibit.

b. Collateral Estoppel Does Not Apply for Limitation 1.h

Before explaining how Carmel teaches limitation 1.h, we address the threshold question whether collateral estoppel precludes Patent Owner from disputing that Carmel teaches limitation 1.h. “Collateral estoppel protects a party from having to litigate issues that have been fully and fairly tried in a previous action and adversely resolved against a party-opponent.” *Ohio Willow Wood Co. v. Alps S., LLC*, 735 F.3d 1333, 1342 (Fed. Cir. 2013); *see MaxLinear, Inc. v. CF CRESPE LLC*, 880 F.3d 1373, 1376 (Fed. Cir. 2018) (holding that collateral estoppel may arise from an administrative proceeding).

Collateral estoppel “is appropriate only if”:

- (1) “the issue is identical to one decided in the first action”;
- (2) “the issue was actually litigated in the first action”;
- (3) “resolution of the issue was essential to a final judgment in the first action”; and
- (4) the party defending against preclusion “had a full and fair opportunity to litigate the issue in the first action.”

SynQor, Inc. v. Vicor Corp., 988 F.3d 1341, 1353 (Fed. Cir. 2021). The “identity of the issues” litigated in the two proceedings determines “whether collateral estoppel should apply.” *Ohio Willow Wood*, 735 F.3d at 1342. “If the differences between the unadjudicated patent claims and adjudicated patent claims do not materially alter the question of invalidity, collateral estoppel applies.” *Id.*

As for the “identity of the issues” litigated in the two proceedings relating to limitation 1.h, we determine that the issues differ materially. As Patent Owner asserts, the “server to send” limitation concerns media data elements sent “at instantaneous points in time” and “refers to the rate at

which requested media data elements are sent.” Sur-reply 9; Ex. 2001 (Federal Circuit Decision in Appeal 2018-1617), 10 (emphasis omitted); *see* Resp. 7 n.1. In contrast, limitation 1.h’s “data rate” concerns “a physical characteristic of a data connection” and references “physical hardware.” Sur-reply 6, 9. Thus, the issue decided in IPR2016-01238 for the “server to send” limitation differs materially from the issue here for limitation 1.h.

Additionally, in IPR2016-01238 the Board applied the “broadest reasonable interpretation” standard for claim construction to the “server to send” limitation in ’141 patent claim 10 when deciding that Carmel teaches that limitation. *See* Ex. 1007 (Final Written Decision on Remand in IPR2016-01238), 9–10, 16–21. Here, however, the Board applies the *Phillips* standard for claim construction to limitation 1.h when deciding whether Carmel teaches that limitation. *See supra* § III.C.

For these reasons, collateral estoppel does not preclude Patent Owner from disputing that Carmel teaches limitation 1.h.

c. Carmel Teaches Limitation 1.h

Carmel teaches limitation 1.h because Carmel discloses that:

- (1) the transmitting computer and the client computers may “monitor the uploading and downloading of data to and from the server, respectively, in order to determine the amount of time required to convey each slice and to verify that the slices are conveyed at a sufficient rate”; and
- (2) “[w]hen the data stream comprises multimedia data, the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer.”

Ex. 1004, 2:51–59; *see id.* at 3:8–11, 7:35–40, 10:55–56, 11:9–11; Ex. 1002 ¶¶ 57, 101.

These disclosures describe the “sufficient rate” during normal streaming operation, i.e., for uploading data to the server and downloading data from the server to a client. Ex. 1004, 2:51–59; *see* Ex. 1002 ¶ 101. For a live “entertainment or informational program to be transmitted to a plurality of clients” in real time, a client would play the data for the live program at the rate “the data are generated at the transmitting computer,” i.e., the generation rate at the transmitting computer for a live program matches the playback rate at a client for the live program. *See* Ex. 1004, 1:23–28, 2:56–59, 6:50–59. For a prerecorded program, the transmitting computer may generate the data slices as fast or faster than the data slices for a live program. *See id.* at 6:57–60, 9:64–10:3; Inst. Dec. 34. But the playback rate for a prerecorded program matches the playback rate for the live program that produced the prerecorded program.

By disclosing that the “sufficient rate” during normal streaming operation for uploading and downloading multimedia data “should be generally equal to or faster than” the rate “the data are generated at the transmitting computer,” Carmel teaches that a data connection used for uploading data to the server or downloading data from the server to a client would have the capability to operate “equal to or faster” than the generation rate at the transmitting computer for the entire program. And because (1) Carmel teaches that a data connection used for uploading data to the server or downloading data from the server to a client would have the capability to operate “equal to or faster” than the generation rate at the transmitting computer for the entire program, (2) the generation rate at the transmitting computer for a live program matches the playback rate at a client for the live program, and (3) the playback rate for a prerecorded

program matches the playback rate for the live program that produced the prerecorded program, Carmel teaches that the data connection's capability should be "equal to or faster" than the playback rate. Hence, Carmel teaches that "the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection" according to limitation 1.h. *See* Ex. 1002 ¶¶ 100–101.

As Patent Owner asserts, Carmel repeatedly discloses an upload rate or a download rate "generally equal to" the data rate of the stream from the transmitting computer. *E.g.*, Ex. 1004, 3:24–39, 3:55–59, 4:20–24, 4:49–62, 5:7–10, 14:19–32, code (57); *see* Resp. 32. But Carmel also discloses that the upload rate and the download rate for multimedia data "should be generally equal to or faster than" the data rate of the stream from the transmitting computer. Ex. 1004, 2:51–59; *see* Ex. 1002 ¶¶ 57, 101. Carmel's disclosure of an "equal to or faster" data rate teaches a "faster" data rate. *See* Ex. 1002 ¶ 101; Ex. 1004, 2:51–59. That limitation 1.h "is not phrased in terms of 'equal to or faster' than the playback rate," as Patent Owner asserts, does not matter because Carmel teaches both scenarios. *See* Ex. 1002 ¶¶ 100–101; Ex. 1004, 2:51–59; Resp. 29.

As Patent Owner also asserts, Carmel discloses using multiple connections between the server and a client. *E.g.*, Ex. 1004, 2:60–67, 4:16–24, 7:45–49, 10:35–41, 10:59–63, 11:39–41, Fig. 4; *see* Sur-reply 10. But Carmel also discloses using a single connection between the server and a client. Ex. 1004, 7:45–49, 10:35–38, 10:67–11:2, 11:9–11; *see* Ex. 1002 ¶ 94; Ex. 2010, 70:23–71:3. For example, Carmel explains that a client connects to the server "optionally using multiple HTTP links" and typically

“opens one or two HTTP links.” Ex. 1004, 10:35–38; *see* Ex. 1002 ¶ 94. As such, Carmel contemplates that a single link can sometimes have sufficient capacity. Hence, Carmel teaches using a single connection for uploading and downloading multimedia data at a rate “generally equal to or faster than” the data rate of the stream from the transmitting computer. Ex. 1004, 2:51–59, 4:48, 7:45–49, 10:35–38, 10:67–11:2, 11:9–11; *see* Ex. 1002 ¶¶ 57, 94, 101.

We disagree with Patent Owner that Carmel “teaches away from the modification of specifying a higher connection data rate” by “eradicat[ing] the increase by decreasing the data compression.” *See* Resp. 33–34. A “reference does not teach away” if “it merely expresses a general preference for an alternative invention but does not ‘criticize, discredit, or otherwise discourage’ investigation into the invention claimed.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1327 (Fed. Cir. 2009) (quoting *In re Fulton*, 391 F.3d 1195, 1201 (Fed. Cir. 2004)). Carmel does not “criticize, discredit, or otherwise discourage” using a data connection between the server and a client having a data rate or bandwidth capacity more rapid than the playback rate. *See, e.g.*, Ex. 1004, 2:51–59, 4:39–62, 7:36–49, 10:24–11:22, Figs. 6A–6B.

Moreover, Patent Owner’s assertions that “modifying Carmel to provide a connection faster than the playback rate, and sending all elements as fast as that connection will permit, risks overflowing the memory of the player,” and “would cause Carmel to fail” wrongly assume that limitation 1.h’s “data rate” concerns “the instantaneous transfer rate of the connection,” i.e., the rate at which the connection is actually utilized to send or receive data. *See* Resp. 32; Sur-reply 12. As Patent Owner concedes,

however, limitation 1.h’s “data rate” does not concern “the instantaneous transfer rate of the connection.” Resp. 13; Sur-reply 6, 8–9.

d. An Ordinarily Skilled
Artisan’s Knowledge Also
Evidences Obviousness

Additionally, Dr. Houh testifies that “it was well-known in the art that the data rate or bandwidth capacity can be more rapid than the encoding or playback rate.” Ex. 1002 ¶ 102. To support his testimony, Dr. Houh cites prior art disclosing data connections having a data rate or bandwidth capacity more rapid than the encoding or playback rate. *Id.* ¶¶ 103–107 (citing Ex. 1020 (U.S. Patent No. 6,848,004 B1 to Chang et al. (“Chang”)), 10:15–21, 10:45–51, code (57); Ex. 1021 (WO 97/44942 to Kliger et al. (“Kliger”)), 7:2–4, 9:24–27, 9:32–10:4, 12:33–13:15, 13:33–14:2, 15:10–13, 18:21–19:2).¹¹

As an example, Chang discloses a client (1) predicting the bandwidth capacity of a data connection between a server and the client and (2) receiving video content encoded at a rate less rapid than the bandwidth capacity of the data connection. Ex. 1020, 10:15–51, code (57); *see* Ex. 1002 ¶¶ 103–104. In particular, Chang describes certain video content encoded at the following rates in kilobits per second (kbps): 20 kbps, 50 kbps, 80 kbps, and 140 kbps. Ex. 1020, 10:15–17; *see* Ex. 1002 ¶ 103. Chang explains that:

- (1) if the client predicts a bandwidth capacity “less than 28 kbps, then the client receives 20 kbps video content”;

¹¹ For Exhibit 1021 (Kliger), we cite to the page numbers of the publication rather than the page numbers that Petitioner applied to the exhibit.

- (2) if the client predicts a bandwidth capacity “less than 56 kbps, then the client receives 50 kbps video content”;
- (3) if the client predicts a bandwidth capacity “less than 128 kbps, then the client receives 80 kbps video content”;
and
- (4) otherwise, “the client receives 140 kbps video content.”

Ex. 1020, 10:35–37, 10:45–51; *see* Ex. 1002 ¶ 104. The client receives the appropriate video content “according to a linear or non-linear prediction algorithm resulting in bandwidth sensitive rich media content delivery.”

Ex. 1020, code (57).

As another example, Kliger discloses “bandwidth scalability,” i.e., that a client may choose to request or a server may choose to transmit different versions of data objects, such as audio or video content, as “a function of” the bandwidth capacity of the data connection between the server and the client. Ex. 1021, 9:24–25, 9:32–33, 12:33–13:8; *see* Ex. 1002 ¶¶ 105, 107. Kliger explains that the client “wants the delivery-to-consumption ratio to be greater than one so that the throughput (supply) is keeping up with the consumption (demand),” i.e., that the bandwidth capacity of the data connection for the throughput (supply) should exceed the consumption (demand)/playback rate. Ex. 1021, 13:34–14:2; *see id.* at 18:21–19:2; Ex. 1002 ¶¶ 106–107.

Hence, Chang and Kliger corroborate Dr. Houh’s testimony that “it was well-known in the art that the data rate or bandwidth capacity can be more rapid than the encoding or playback rate.” *See* Ex. 1002 ¶¶ 102–107; Ex. 1020, 10:15–51, code (57); Ex. 1021, 9:24–25, 9:32–33, 12:33–13:8, 13:34–14:2, 18:21–19:2. Petitioner properly relies on Chang and Kliger to corroborate Dr. Houh’s testimony about an ordinarily skilled artisan’s

knowledge. *See Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1337–38 (Fed. Cir. 2020); *Ariosa Diagnostics v. Verinata Health, Inc.*, 805 F.3d 1359, 1365 (Fed. Cir. 2015).

Moreover, an ordinarily skilled artisan’s knowledge is “part of the store of public knowledge that must be consulted when considering whether a claimed invention would have been obvious.” *Randall Mfg. v. Rea*, 733 F.3d 1355, 1362 (Fed. Cir. 2013). When a patent challenger relies on a patent or publication to demonstrate an ordinarily skilled artisan’s knowledge, there is “no obligation” to establish a motivation to combine the patent or publication with the reference or references used to show obviousness. *See Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1372–73 (Fed. Cir. 2019). Hence, Patent Owner’s argument that Petitioner “offers no rationale for why it would have been obvious to have modified Carmel” in view of Chang and Kliger does not identify a deficiency in Petitioner’s showing. *See Resp. 32.*

Patent Owner’s arguments boil down to the position that an ordinarily skilled artisan would not have selected a data connection between the server and a client having a data rate or bandwidth capacity more rapid than the playback rate despite Carmel’s disclosures and the skilled artisan’s knowledge. *See Resp. 29–32; Sur-reply 9–12.* According to that position, an ordinarily skilled artisan would have selected a data connection between the server and a client having a data rate or bandwidth capacity insufficient to permit program streaming. For the reasons discussed above and because a “person of ordinary skill is also a person of ordinary creativity” with “common sense, common wisdom, and common knowledge,” we reject that

position. *See KSR*, 550 U.S. at 421; *Elektro Ltd. v. ZAP Surgical Sys., Inc.*, 81 F.4th 1368, 1374 (Fed. Cir. 2023).

e. Summary

For the reasons discussed above, Carmel teaches limitation 1.h, and an ordinarily skilled artisan’s knowledge also evidences obviousness. *See supra* §§ III.D.2(h)(iii).c–III.D.2(h)(iii).d.

(i) Limitation 1.i

Claim 1 recites “each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow[s].” Ex. 1001, 17:1–3 (limitation 1.i).

Petitioner contends that Carmel teaches limitation 1.i because Carmel discloses adjusting “the data streaming rate” of the data slices sent by network server 36 and received by a client computer to “the available bandwidth” of the link between network server 36 and the client computer. Pet. 38 (emphases omitted). As support, Petitioner quotes Carmel’s disclosure as follows:

[T]he slices are provided by the server at multiple resolution or quality levels. Each such level has a different degree of data compression, and thus corresponds to a different data bandwidth requirement. The client or the server monitors the data transfer rate of a data link opened therebetween and selects the level that is appropriate to the link bandwidth. If the monitored data transfer rate changes during transmission, the quality level is preferably reselected accordingly.

Id. at 38–39 (emphases omitted) (quoting Ex. 1004, 3:5–13).

Further, Petitioner asserts that Patent Owner in a related litigation acknowledged that limitation 1.i covers each of the following three scenarios:

- (1) “a theoretical maximum rate of the data connection”;
- (2) “the available bandwidth of the data connection”; and
- (3) “the permitted bandwidth for the data connection.”

Pet. 39–40 (citing Ex. 1012, 13–14).¹²

Petitioner also asserts that Patent Owner’s declarant in a related litigation stated, “‘As fast as the data connection . . . allows’ requires nothing more than simply handing the data to the transport mechanism.” Pet. 40 (emphasis omitted) (quoting Ex. 1013, 9).¹³

Petitioner then contends that Carmel teaches limitation 1.i because Carmel discloses using the Transmission Control Protocol (TCP) as a transport mechanism to send data slices from network server 36 to a client computer. Pet. 40–41 & n.8 (citing Ex. 1002 ¶ 122; Ex. 1004, 6:28–31, 6:52–54; Ex. 1012, 15; Ex. 1013, 9). Petitioner also contends that TCP:

- (1) “was well known in the art as a protocol for reliable transmission of data described by 1981 and published as a ‘Request for Comments’ (‘RFC’) by the Internet Engineering Task Force (‘IETF’)”; and
- (2) “incorporates a method of flow control, which allows ‘the capacity of the internet [to] be reached’ thus maximizing the bandwidth between the TCP endpoints.”

¹² For Exhibit 1012 (Plaintiff’s Responsive Claim Construction Brief), we cite to the page numbers of the brief rather than the page numbers that Petitioner applied to the exhibit.

¹³ For Exhibit 1013 (Declaration of Keith J. Teruya), we cite to the page numbers of the declaration rather than the page numbers that Petitioner applied to the exhibit.

Id. at 41–42 (alteration by Petitioner) (quoting Ex. 1019, 286) (citing Ex. 1002 ¶ 122; Ex. 1018 (RFC 793 “Transmission Control Protocol”); Ex. 1019, 286–92).¹⁴

Patent Owner makes no arguments specific to limitation 1.i. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.i. *See* Pet. 38–42; Ex. 1002 ¶¶ 110–122.

(j) Limitation 1.j

Claim 1 recites “the one or more media data element[s] sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems.” Ex. 1001, 17:4–7 (limitation 1.j).

(i) Petitioner’s Contentions

Petitioner contends that Carmel teaches limitation 1.j because in Carmel’s system a client computer (1) “requests numbered slices from” network server 36 and (2) “maintains a graphical slider to track each slice, including the last media data element, without regard to the server.” Pet. 42. As support, Petitioner quotes Carmel’s disclosure as follows:

[C]lient 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation, but as in the case of transmitting computer 34, a greater number of links may similarly be opened. The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data

¹⁴ For Exhibit 1019 (TCP/IP Illustrated, Vol. 1: The Protocols), we cite to the page numbers of the publication rather than the page numbers that Petitioner applied to the exhibit.

stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc. In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user.

Id. (emphasis omitted) (quoting Ex. 1004, 10:37–50); *see* Reply 12–13 (citing Ex. 1002 ¶¶ 123–126; Ex. 1004, 8:23–41, 10:36–54).

Further, Petitioner asserts that when a client computer connects to network server 36 and “begins to download the data stream,” the client computer first reads an index file “to identify at what point in stream 40 to begin.” Pet. 43 (emphasis omitted) (quoting Ex. 1004, 8:1–4). Petitioner also asserts that a user “may choose to begin downloading data stream 40” from a point earlier than the most recently uploaded data slice. *Id.* (quoting Ex. 1004, 8:7–9); *see* Reply 13. According to Petitioner, this feature provides “client-side control for requesting sequential media data elements” that “does not depend on the server system maintaining a record of the last media data element sent to the requesting user systems.” Pet. 43 (emphases omitted) (citing Ex. 1002 ¶¶ 125–126).

Petitioner asserts that the following disclosure in Carmel reinforces the teaching of client-side control: a client uses the slice indices for the data slices in a data stream in “maintaining synchronization” that “allows the broadcast to go on substantially in real time without the use of special-purpose hardware.” Reply 13 (quoting Ex. 1004, 2:19–21). Petitioner identifies Carmel’s disclosure about “a lack of specialized server software” as further evidence that Carmel teaches limitation 1.j. *Id.* at 12.

Additionally, based on the Board’s decision in IPR2016-01238 involving Patent Owner and the related ’141 patent, Petitioner contends that

collateral estoppel precludes Patent Owner from disputing that Carmel teaches limitation 1.j. Reply 11–12 (citing Ex. 1006 (Final Written Decision in IPR2016-01238), 26); *see* Pet. 43. Specifically, Petitioner identifies the following limitation in ’141 patent claim 15 that Carmel teaches: “said server does not maintain a pointer into a buffer established within said server, for each said user” (the “no pointer” limitation). Reply 11; *see* Ex. 1005, 14:38–40.

Petitioner asserts that the Board’s decision in IPR2016-01238 finding that Carmel teaches the “no pointer” limitation precludes Patent Owner from disputing that Carmel teaches limitation 1.j because the issues are “the same.” Reply 11.

(ii) Patent Owner’s Contentions

Patent Owner disputes that Carmel teaches limitation 1.j. *See* Resp. 34–37; Sur-reply 13–18. In particular, Patent Owner contends that any client-side control reflected by Carmel’s graphic slider 55 “goes only so far as control over selecting a single starting element,” i.e., “the first element of the stream.” Resp. 35–36; *see* Sur-reply 13, 18. Patent Owner contends that Carmel’s client-side control (1) “is not commensurate with the claim language” and (2) “cannot substitute for pointing to where” Carmel teaches limitation 1.j. Resp. 35; Sur-reply 13, 18. According to Patent Owner, Petitioner “needs to show individual requests by serial ID in order to negate the otherwise certain conclusion that the server must be determining the elements to send.” Sur-reply 14.

Further, Patent Owner asserts that Petitioner fails to address “how Carmel determines which element to send every time it sends successive elements following an initial request that specifies the starting element.”

Resp. 36. Patent Owner similarly asserts that Petitioner fails to argue that “the additional elements are sent without the dependency recited in the claim.” *Id.* at 36–37.

Patent Owner asserts that Carmel’s disclosure about “a lack of specialized server software” involves “only software that makes use of open standards, such as HTTP 1.1,” i.e., HTTP version 1.1. Sur-reply 14.

Also, citing Dr. Houh’s deposition testimony, Patent Owner asserts that Dr. Houh “confirmed” that “the server-side process would have information identifying a user’s links and the slices sent to the user.” Sur-reply 15 n.3 (citing Ex. 2010, 90:10–98:25).

Additionally, Patent Owner contends that collateral estoppel does not preclude it from disputing that Carmel teaches limitation 1.j because the issue whether Carmel teaches the “no pointer” limitation in ’141 patent claim 15 differs materially from the issue whether Carmel teaches limitation 1.j. *See* Resp. 4–6, 37 n.4; Sur-reply 8–9. Patent Owner asserts that the issues differ because:

- (1) in IPR2016-01238 the Board applied the “broadest reasonable interpretation” standard for claim construction to the “no pointer” limitation when deciding that Carmel teaches that limitation; and
- (2) here the Board will apply the *Phillips* standard for claim construction to limitation 1.j when deciding whether Carmel teaches that limitation.

Resp. 4–5.

Patent Owner identifies another reason that the issues differ: The “pointer” prohibited by the “no pointer” limitation differs from the “record” prohibited by limitation 1.j, and the limitations have different scope. Sur-reply 13; *see* Resp. 37 n.4. According to Patent Owner, “[t]he term

‘pointer’ is narrower than ‘record,’” and “there are records that are not pointers.” Sur-reply 13.

(iii) Analysis

For the reasons explained below, we agree with Petitioner that Carmel teaches limitation 1.j. *See* Pet. 42–43; Reply 12–13; Ex. 1002 ¶¶ 123–126.

a. Collateral Estoppel Does Not Apply for Limitation 1.j

Before explaining how Carmel teaches limitation 1.j, we address the threshold question whether collateral estoppel precludes Patent Owner from disputing that Carmel teaches limitation 1.j.

We agree with Patent Owner that collateral estoppel does not preclude it from disputing that Carmel teaches limitation 1.j because the issue whether Carmel teaches the “no pointer” limitation in ’141 patent claim 15 differs materially from the issue whether Carmel teaches limitation 1.j. *See* Resp. 4–6, 37 n.4; Sur-reply 8–9. As Patent Owner asserts, “[t]he term ‘pointer’ is narrower than ‘record.’” *See* Sur-reply 13. Hence, a reference could disclose that a server employs a record-keeping mechanism other than a pointer to select media data elements for streaming. If so, the reference could teach that the server “does not maintain a pointer into a buffer established within said server” according to the “no pointer” limitation and yet fail to teach limitation 1.j.

Additionally, in IPR2016-01238 the Board applied the “broadest reasonable interpretation” standard for claim construction to the “no pointer” limitation when deciding that Carmel teaches that limitation. *See* Ex. 1006 (Final Written Decision in IPR2016-01238), 9–10, 24–26. Here, however, the Board applies the *Phillips* standard for claim construction to

limitation 1.j when deciding whether Carmel teaches that limitation. *See supra* § III.C.

For these reasons, collateral estoppel does not preclude Patent Owner from disputing that Carmel teaches limitation 1.j.

b. Carmel Teaches Limitation 1.j

Carmel discloses that when a client computer connects to network server 36 and “begins to download the data stream,” the client computer first reads an index file “to identify at what point in stream 40 to begin.” Ex. 1004, 8:1–4; *see* Ex. 1002 ¶ 125. Carmel also discloses that a user “may choose to begin downloading data stream 40” from a point earlier than the most recently uploaded data slice. Ex. 1004, 8:7–9; *see id.* at 15:3–7; Ex. 1002 ¶ 125.

For instance, graphic slider 55 “is displayed by the client” with movable indicator 58 preferably marking “the most recent slice.” Ex. 1004, 8:32–34, 10:41–43, Fig. 3C; *see* Ex. 1002 ¶¶ 86–87, 95–96, 124. By controlling indicator 58, “a user can decide and indicate at which slice of data stream 40 to begin downloading.” Ex. 1004, 10:42–44, Fig. 3C; *see* Ex. 1002 ¶¶ 95, 124. If a user decides to begin downloading earlier than the most recent slice, the user may move indicator 58 “to the left along bar 56 to that point, preferably using a mouse or other pointing device, as is known in the art.” Ex. 1004, 8:36–39; *see id.* at 4:7–11; Ex. 1002 ¶¶ 87, 96. Then, “[r]esponsive to a user input,” the client “selects an appropriate starting slice and begins to download and decode (decompress) files” containing data slices. Ex. 1004, 10:44–47; *see* Ex. 1002 ¶¶ 95, 98, 124. Moreover, a user may move indicator 58 “back and forth along bar 56 to jump back and forth

along stream 40.” Ex. 1004, 8:39–41; *see* Resp. 26 (quoting Ex. 1004, 8:32–41).

The ability to move indicator 58 “back and forth along bar 56” and “jump back and forth along stream 40” evidences client-side control and selection “without depending on the server system maintaining a record of the last” data slice sent. *See* Ex. 1002 ¶ 126; Ex. 1004, 8:39–41. As Dr. Houh testifies, “it is unnecessary for the server to track the segment state” because the slider-indicator “allows for random access into the file stream, and playback occurs by requesting sequential segments.” Ex. 1002 ¶ 126; *see id.* ¶ 56; Ex. 2010, 95:20–96:8, 96:21–97:1, 97:24–98:1; *infra* § III.D.2(k)(v).c. As Dr. Houh also testifies, “the server tracking the segment state could conflict and/or cause race conditions with new input selections made by the user.” Ex. 1002 ¶ 126. Accordingly, Carmel teaches that the data slices sent are selected “without depending on the server system maintaining a record of the last” data slice sent. *See* Ex. 1002 ¶¶ 123–126; Ex. 1004, 8:1–4, 8:7–9, 8:32–41, 10:41–47, 15:3–7, Fig. 3C.

The following disclosures in Carmel also evidence client-side control and selection “without depending on the server system maintaining a record of the last” data slice sent:

- (1) a client uses the slice indices for the data slices in a data stream in “maintaining synchronization” that “allows the broadcast to go on substantially in real time without the use of special-purpose hardware”; and
- (2) “the server need not be equipped with any special-purpose hardware or software for real-time data broadcasting, unlike broadcast systems known in the art.”

Ex. 1004, 2:15–21, 6:45–48; *see id.* at 10:50–52. That a client “maintain[s] synchronization” during streaming indicates that the server need not

maintain a record of the last data slice sent. That the server need not include “any special-purpose hardware or software for real-time data broadcasting” indicates that the server need not include hardware or software for tracking data slices during streaming, e.g., for maintaining a record of the last data slice sent.

Consistent with Carmel’s disclosure that the server need not include “any special-purpose hardware or software for real-time data broadcasting,” Carmel discloses adapting a client’s web browser by adding an “applet” to the client’s web browser to enable the client to download a data stream from network server 36, i.e., download according to Figure 6A. Ex. 1004, 7:12–17, 10:24–34; *see* Ex. 1002 ¶ 189; Ex. 2005, 194:15–196:1. Carmel provides an example “applet” for a client, e.g., “LiveFeedClient.exe,” in a software appendix. Ex. 1004, 10:33–34, 13:54–14:4; *see* Ex. 1002 ¶ 68. Carmel does not disclose an analogous “applet” for a server, e.g., one that provides the ability to track data slices during streaming. *See, e.g.*, Ex. 1004, 7:4–17, 10:24–63, 13:54–14:11.

We disagree with Patent Owner that Carmel’s client-side control (1) “is not commensurate with the claim language” and (2) “cannot substitute for pointing to where” Carmel teaches limitation 1.j. *See* Resp. 35; Sur-reply 13, 18. As Dr. Houh testifies, “it is unnecessary for the server to track the segment state” because the slider-indicator “allows for random access into the file stream, and playback occurs by requesting sequential segments.” Ex. 1002 ¶ 126; *see id.* ¶ 56; Ex. 2010, 95:20–96:8, 96:21–97:1, 97:24–98:1; *infra* § III.D.2(k)(v).c. Additionally, an obviousness analysis “need not seek out” the “precise” disclosure of a specific feature in a challenged claim because the analysis “can take account

of the inferences and creative steps that a person of ordinary skill in the art would employ.” *KSR*, 550 U.S. at 418. Based on Carmel’s disclosures, a person of ordinary skill would have understood that “Carmel works with the client tracking which slice files it’s requesting.” *See* Ex. 1002 ¶¶ 125–126; Ex. 1004, 2:15–21, 4:7–11, 6:45–48, 8:1–4, 8:7–9, 8:32–41, 10:41–47, 10:50–52, 15:3–7, Fig. 3C; Ex. 2010, 96:21–97:1, 97:24–98:1.

As for Patent Owner’s contention that Petitioner “needs to show individual requests by serial ID in order to negate the otherwise certain conclusion that the server must be determining the elements to send,” Petitioner makes that showing as explained below for limitation 1.k. *See infra* § III.D.2(k)(v).c; Sur-reply 14.

Additionally, we disagree with Patent Owner that Carmel’s disclosure about “a lack of specialized server software” involves “only software that makes use of open standards, such as HTTP 1.1.” *See* Sur-reply 14. Carmel explains that “the server need not be equipped with any special-purpose hardware or software for real-time data broadcasting, unlike broadcast systems known in the art,” in part because uploading and downloading involves a data stream with data slices and associated slice indices according to Carmel’s novel disclosures. Ex. 1004, 1:3–4, 2:1–21, 6:45–48, 10:27–34, 13:54–14:4; *see id.* at 1:23–47, Fig. 1.

We also disagree with Patent Owner that Dr. Houh “confirmed” that “the server-side process would have information identifying a user’s links and the slices sent to the user.” *See* Sur-reply 15 n.3. During a deposition, Dr. Houh explained that “in the general context of web servers” many servers have the ability to log some information, such as (1) “the host name or the IP address,” (2) “[t]he GET request itself and the type of request,”

(3) “the path or file requested,” (4) “a response code,” and (5) “a time and date.” Ex. 2010, 98:4–25; *see id.* at 99:9–12. When specifically addressing Carmel, however, Dr. Houh said, “Carmel works with the client tracking which slice files it’s requesting,” and “it’s the client that’s tracking which slice files it’s requesting.” *Id.* at 96:21–97:1, 97:24–98:1; *see id.* at 95:20–96:8.

c. Summary

For the reasons discussed above, Carmel teaches limitation 1.j. *See supra* § III.D.2(j)(iii).b.

(k) Limitation 1.k

Claim 1 recites “all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests.” Ex. 1001, 17:8–10 (limitation 1.k).

(i) Petitioner’s Contentions in the Petition

Petitioner contends that Carmel teaches limitation 1.k for several independent reasons. *See* Pet. 43–46; *see* Tr. 23:4–9. First, Petitioner asserts that Carmel’s network server 36 “[r]esponsive to a user input’ for slices (e.g., ‘42, 44, 46, etc.’) provides all such requested slices to the client and no other slices.” Pet. 44 (alteration by Petitioner) (quoting Ex. 1004, 10:44–47). As support, Petitioner cites Carmel’s Figure 6A depicting steps in a method that a client computer performs when downloading Figure 3A’s data stream from network server 36, including the following steps: “HTTP from Server,” “Read Index File,” and “Select Slice.” *Id.* (citing Ex. 1002 ¶ 128; Ex. 1004, 6:3–6, Fig. 6A).

Second, Petitioner asserts that Carmel’s network server 36 “[r]esponsive to the [client] selection,’ transmits requested quality level

files.” Pet. 44 (alterations by Petitioner) (quoting Ex. 1004, 11:5). As support, Petitioner quotes Carmel’s disclosure as follows: “[T]he client selects one of the available quality levels in the stream. Responsive to the selection, server 36 begins to transmit data slices at the chosen quality level. The slices are received, decoded and output by the client.” *Id.* at 44–45 (quoting Ex. 1004, 11:4–8). As additional support, Petitioner cites Carmel’s Figure 6B depicting steps in a method that a client computer performs when downloading Figure 3D’s multilevel data stream from network server 36, including the following steps: “HTTP from Server,” “Choose Level,” and “Read Slice.” *Id.* at 45 (citing Ex. 1002 ¶ 129; Ex. 1004, 6:7–10, Fig. 6B).

Further, Petitioner asserts that HTTP downloading “requires every slice file to be requested individually using an HTTP GET request and the server sends each requested slice to the user system in response.” Pet. 45–46. To support Petitioner’s assertions, Dr. Houh testifies that “it is inherent that when using the HTTP protocol to download each of the slice files, every slice file is requested individually using an HTTP GET request and the server sends each requested slice to the user system in response.” Ex. 1002 ¶ 130.

Alternatively, Petitioner argues that an ordinarily skilled artisan would have been motivated to modify Carmel to request each slice file individually, e.g., by using individual HTTP GET requests for individual slice files. Pet. 46 (citing Ex. 1002 ¶ 131); *see* Tr. 23:4–9, 23:23–24:4. Petitioner explains that requesting each slice file individually would have “improve[d] server efficiency by only sending the data elements requested by the client.” Pet. 46.

(ii) Patent Owner's Contentions in the Response

Patent Owner disputes that Carmel teaches limitation 1.k. *See* Resp. 19–20, 37–51; Sur-reply 18–21. Specifically, Patent Owner asserts that limitation 1.k requires that “each and every element sent must be responsive to a request for that element by its serial ID.” Resp. 20. Patent Owner also asserts that Carmel’s network server 36 sends “a plurality of elements in response to a single user request” from a client. *Id.* at 39; *see id.* at 38–39 (quoting Ex. 1004, 4:7–11, 7:4–5, 8:1–4, 8:9–11, 8:32–34, 10:44–47, 11:2–8).

Regarding the downloading method in Carmel’s Figure 6A, Patent Owner contends that Figure 6A illustrates the following:

- (1) an “HTTP from Server” step where the client performs an “HTTP read” to obtain a “plurality of slices” from the server;
- (2) a “Select Slice” step where the client “selects a slice” from “whatever elements the server returned,” i.e., from the “plurality of slices,” by “checking the index file”; and
- (3) a “Read Index File” step that would be “pointless after the request was made and before playing back the slice.”

Resp. 39–40.

Regarding the downloading method in Carmel’s Figure 6B, Patent Owner contends that Carmel “first says the client selects one of the available quality levels” and “then says that ‘responsive to the selection,’ ‘server 36 begins to transmit data slices at the chosen quality level.’” Resp. 43–44 (emphases omitted) (citing Ex. 1004, 10:64–11:8) (quoting Ex. 1004, 11:5–7).

Additionally, Patent Owner contends that Dr. Houh’s testimony that “it is inherent that when using the HTTP protocol to download each of the

slice files, every slice file is requested individually using an HTTP GET request and the server sends each requested slice to the user system in response” is “flat-out an incorrect assertion.” Resp. 46 (quoting Ex. 1002 ¶ 130). According to Patent Owner, RFC 1945 (published in 1996) specifies standards for HTTP version 1.0 (HTTP 1.0), and RFC 2068 (published in 1997) specifies standards for HTTP version 1.1 (HTTP 1.1). *Id.* at 44 (citing Ex. 2002 (RFC 1945 “Hypertext Transfer Protocol -- HTTP/1.0”); Ex. 2004 (RFC 2068 “Hypertext Transfer Protocol -- HTTP/1.1”); Ex. 2006 ¶¶ 75, 127). Patent Owner asserts that Carmel (filed in 1999) expressly references HTTP 1.1. *Id.* (citing Ex. 1004, 2:27–29); *see id.* at 46 (quoting Ex. 1004, 2:22–28).

Further, Patent Owner asserts that Dr. Houh’s testimony that “every slice file is requested individually using an HTTP GET request and the server sends each requested slice to the user system in response” relates to HTTP 1.0, not HTTP 1.1. Resp. 46 (citing Ex. 2005, 188:25–189:7). According to Patent Owner, HTTP 1.1 “differs considerably” from HTTP 1.0. *Id.* at 45 (citing Ex. 2004 § 8.1.2). Among other differences, Patent Owner identifies the following features in HTTP 1.1 absent from HTTP 1.0:

- (1) “chunked encoding” for modifying the body of a response message to “transfer it as a series of chunks”; and
- (2) “range retrieval” for requesting “one or more sub-ranges” of an entity where a GET request header may specify “a range, in bytes or other units, within a URI [Uniform Resource Identifier], to return a subset of the contents of the entity pointed to by the URI.”

Id. (citing Ex. 2006 ¶¶ 79–80; Ex. 2004 § 14.40); Ex. 2004 §§ 3.6, 14.36.2.

To refute Dr. Houh’s testimony about how an HTTP GET request necessarily operates and show other possibilities, Patent Owner explains that a GET request may identify a “data-producing process” as the target of the request. Resp. 47 (citing Ex. 2004 § 9.3). Patent Owner then proposes two examples based on HTTP 1.1 where Carmel’s network server 36 sends a plurality of slices to client 30 in response to a single GET request from client 30. *Id.* at 47–48.

Patent Owner’s first example involves chunked encoding as follows:

- (1) client 30 “makes a separate request specifying a data compression level”;
- (2) server 36 “accepts the change” and “stops sending any other stream in process in favor of” a stream with a new compression level;
- (3) client 30 “makes a request to” a data-producing process (a data-producing URI) on server 36 with “a starting slice number coded into the request”;
- (4) server 36 “sends a chunked encoding header, then successively reads the slice files, starting with the one specified, and begins sending them one-by-one (with no further GET request), each preceded by a one-line chunk header stating the size of the chunk”;
- (5) the “data-producing process on server 36, once triggered, can open/close transmission links and alternate transmission over them, with no need for additional client requests”;
- (6) server 36 “will close the connection” if client 30 is “inactive or not acknowledging receipt”; and
- (7) client 30 can also send a “stop” instruction.

Resp. 47–48 (citing Ex. 2004 §§ 3.6, 8.1.4).

For the data-producing process in this example, Patent Owner asserts that an ordinarily skilled artisan could have created the process to (1) function as a GET request's target on Carmel's network server 36 and (2) cause the server to "transmit a sequence of slices" in response to the GET request. *See* Resp. 47–48. To support Patent Owner's assertion, Mr. Hoarty explains that a computer program or "script" written in "a server-side scripting language, such as any one of ASP, PHP, ColdFusion, CGI, JSP, etc.," and representing a data-producing process "could easily have served a chunked HTTP response."¹⁵ Ex. 2006 ¶¶ 86–88.

Patent Owner's second example involves range retrieval as follows:

- (1) server 36 "can save the individual slices 41–45 for each quality level, each level in a single indexed file";
- (2) client 30 "can send a request with a beginning range specifier (specifying the beginning slice), leaving out the ending specifier";
- (3) server 36 "will start sending from the specified beginning range"; and
- (4) server 36 "otherwise implements the same level change and stop logic" as in the first example.

Resp. 48 (citing Ex. 1004, 2:23–28).

As for Petitioner's alternative argument about modifying Carmel to request each slice file individually, Patent Owner disagrees that improving server efficiency would have been a reason to do so. *See* Resp. 49–51.

¹⁵ The Microsoft Computer Dictionary contains the following entry for "script": "A program consisting of a set of instructions to an application or a utility program. The instructions usually use the rules and syntax of the application or utility. On the World Wide Web, scripts are commonly used to customize or add interactivity to Web pages." Microsoft Comput. Dictionary (5th ed. 2002) 467 (Exhibit 3003).

First, Patent Owner asserts that “the server must always send all slices to the clients” for proper streaming regardless of any modification to Carmel. *Id.* at 49. According to Patent Owner, “[n]othing is gained” by converting Carmel’s “push” architecture to a “pull” architecture. *Id.*

Second, Patent Owner asserts that the proposed modification would make Carmel less efficient because it would “require more resource and bandwidth utilization” to support not only “the bandwidth requirements of the slices themselves” but also “the overhead associated with the individual requests for each slice.” Resp. 50. According to Patent Owner, the need for “more headroom in the connection” conflicts with Carmel’s objective to “send content over relatively slow connections.” *Id.*

Third, Patent Owner asserts that the proposed modification would convert Carmel’s “push” architecture to a “pull” architecture and require modifying both server-side operations and client-side operations. Resp. 50–51. According to Patent Owner, that “significant” reworking of Carmel “fundamentally changes” Carmel’s principle of operation. *Id.* at 51.

(iii) Petitioner’s Contentions in the Reply

Petitioner contends that Patent Owner wrongly asserts that Carmel’s network server 36 can send “a plurality of elements in response to a single user request” from a client. Reply 15 (quoting Resp. 39). Petitioner asserts that an ordinarily skilled artisan would have needed to write a computer program or “script” to cause Carmel’s network server 36 to send “a plurality of elements in response to a single user request” from a client. *Id.* at 19 (citing Resp. 44–47; Ex. 2006 ¶¶ 82, 86–93).

Petitioner contends that the following disclosures in Carmel teach that a client requests “each slice one after another by identifier from the server”:

- (1) “client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation”;
- (2) “[r]esponsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.”; and
- (3) “a method of downloading broadcast data from a server to a client.”

Reply 15 (alteration by Petitioner) (emphases omitted) (quoting Ex. 1004, 6:3–5, 10:37–39, 10:44–47).

Petitioner contends that the downloading method in Carmel’s Figure 6A concerns a “client request system” where “the client will ‘Read Index File’ to ‘Select Slice’—singular, not plural—from the server, and the client will repeatedly loop requesting each time a single slice from the server.” Reply 16 (citing Ex. 1002 ¶ 128; Ex. 1004, Fig. 6A; Ex. 2005, 60:7–61:6, 74:2–22). To support its contention, Petitioner provides an annotated version of Carmel’s Figure 6A as reproduced below:

FIG. 6A

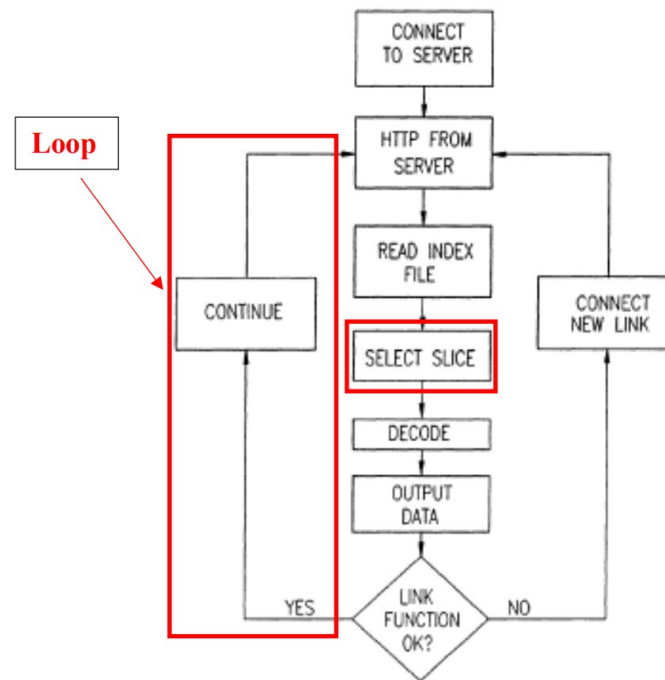


Figure 6A depicts “a method of downloading broadcast data from a server to a client” according to an embodiment of Carmel’s invention and shows steps that a client performs “in downloading and playing back” data stream 40 in Figure 3A. Ex. 1004, 6:3–6, 10:24–27, Figs. 3A, 6A; *see* Ex. 1002 ¶¶ 95, 128. The above annotated version of Figure 6A includes the following annotations:

- (1) a red box around the “Select Slice” step;
- (2) a red box around (i) the arrow from the “Yes” branch of the “Link Function OK?” determination to the “Continue” step, (ii) the “Continue” step, and (iii) the arrow from the “Continue” step to the “HTTP from Server” step; and
- (3) the red annotation “Loop” identifying the red box around the “Continue” step and the arrows to and from that step.

See Reply 16.

Petitioner asserts that Patent Owner misreads Figure 6A because, contrary to Patent Owner's contentions, the "HTTP from Server" step does not cause the server to provide a "plurality of slices." Reply 17. Petitioner also asserts that in the "Select Slice" step the client does not select from among an already downloaded "plurality of slices" because the "HTTP from Server" step does not cause the server to provide a "plurality of slices." *Id.* According to Petitioner, Figure 6A shows the "HTTP from Server" step, the "Read Index File" step, and the "Select Slice" step in a repeated loop. *Id.*

Petitioner contends that the downloading method in Carmel's Figure 6B "reinforces the conclusion" that Carmel concerns a "client request system" because Figure 6B "shows the client 'Read Slice' (singular) and then repeatedly looping to select a different slice encoding rate (audio/video level)" if the encoding rate is either too high or too low. Reply 17. To support its contention, Petitioner provides an annotated version of Carmel's Figure 6B as reproduced below:

FIG. 6B

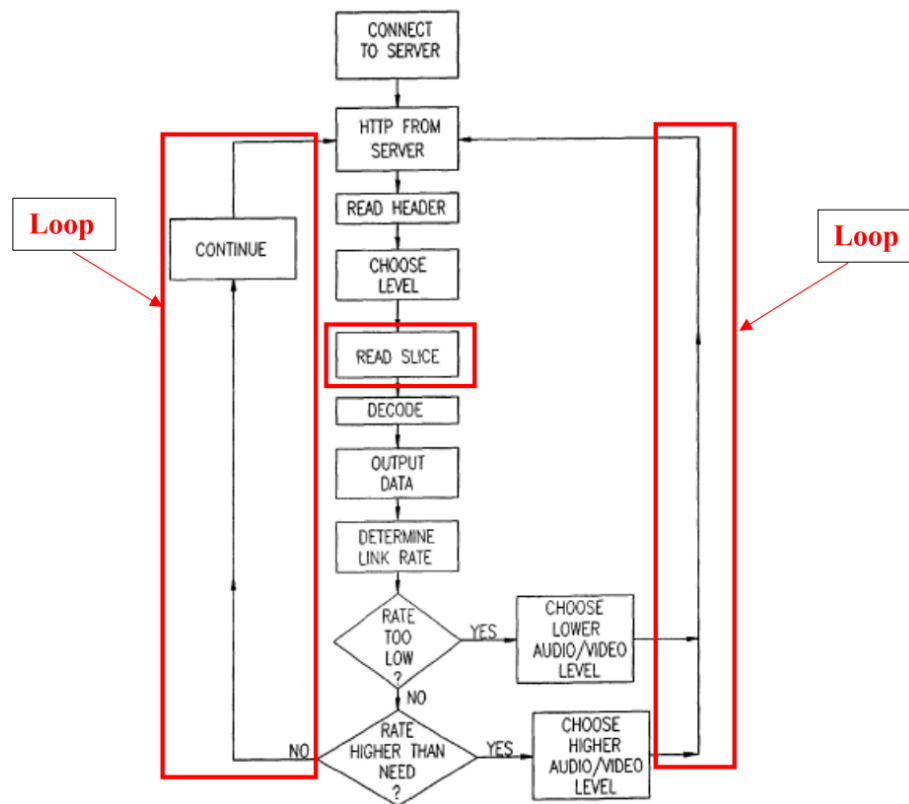


Figure 6B depicts “a method of downloading broadcast data from a server to a client” according to another embodiment of Carmel’s invention and shows steps that a client performs “in downloading and playing back” multilevel data stream 41 in Figure 3D. Ex. 1004, 6:7–10, 10:64–67, Figs. 3D, 6B; *see id.* at 9:8–9; Ex. 1002 ¶¶ 82, 129. The above annotated version of Figure 6B includes the following annotations:

- (1) a red box around the “Read Slice” step;
- (2) a red box around (i) the arrow from the “No” branch of the “Rate Higher than Need?” determination to the “Continue” step, (ii) the “Continue” step, and (iii) the arrow from the “Continue” step to the “HTTP from Server” step;

- (3) the red annotation “Loop” identifying the red box around the “Continue” step and the arrows to and from that step;
- (4) a red box around the arrows from the “Choose Lower Audio/Video Level” and “Choose Higher Audio/Video Level” steps to the “HTTP from Server” step; and
- (5) the red annotation “Loop” identifying the red box around the arrows from the “Choose Lower Audio/Video Level” and “Choose Higher Audio/Video Level” steps to the “HTTP from Server” step.

See Reply 18. According to Petitioner, Figure 6B shows “no way” for a client to tell the server “to switch to a new encoding rate without the client generating a request for it.” *Id.* at 17.

Additionally, Petitioner disputes that Carmel employs “push” architecture rather than “pull” architecture. Reply 20. Petitioner asserts that (1) Patent Owner’s citations to Carmel do not disclose server-side hardware and server-side software for a “push” system and (2) Carmel expressly teaches that “the server need not be equipped with any special-purpose hardware or software.” *Id.* (emphasis omitted) (quoting Ex. 1004, 6:45–47). Petitioner also asserts that the Petition relies on “portions of Carmel” that disclose a “client request system.” *Id.*

Further, Petitioner asserts that if “Carmel does not expressly disclose a client request system, Carmel inherently teaches one.” Reply 20 (citing Ex. 1030 ¶¶ 9–11, 28–51). Petitioner asserts that a GET request is the primary method to fetch data in both HTTP 1.0 and HTTP 1.1. *Id.* at 21. Petitioner asserts that a GET request “means retrieve whatever information (in the form of an entity) is identified by the Request-URI”¹⁶ so “the server

¹⁶ The acronym “URI” stands for “Uniform Resource Identifier.” Ex. 2002 § 5.1.2; Ex. 2004 § 5.1.2.

only sends the data requested (GET) by the client and no more.” *Id.* (quoting Ex. 2004 § 9.3) (citing Ex. 1030 ¶¶ 32–33).

As for Patent Owner’s two examples based on HTTP 1.1 where Carmel’s network server 36 sends a plurality of slices to client 30 in response to a single GET request from client 30, Petitioner asserts that the examples “mischaracterize HTTP.” Reply 19, 21. For the first example involving chunked encoding, Petitioner asserts that chunked encoding applies only “in the context of a single requested slice where the server breaks that single slice into smaller chunks.” *Id.* at 21 (emphasis omitted) (citing Ex. 1030 ¶ 36). Petitioner further asserts that the first example “falls apart when applied to Carmel” because the server sends only “what is requested by the client (via a GET request), i.e., a request for a slice 42 to the server will return the same slice 42, not additional slices 44, 46, etc.” *Id.* at 21–22 (citing Ex. 1002 ¶ 130; Ex. 1029, 66:5–67:1; Ex. 1030 ¶ 37; Ex. 2004 § 3.6).

For the second example involving range retrieval, Petitioner asserts that the second example relies on Carmel’s “single file” embodiment with all slices “contained in a single indexed file,” whereas the Petition relies on Carmel’s “separate files” embodiment with each slice “contained in a separate, respective file.” Reply 22 (citing Ex. 1002 ¶¶ 62–138; Ex. 1004, 2:22–23; Ex. 1030 ¶¶ 42–44). Petitioner also asserts that Patent Owner “misstates the operation of” range retrieval because range retrieval permits only the transfer of “less than one slice file” or “the entire single slice file.” *Id.* at 22–23 (citing Ex. 1030 ¶¶ 45–46).

Further, Petitioner asserts that a range retrieval request according to HTTP 1.1 “may request one or more sub-ranges of the entity, instead of the

entire entity.” Reply 23 (quoting Ex. 2004 § 14.36.2) (citing Ex. 1030 ¶ 47). According to Petitioner, a range retrieval request used in Carmel’s “separate files” embodiment would mean that “the client could request ‘subranges of the entity,’ i.e., a single slice file.” *Id.* (emphasis omitted) (citing Ex. 1030 ¶ 48). Petitioner also asserts that “leaving out the ending specifier,” as Patent Owner proposes, would cause the server to transmit “the full single slice file . . . instead of ‘sub-ranges’ of that single slice file” and would not cause the server to transmit multiple slice files. *Id.* (citing Ex. 1030 ¶ 50).

As for Petitioner’s alternative argument about modifying Carmel to request each slice file individually, Petitioner asserts that improving server efficiency suffices as a reason to do so. Reply 24 (citing *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1381 (Fed. Cir. 2023)). Further, responding to Patent Owner’s assertion that the proposed modification would make Carmel less efficient because it would “require more resource and bandwidth utilization,” Petitioner contends that the ’824 patent contradicts that assertion by explaining that the “TCP manages the transmitted media data elements.” *Id.* (citing Ex. 1001, 8:38–48). Also, Petitioner quotes Patent Owner’s statement from a related litigation as follows: “It is second nature to [an ordinarily skilled artisan] what a TCP connection does with data delivered to it.” *Id.* (quoting Ex. 1012, 15).

Additionally, based on the Board’s decision in IPR2016-01238 involving Patent Owner and the related ’141 patent, Petitioner contends that collateral estoppel precludes Patent Owner from disputing that Carmel teaches limitation 1.k. Reply 14 (citing Ex. 1006 (Final Written Decision in IPR2016-01238), 5–6, 16, 28; Ex. 1007 (Final Written Decision on Remand

in IPR2016-01238), 16–23). Specifically, Petitioner identifies the following limitations in ’141 patent claim 10 that Carmel teaches:

- “instructions to cause the server to receive requests from the user system for one or more media data elements specifying the identifiers of the requested data elements” (the “server to receive” limitation); and
- “instructions to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user” (the “server to send” limitation).

Id.; see Ex. 1005, 14:20–28.

Petitioner asserts that the Board’s decision in IPR2016-01238 finding that Carmel teaches the “server to receive” and the “server to send” limitations precludes Patent Owner from disputing that Carmel teaches limitation 1.k. Reply 14. Petitioner further asserts that Mr. Hoarty “reinforces this estoppel conclusion” because he admits that both the ’141 patent and the ’824 patent claim a “client request system.” *Id.* (citing Ex. 2006 ¶ 29). According to Petitioner, Carmel teaches that a client requests “each media data element by serial identifier from the server which sends such requested elements.” *Id.*

(iv) Patent Owner’s Contentions in the Sur-reply

Regarding the downloading method in Carmel’s Figure 6A, Patent Owner contends that Figure 6A does not show a client requesting individual slices from the server. See Sur-reply 19–20; see *id.* at 21. Further, Patent Owner asserts that “[o]n each loop traversal there may be a singular slice, but this does not reflect whether the slice got there because it was individually requested or simply because the server sent it as part of a

series.” *Id.* at 20. Patent Owner also asserts that the “HTTP from Server” step in Figure 6A refers to “receiving slices pushed from the server.” *Id.* at 21 (emphasis omitted).

Regarding the downloading method in Carmel’s Figure 6B, Patent Owner contends that Figure 6B does not show a client requesting individual slices from the server. *See* Sur-reply 20. Further, Patent Owner asserts that selecting a new encoding rate according to Figure 6B means “asking the server to push elements with a different encoding than the server is currently pushing.” *Id.*

As for Patent Owner’s two examples based on HTTP 1.1 where Carmel’s network server 36 sends a plurality of slices to client 30 in response to a single GET request from client 30, Patent Owner asserts that Petitioner does not dispute that they would work. Sur-reply 22. Responding to Petitioner’s arguments about the two examples, Patent Owner asserts that:

- (1) a GET request is for an “entity”;
- (2) an “entity” is defined as the “payload” of a response to the GET request;
- (3) an “entity” is not synonymous with a “file”; and
- (4) Petitioner wrongly equates an “entity” to a “file.”

Id. at 23–24 (emphasis omitted) (citing Ex. 2004 § 9.3).

For the first example involving chunked encoding, Patent Owner asserts that “each chunk can be a slice.” Sur-reply 24. For the second example involving range retrieval, Patent Owner asserts that a range retrieval request “does not have to be [for] a single file.” *Id.* According to Patent Owner, a range retrieval request would work with “a file structure backed by the plurality of slices” as shown in Carmel’s Figure 3D. *Id.*

Regarding Carmel's teaching that "the server need not be equipped with any special-purpose hardware or software," Patent Owner asserts that:

- (1) HTTP 1.1 is an "open standard" accommodating "routine" scripted server responses; and
- (2) a server "does not need any special purpose hardware" to implement chunked encoding or range retrieval.

Sur-reply 21 (citing Ex. 2006 ¶¶ 86–90).

As for Petitioner's alternative argument about modifying Carmel to request each slice file individually, Patent Owner contends that Petitioner does not explain how processing multiple requests for multiple slice files improves server efficiency over "processing a single request." Sur-reply 25. Patent Owner contends that Petitioner does not address Patent Owner's assertion that "the server must always send all slices to the clients" for proper streaming regardless of any modification to Carmel. *Id.* (citing Resp. 49). Further, responding to Petitioner's position that the "TCP manages the transmitted media data elements," Patent Owner asserts that Petitioner "does not address the point about delays in establishing TCP connections." *Id.*

Additionally, Patent Owner contends that collateral estoppel does not preclude it from disputing that Carmel teaches limitation 1.k because the issue whether Carmel teaches the "server to receive" and the "server to send" limitations in '141 patent claim 10 differs materially from the issue whether Carmel teaches limitation 1.k. *See* Sur-reply 18–19. Patent Owner asserts that the issues differ because limitation 1.k requires that "the server sends no elements other than those requested by serial ID" and '141 patent claim 10 "did not recite this." *Id.* at 19.

(v) Analysis

For the reasons explained below, we (1) disagree with Patent Owner that limitation 1.k requires that “the server sends no elements other than those requested by serial ID” and (2) agree with Petitioner that Carmel teaches limitation 1.k. *See* Pet. 43–46; Resp. 19–20; Reply 15–23; Sur-reply 19; Ex. 1002 ¶¶ 94–96, 98–99, 127–131; Ex. 1030 ¶¶ 27–51.

a. Claim Scope

Limitation 1.k recites “all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests.” Ex. 1001, 17:8–10. In related IPR2022-01412 for the ’824 patent, we determined preliminarily that “[a]lthough the limitation calls for ‘all’ of the media data elements to be sent in response to requests, it does not require every slice to be requested by slice number.” IPR2022-01412, Paper 8 at 38. Patent Owner disputes this interpretation. Resp. 19 n.2; IPR2022-01412, Paper 11 at 21–23. But the claim language supports this interpretation. *See* Ex. 1001, 16:52–63, 17:8–10.

In particular, regarding “the requests” in limitation 1.k, limitation 1.f recites “receiving requests at the server system” with “each received request specifying one or more serial identifiers of the requested one or more media data elements.” Ex. 1001, 16:52–57. Thus, limitation 1.f provides an antecedent basis for “the requests” in limitation 1.k. Further, regarding “the requests” in limitation 1.k, limitation 1.g recites “responsive to the requests, sending, by the server system, the one or more media data elements having the one or more specified serial identifiers.” *Id.* at 16:60–62.

According to claim 1, each “request” received by the server system may specify “one or more serial identifiers of the requested one or more

media data elements,” and the server system responds by sending “the one or more media data elements having the one or more specified serial identifiers.” Ex. 1001, 16:52–57, 16:60–62. Also, claim 1 uses the transitional term “comprising” and, therefore, “does not exclude unrecited elements.” *See id.* at 16:36–39; *Regeneron Pharm., Inc. v. Merus N.V.*, 864 F.3d 1343, 1352 (Fed. Cir. 2017) (citing *Manual of Patent Examining Procedure* § 2111.03). Hence, a first “request” specifying a first “media data element” that results in the server sending the specified element and additional elements and a second “request” specifying a second “media data element” that results in the server sending the specified element and additional elements collectively satisfy limitation 1.k’s requirement that “all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests.”

This interpretation is consistent with the ’824 patent’s specification. *See, e.g.*, Ex. 1001, 3:43–4:15, 14:48–15:25. For the embodiment that Patent Owner contends “corresponds to the claims,” the specification explains as follows:

In another embodiment, the server is connected to the Internet and provisioned as initially described. The server buffer manager, or the media source, provides for sequentially numbering the media data elements. The server buffer manager does not maintain a pointer into the server buffer for each user. Instead, the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received.

Via the use of standard data communications protocol techniques such as TCP, the user computer transmits a request to the server to send one or more data elements, specifying the serial numbers of the data elements. The server responds by sending the requested data elements

Id. at 14:48–60; *see* Resp. 7–8 (citing Ex. 1001, 14:48–15:25). That the “server responds by sending the requested data elements” encompasses the server sending the requested data elements and additional data elements. Indeed, the specification describes “changes, substitutions, and alterations” as “readily ascertainable” by an ordinarily skilled artisan. Ex. 1001, 16:29–31. Further, Federal Circuit case law “counsels against incorporating a feature of a preferred embodiment into the claims.” *Arthrex, Inc. v. Smith & Nephew, Inc.*, 935 F.3d 1319, 1330 (Fed. Cir. 2019).

No party cites the ’824 patent’s prosecution history as relevant to limitation 1.k’s scope. *See, e.g.*, Pet. 43–46; Resp. 19–20, 37–49; Reply 6, 15–23; Sur-reply 18–24.

Based on the claim language and the ’824 patent’s specification, we disagree with Patent Owner that limitation 1.k requires that “the server sends no elements other than those requested by serial ID.” *See* Sur-reply 19.

b. Collateral Estoppel Does Not Apply for Limitation 1.k

Before explaining how Carmel teaches limitation 1.k, we address the threshold question whether collateral estoppel precludes Patent Owner from disputing that Carmel teaches limitation 1.k.

We agree with Patent Owner that collateral estoppel does not preclude it from disputing that Carmel teaches limitation 1.k because the issue whether Carmel teaches the “server to receive” and the “server to send” limitations in ’141 patent claim 10 differs materially from the issue whether Carmel teaches limitation 1.k. *See* Sur-reply 18–19; *see also* Resp. 4–6. As Patent Owner asserts, the “server to send” limitation concerns media data elements sent “at instantaneous points in time” and “refers to the rate at which requested media data elements are sent.” Sur-reply 9; Ex. 2001

(Federal Circuit Decision in Appeal 2018-1617), 10 (emphasis omitted). In contrast, limitation 1.k does not refer to any rate. Ex. 1001, 17:8–10.

Further, Petitioner fails to explain why the differences between limitation 1.k’s response-only mechanism and the request-response mechanism covered by the “server to receive” and the “server to send” limitations “do not materially alter” the question of unpatentability.

Reply 14; *see Ohio Willow Wood*, 735 F.3d at 1342.

Additionally, in IPR2016-01238 the Board applied the “broadest reasonable interpretation” standard for claim construction to the “server to receive” and the “server to send” limitations when deciding that Carmel teaches those limitation. *See* Ex. 1006 (Final Written Decision in IPR2016-01238), 9–10, 15–16; Ex. 1007 (Final Written Decision on Remand in IPR2016-01238), 9–10, 16–21. Here, however, the Board applies the *Phillips* standard for claim construction to limitation 1.k when deciding whether Carmel teaches that limitation. *See supra* § III.C.

For these reasons, collateral estoppel does not preclude Patent Owner from disputing that Carmel teaches limitation 1.k.

c. Carmel Teaches Limitation 1.k

i. Express Teaching

Carmel discloses that the transmitting computer “generates a data stream . . . divided into a sequence of segments or slices of the data, preferably time slices.” Ex. 1004, 2:1–6; *see* Ex. 1002 ¶¶ 53–55.¹⁷ Preferably, “each segment or slice is contained in a separate, respective file,” and each file “includes one or more time stamps, indicating a real time at

¹⁷ This citation references paragraphs 53 to 55 on bottom-right pages 34 to 36 in Exhibit 1002.

which the data in the file were recorded or an elapsed time relative to the beginning of [the] stream.” Ex. 1004, 2:22–23, 7:27–31; *see* Ex. 1002 ¶¶ 55, 130. The transmitting computer “uploads the sequence of slices to” the network server “substantially in real time” and “preferably using an Internet protocol, most preferably the File Transfer Protocol (FTP).” Ex. 1004, 2:7–10; *see id.* at 3:60–62, 5:19–21, 6:50–54; Ex. 1002 ¶¶ 67, 72, 79.

The client computers “download the data stream from” the network server “preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP)” or another protocol “similarly known in the art.” Ex. 1004, 2:11–15; *see id.* at 3:63–66, 5:25–28, 7:5–9, 10:37–39, 11:1–2, 15:18–21; Ex. 1002 ¶¶ 56, 65, 124–125. The client computers “request or pull the sequential slices by identifier at a fast rate over the network.” Ex. 1002 ¶ 56. Carmel’s Figure 2 shows the client computers using HTTP to download from the network server. Ex. 1004, Fig. 2; *see* Ex. 1002 ¶ 53.¹⁸

Carmel’s Figure 6A depicts steps in a method that a client computer performs when downloading Figure 3A’s data stream from the network server, including the following steps: “HTTP from Server,” “Read Index File,” and “Select Slice.” Ex. 1004, 6:3–6, 7:18–22, 10:24–27, Figs. 3A, 6A; *see* Ex. 1002 ¶ 128. In that downloading method, a client connects to the server “optionally using multiple HTTP links” and typically “opens one or two HTTP links.” Ex. 1004, 10:35–38; *see* Ex. 1002 ¶ 94.

¹⁸ This citation references paragraph 53 on bottom-right pages 34 and 35 in Exhibit 1002.

Referring to Figure 6A, Carmel explains that the “client first reads index file 50 (FIG. 3B)” that the transmitting computer created and uploaded to the server. Ex. 1004, 10:41–42; *see id.* at 7:59–62, 8:1–5; Ex. 1002 ¶¶ 56, 80, 95, 124–125. Carmel’s Figure 3B (reproduced below) depicts an index file associated with Figure 3A’s data stream 40:

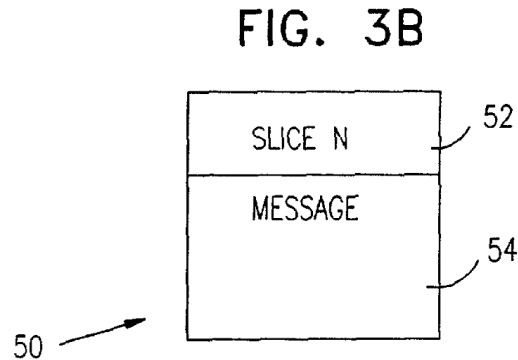


Figure 3B illustrates index file 50 including slice ID 52 and message 54. Ex. 1004, 5:49–52, 7:59–64, 8:12–13, Fig. 3B. “Each time a new file is uploaded to the server, index file 50 (FIG. 3B) is updated” by the transmitting computer. *Id.* at 10:3–5, Fig. 5 (step 86).

In index file 50, slice ID 52 indicates the index of the most recent slice in data stream 40 uploaded to the server by the transmitting computer and preferably “holds the file name of the new file, wherein the name typically comprises a string followed by the index of the file.” Ex. 1004, 7:62–8:1; *see* Ex. 1002 ¶ 80. In index file 50, message 54 “contains parameters relating generally to the data stream and/or instructions to” a client. Ex. 1004, 8:15–17.

When a client reads index file 50, “graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading.” Ex. 1004, 10:41–44; *see id.* at 8:7–9; Ex. 1002 ¶ 95. Then, “[r]esponsive to a user input,” the client “selects an

appropriate starting slice and begins to download and decode (decompress) files” containing data slices. Ex. 1004, 10:44–47; *see id.* at 8:7–9; Ex. 1002 ¶¶ 95, 98, 124. The client “request[s] or pull[s] the sequential slices by identifier.” Ex. 1002 ¶ 56; *see id.* ¶ 94.

Because Carmel discloses that (1) the “Read Index File” step follows the “HTTP from Server” step, (2) the “Read Index File” step involves a client reading index file 50, and (3) index file 50 resides on the server, Carmel indicates that the “HTTP from Server” step involves a client receiving index file 50 from the server. Ex. 1004, 10:24–27, 10:34–44, Fig. 6A; *see* Ex. 1002 ¶¶ 95, 98, 124–125. Carmel does not indicate that index file 50 contains any data slices. *See, e.g.*, Ex. 1004, 4:3–11, 5:49–52, 7:59–8:9, 8:12–18, 8:32–34, 10:3–5, 10:41–44, Fig. 3B. Hence, we disagree with Patent Owner that the “HTTP from Server” step refers to “receiving slices pushed from the server.” *See* Sur-reply 21; *see also* Resp. 39–40.

Patent Owner asserts that Carmel seeks to “send content over relatively slow connections.” Resp. 50. But Patent Owner does not explain why the “HTTP from Server” step would consume bandwidth by sending data slices to a client that the client may not use for playback. For example, a user may decide to begin playback at the most recent slice in data stream 40 uploaded to the server. *See* Ex. 1004, 8:1–7, 10:41–44. If so, the client would not use any earlier slices in data stream 40 that the client received in the “HTTP from Server” step according to Patent Owner’s position, thus wasting bandwidth. *See* Resp. 39–40; Sur-reply 21.

Further, referring to Figure 6A, Carmel explains that a client connects to the server, typically “opens one or two HTTP links,” and downloads files containing data slices “in successive alternation.” Ex. 1004, 10:35–39; *see*

Ex. 1002 ¶¶ 94, 124, 130. An ordinarily skilled artisan would have understood that downloading by alternating links requires a separate request for each link that causes the server to return a single file on each link. *See* Ex. 2005, 77:3–78:10, 79:9–20, 80:3–25, 82:6–83:12, 85:15–86:1, 90:2–91:1, 96:15–97:16.

Also, Figure 6A illustrates a loop including, among other things, the “Link Function OK?” determination, the “Continue” step, the “HTTP from Server” step, the “Read Index File” step, and the “Select Slice” step. Ex. 1004, Fig. 6A; *see* Ex. 1002 ¶ 128; Ex. 2005, 60:7–61:1, 74:2–22. When a client repeatedly performs the steps in that loop, the client repeatedly selects “an appropriate starting slice” (singular not plural) in the “Select Slice” step and causes the server to send all of the selected slices in response to requests from the client. Ex. 1004, 10:44–47, Fig. 6A; *see* Ex. 1002 ¶¶ 56, 95, 98, 128; Ex. 2005, 93:23–94:22. If the server sends all of the selected slices but also sends other slices, that does not negate Figure 6A’s relevance to limitation 1.k. As discussed above, claim 1 uses the transitional term “comprising” and, therefore, “does not exclude unrecited elements.” *See* Ex. 1001, 16:36–39; *Regeneron Pharm.*, 864 F.3d at 1352; *supra* § III.D.2(k)(v).a.

For these reasons, Carmel teaches limitation 1.k.

ii. Inherent Teaching or What an Ordinarily Skilled Artisan Would Have Understood Based on Carmel's Disclosures

Carmel's Figure 2 shows the client computers using HTTP to download from the network server. Ex. 1004, Fig. 2; *see* Ex. 1002 ¶ 53.¹⁹ Dr. Houh testifies that "it is inherent that when using the HTTP protocol to download each of the slice files, every slice file is requested individually using an HTTP GET request and the server sends each requested slice to the user system in response." Ex. 1002 ¶ 130; Ex. 1030 ¶ 29. We credit Dr. Houh's testimony because, as explained in more detail below, his testimony comports with the following:

- (1) RFC 1945 (Exhibit 2002) published in 1996 that specifies standards for HTTP 1.0;
- (2) RFC 2068 (Exhibit 2004) published in 1997 that specifies standards for HTTP 1.1; and
- (3) Carmel's disclosures.

HTTP 1.0 and HTTP 1.1 are based on a request-response framework where a client initially sends a request to a server and the server sends a response to the client. Ex. 2002 § 1.3; Ex. 2004 § 1.4; *see* Ex. 2002 § 1.2 (defining "client," "server," "request," and "response"); Ex. 2004 § 1.3 (defining "client," "server," "request," and "response"). "HTTP messages consist of requests from client to server and responses from server to client." Ex. 2002 § 4.1; Ex. 2004 § 4.1. A request or response may "transfer an entity," such as a file. Ex. 2002 §§ 1.2, 7; Ex. 2004 §§ 1.3, 7; *see* Ex. 2002 § 1.2 (defining "entity"); Ex. 2004 § 1.3 (defining "entity"). HTTP 1.1

¹⁹ This citation references paragraph 53 on bottom-right pages 34 and 35 in Exhibit 1002.

describes the information transferred as “the payload of a request or response” and “the payload of the message.” Ex. 2004 §§ 1.3, 4.1.

In both HTTP 1.0 and HTTP 1.1, a GET request “means retrieve whatever information (in the form of an entity) is identified by the Request-URI.” Ex. 2002 § 8.1; Ex. 2004 § 9.3; Ex. 1030 ¶ 32. The “Request-URI is a Uniform Resource Identifier” and “identifies the resource upon which to apply the request,” such as a file. Ex. 2002 § 5.1.2; Ex. 2004 § 5.1.2; *see* Ex. 1029, 58:8–16; Ex. 2005, 188:25–189:6. “As far as HTTP is concerned, Uniform Resource Identifiers are simply formatted strings” that identify a resource, e.g., by “name, location, or any other characteristic.” Ex. 2002 § 3.2; Ex. 2004 § 3.2.

If the Request-URI in a GET request refers to a file, the server sends only the specified file. Ex. 1030 ¶¶ 32–33, 37; Ex. 2005, 61:18–62:12, 65:10–66:6, 69:10–18, 78:7–10, 79:9–12, 82:6–83:12; *see* Ex. 2002 § 8.1; Ex. 2004 § 9.3. If the Request-URI in a GET request refers to a “data-producing process,” the server sends “the produced data” as “the entity in the response and not the source text of the process, unless that text happens to be the output of the process.” Ex. 2002 § 8.1; Ex. 2004 § 9.3; *see* Sur-reply 23–24 (quoting Ex. 2004 § 9.3).

HTTP 1.1 provides for chunked encoding. *See* Ex. 2004 § 3.6; Ex. 2006 ¶ 79. Specifically, the “chunked encoding modifies the body of a message in order to transfer it as a series of chunks, each with its own size indicator, followed by an optional footer containing entity-header fields.” Ex. 2004 § 3.6; *see* Ex. 2006 ¶ 79.

HTTP 1.1 also provides for range retrieval. *See* Ex. 2004 §§ 14.36.1–14.36.2; Ex. 2006 ¶ 80. Specifically, “all HTTP entities are represented in

HTTP messages as sequences of bytes,” and a “byte range operation may specify a single range of bytes, or a set of ranges within a single entity,” such as a file. Ex. 2004 § 14.36.1; *see* Ex. 1030 ¶¶ 47–48; Ex. 2006 ¶ 80. When using range retrieval, a GET request “may request one or more sub-ranges of the entity, instead of the entire entity.” Ex. 2004 § 14.36.2; *see* Ex. 1030 ¶ 47; Ex. 2006 ¶ 80.

Carmel discloses that a data stream includes multiple data slices, “preferably time slices” or slices with “multiple resolution or quality levels.” Ex. 1004, 2:1–6, 3:5–6; *see id.* at 8:56–9:5, 11:53–64; Ex. 1002 ¶¶ 73, 81, 112, 130. Carmel discloses two alternative arrangements for the slices in a data stream as follows:

- (1) each slice is “contained in a separate, respective file” (the “separate files” arrangement); and
- (2) all slices are “contained in a single indexed file” (the “single file” arrangement).

Ex. 1004, 2:22–26.

Carmel’s Figure 3A depicts a preferred embodiment of a data stream including multiple data slices, i.e., data stream 40. Ex. 1004, 5:46–48, 7:18–22, Fig. 3A; *see* Ex. 1002 ¶¶ 55, 75, 85;²⁰ Ex. 2006 ¶¶ 54–55. Carmel indicates that data stream 40 preferably implements the “separate files” arrangement, i.e., by describing storage of “each slice as a corresponding file.” Ex. 1004, 7:22–28; *see* Ex. 1002 ¶ 140; Ex. 2006 ¶ 56. Carmel’s Figure 6A shows steps that a client performs “in downloading and playing

²⁰ This citation references paragraph 55 on bottom-right pages 35 and 36 in Exhibit 1002.

back” data stream 40 in Figure 3A. Ex. 1004, 6:3–6, 10:24–27, Figs. 3A, 6A; *see* Ex. 1002 ¶¶ 95, 128.

Carmel’s Figure 3D depicts another preferred embodiment of a data stream including multiple data slices, i.e., multilevel data stream 41. Ex. 1004, 5:58–60, 8:42–47, Fig. 3D; *see* Ex. 1002 ¶ 81; Ex. 2006 ¶ 57. Carmel indicates that multilevel data stream 41 preferably implements the “single file” arrangement, i.e., by explaining that Figure 3D “illustrates a file format.” Ex. 1004, 8:42–44, Fig. 3D; *see* Ex. 1002 ¶ 81. Carmel’s Figure 6B shows steps that a client performs “in downloading and playing back” multilevel data stream 41 in Figure 3D. Ex. 1004, 6:7–10, 10:64–67, Figs. 3D, 6B; *see id.* at 9:8–9; Ex. 1002 ¶¶ 82, 129.

To establish unpatentability, Petitioner relies on the 3A-6A “separate files” embodiment and alternatively on the 3D-6B “single file” embodiment. *See* Pet. 43–46; Reply 15–23; Ex. 1002 ¶¶ 55–56, 98, 127–130;²¹ Ex. 1030 ¶¶ 37, 40, 43–44.

Although Carmel discloses client computers using HTTP for downloading from the network server, Carmel does not mention an HTTP GET request. *See, e.g.*, Ex. 1004, 2:11–14, 3:63–66, 5:25–28, 7:5–9, 10:37–39, 11:1–2, 15:18–21, Fig. 2. But an ordinarily skilled artisan would have recognized a GET request as the primary method to fetch data in both HTTP 1.0 and HTTP 1.1. Ex. 2006 ¶ 81; *see* Ex. 1030 ¶ 31; Ex. 2005, 97:22–98:2, 98:24–99:5.

Further, for the 3A-6A “separate files” embodiment, an ordinarily skilled artisan would have understood that “every slice file is requested

²¹ This citation references paragraph 55 on bottom-right pages 35 and 36 in Exhibit 1002.

individually using an HTTP GET request” and that the server sends only the requested slice file in response to a GET request for that slice file. Ex. 1002 ¶¶ 128, 130; Ex. 1029, 66:17–67:1; Ex. 1030 ¶¶ 32–33, 37; Ex. 2005, 61:18–62:12, 65:10–66:6, 69:10–18, 78:7–10, 79:9–12, 82:6–83:12. As Dr. Houh explains, “the slice files are labeled 42, 44, 46” in Figure 3A, and the server “provides all such requested slices to the client and no other slices.” Ex. 2005, 61:18–62:3; *see id.* at 82:6–83:12; Ex. 1002 ¶¶ 85, 128. Patent Owner’s two examples based on HTTP 1.1 where Carmel’s network server 36 sends a plurality of slices to client 30 in response to a single GET request from client 30 do not undermine what an ordinarily skilled artisan would have understood about the 3A-6A “separate files” embodiment. *See* Resp. 47–48; Ex. 1030 ¶¶ 28–51.

For the first example involving chunked encoding, the “chunked encoding modifies the body of a message in order to transfer it as a series of chunks.” Ex. 2004 § 3.6; *see* Ex. 2006 ¶ 79. Thus, for the 3A-6A “separate files” embodiment, the server may respond to a GET request for a single slice file by dividing the requested slice file into a series of chunks to transfer to the client. *See* Ex. 1030 ¶ 36; Ex. 2004 § 3.6; Ex. 2006 ¶ 79. But the server does not send multiple slice files in response to a GET request for a single slice file. *See* Ex. 1030 ¶¶ 32–33, 36–37; Ex. 2004 § 3.6; Ex. 2005, 61:18–62:12, 65:10–66:6, 69:10–18, 78:7–10, 79:9–12, 82:6–83:12, 118:16–120:2.

For the first example involving chunked encoding, Patent Owner asserts that an ordinarily skilled artisan could have created a data-producing process to (1) function as a GET request’s target on Carmel’s network server 36 and (2) cause the server to “transmit a sequence of slices” in

response to the GET request. *See* Resp. 47–48; Ex. 2006 ¶¶ 86–90. That an ordinarily skilled artisan could have modified Carmel’s server to do something other than what Carmel discloses does not negate Carmel’s disclosures or alter what an ordinarily skilled artisan would have understood based on those disclosures.

For the second example involving range retrieval, a GET request “may request one or more sub-ranges of the entity, instead of the entire entity.” Ex. 2004 § 14.36.2; *see* Ex. 1030 ¶ 47; Ex. 2006 ¶ 80. Thus, for the 3A-6A “separate files” embodiment, the server may respond to a GET request for one or more sub-ranges of a single slice file by sending the requested one or more sub-ranges. *See* Ex. 1030 ¶¶ 47–48; Ex. 2004 § 14.36.2; Ex. 2006 ¶ 80. But the server does not send multiple slice files in response to a GET request for one or more sub-ranges of a single slice file. *See* Ex. 1030 ¶¶ 47–50; Ex. 2004 § 14.36.2.

For the 3D-6B “single file” embodiment, however, the server may respond to a GET request for one or more sub-ranges of the “single indexed file” by sending the requested one or more sub-ranges corresponding to one or more slices (not files). *See* Ex. 2004 § 14.36.2; Ex. 2006 ¶ 92; *see also* Ex. 1004, 2:23–27 (explaining that a streamed packet may include “a range of one or more indices”). But Petitioner does not rely on the 3D-6B “single file” embodiment when asserting that Carmel inherently teaches limitation 1.k. Reply 22; Ex. 1030 ¶¶ 43–44. Rather, Petitioner relies on the 3A-6A “separate files” embodiment. Reply 22; Ex. 1030 ¶¶ 43–44.

To summarize, for the 3A-6A “separate files” embodiment, an ordinarily skilled artisan would have understood that “every slice file is requested individually using an HTTP GET request” and that the server

sends only the requested slice file in response to a GET request for that slice file. Ex. 1002 ¶¶ 128, 130; Ex. 1029, 66:17–67:1; Ex. 1030 ¶¶ 32–33, 37; Ex. 2005, 61:18–62:12, 65:10–66:6, 69:10–18, 78:7–10, 79:9–12, 82:6–83:12. In view of Carmel’s disclosures and what an ordinarily skilled artisan would have understood based on those disclosures, Carmel teaches limitation 1.k. And Carmel teaches limitation 1.k even if, as Patent Owner asserts, limitation 1.k requires that “the server sends no elements other than those requested by serial ID,” i.e., “each and every element sent must be responsive to a request for that element by its serial ID.” *See* Resp. 20; Sur-reply 19.

d. Modifying Carmel to Request
Each Slice File Individually

Because we find that Carmel teaches limitation 1.k, we do not reach Petitioner’s alternative argument about modifying Carmel to request each slice file individually. *See* Pet. 46; *supra* § III.D.2(k)(v).c.

(l) Limitation 1.l

Claim 1 recites “all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein.” Ex. 1001, 17:11–15 (limitation 1.l).

Petitioner contends that Carmel teaches limitation 1.l because the “format” and “order” of the data slices uploaded to network server 36 by transmitting computer 34 “does not change when they are downloaded by” a client computer. Pet. 47–49. Petitioner also contends that network server 36 sends the data slices to a client computer in the same “format” as stored on the server because the server’s memory “is limited, and files 42, 44, 46, etc.,

will be stored on the server and erased therefrom in a ‘first-in-first-out’ sequence.” *Id.* at 46–47 (emphasis omitted) (quoting Ex. 1004, 7:55–58). Further, Petitioner asserts that “[f]or such a ‘first-in-first-out’ sequence, the order in which” the data slices “are sent from the server to” a client computer “remains the same as the order in which they were received from” transmitting computer 34. *Id.* at 48 (citing Ex. 1002 ¶ 136).

Patent Owner makes no arguments specific to limitation 1.1. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches limitation 1.1. *See* Pet. 46–50; Ex. 1002 ¶¶ 132–138.

(m) Conclusion About Obviousness for Claim 1

For the reasons discussed above, Carmel in light of an ordinarily skilled artisan’s knowledge teaches claim 1’s subject matter. *See supra* §§ III.D.2(a)–(l). Hence, Petitioner has shown by a preponderance of the evidence that claim 1 is unpatentable under § 103(a) as obvious over Carmel.

3. INDEPENDENT CLAIMS 5 AND 9

Independent claims 5 and 9 recite subject matter similar to claim 1. *Compare* Ex. 1001, 16:36–17:15, *with id.* at 17:22–18:5, *and id.* at 18:13–65; *see* Ex. 1002 ¶¶ 58–60. Primarily, the independent claims differ in their preamble language as follows:

- claim 1’s preamble recites “[a] method for distributing over the Internet, from a server system to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media”;

- claim 5’s preamble recites “[a] server system for distributing a pre-recorded audio or video program over the Internet to one or more user systems”; and
- claim 9’s preamble recites “[a] computer program product for distributing over the Internet from a server system comprising at least one computer to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media, the computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith.”

Id. at 16:36–39, 17:22–23, 18:13–19.

For the “server system” recited in claim 5’s preamble and the “server system comprising at least one computer” recited in claim 9’s preamble, Petitioner identifies Carmel’s network server 36 comprising “any suitable type of computer or computer system, for example, a Sun Microsystems UltraSPARC station or a Windows NT server, as are commonly used by Internet Service Providers (ISPs).” Pet. 15 (emphasis omitted) (quoting Ex. 1004, 6:40–43).

For the “computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith” recited in claim 9’s preamble, Petitioner identifies Carmel’s disclosure of “[a]n appendix [with] computer-readable files” that (1) “exemplify aspects of the operation of system 32 (FIG. 2) and of the file structures and methods described hereinabove” and (2) are “stored on disk in a common folder or directory.” Pet. 16–17 (alterations by Petitioner) (emphases omitted) (quoting Ex. 1004, 13:55–59, 14:8–9).

As discussed above, Patent Owner proposes that the Board construe the term “program” in the preambles of the independent claims to mean “an

entire program.” Resp. 10–11 (emphasis omitted); *see id.* at 14, 19; *supra* § III.C. As also discussed above, Patent Owner does not argue that Carmel fails to teach “an entire program.” *See, e.g.*, Resp. 21–51; Sur-reply 8–25; *supra* § III.D.2(a).

Generally, a preamble does not limit a claim. *Allen Eng’g*, 299 F.3d at 1346. We need not decide whether the preambles in claims 5 and 9 limit the claims because we agree with Petitioner that Carmel teaches the preambles. *See* Pet. 15–17; Ex. 1002 ¶¶ 63–64, 67–68.

For the reasons discussed above for claim 1, Carmel teaches distributing “an entire program” according to Patent Owner’s proposed construction of the term “program.” Ex. 1004, 6:57–59, 10:44–49; *see id.* at 1:24–28; Ex. 1002 ¶¶ 66–67, 98; Resp. 11; *supra* § III.D.2(a). For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Carmel teaches the other aspects of claim 5’s preamble and claim 9’s preamble. *See* Pet. 15–17; Ex. 1002 ¶¶ 64–68; Ex. 1004, 6:40–43, 13:55–59, 14:8–9.

Petitioner contends that Carmel teaches claim 5’s subject matter and claim 9’s subject matter for the reasons Petitioner contends that Carmel teaches claim 1’s subject matter. *See* Pet. 18–50; Reply 7–24.

Patent Owner disputes that Carmel teaches claim 5’s subject matter and claim 9’s subject matter for the reasons Patent Owner disputes that Carmel teaches claim 1’s subject matter. *See* Resp. 21–51; Sur-reply 8–25.

For the reasons discussed above, we consider Patent Owner’s arguments for claim 1 unavailing. *See supra* §§ III.D.2(h)(iii), III.D.2(j)(iii), III.D.2(k)(v). For the reasons discussed above, Carmel in light of an ordinarily skilled artisan’s knowledge teaches claim 1’s subject matter and

the similar subject matter in claims 5 and 9. *See supra* §§ III.D.2(a)–(l); *see also* Ex. 1002 ¶¶ 62–138; Ex. 1030 ¶¶ 27–51. Hence, Petitioner has shown by a preponderance of the evidence that claims 5 and 9 are unpatentable under § 103(a) as obvious over Carmel.

4. DEPENDENT CLAIMS 2–4, 6–8, AND 10–12

(a) Dependent Claims 2, 6, and 10

Claim 2 depends directly from claim 1 and specifies that “the serial identifiers are sequential.” Ex. 1001, 17:16–17. Claims 6 and 10 depend directly from claims 5 and 9, respectively, and recite subject matter similar to claim 2. *Id.* at 18:6–7, 18:66–67.

Petitioner contends that Carmel teaches the limitations in claims 2, 6, and 10 because Carmel discloses that a data stream from the network server to a client computer “is divided into a sequence of segments or slices of the data, preferably time slices,” where each slice “is preferably assigned a respective slice index.” Pet. 51 (citing Ex. 1004, 2:1–21, 8:21–29). As support, Petitioner quotes Carmel’s disclosure about an example data stream as follows:

Data stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T1, T2, T3, etc. . . . Computer 34 stores each slice as a corresponding file, having a running slice index 1, 2, 3 . . . N.

Id. (first ellipsis added by Petitioner and second ellipsis in original)

(emphasis omitted) (quoting Ex. 1004, 7:22–28).

As further support, Petitioner reproduces Carmel's Figure 3A as shown below (Pet. 52):

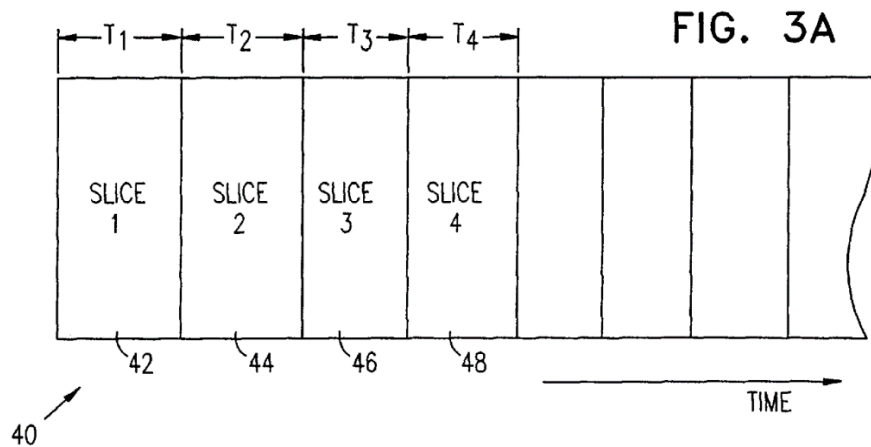


Figure 3A illustrates data stream 40 with broadcast data including “a series of data slices 42, 44, 46, 48, etc.” corresponding to successive time intervals labeled T_1 , T_2 , T_3 , T_4 , etc. Ex. 1004, 5:46–48, 7:18–25, Fig. 3A; *see* Ex. 1002 ¶ 141. Petitioner asserts that Figure 3A shows “sequential and numbered slices 1–4.” Pet. 51–52 (citing Ex. 1002 ¶ 141; Ex. 1004, Fig. 3A).

Patent Owner makes no arguments specific to claims 2, 6, and 10 and instead “disputes the Petition with regard to” these claims for the reasons Patent Owner disputes the obviousness of claims 1, 5, and 9. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh's testimony and for the reasons discussed above for claims 1, 5, and 9, we agree with Petitioner that Carmel teaches the subject matter in claims 2, 6, and 10. *See* Pet. 51–52; Ex. 1002 ¶¶ 139–141; *supra* §§ III.D.2(a)–(l), III.D.3.

(b) Dependent Claims 3, 4, 7, 8, 11, and 12

Claim 3 depends directly from claim 1 and specifies that “the sending is via a reliable transmission protocol.” Ex. 1001, 17:18–19. Claims 7 and 11 depend directly from claims 5 and 9, respectively, and recite subject matter similar to claim 3. *Id.* at 18:8–9, 19:1–2.

Claim 4 depends directly from claim 3 and specifies that “the reliable transmission protocol is TCP.” Ex. 1001, 17:20–21. Claims 8 and 12 depend directly from claims 5 and 11, respectively, and recite subject matter similar to claim 4. *Id.* at 18:10–11, 19:3–4.

Petitioner contends that Carmel teaches the limitations in claims 3, 4, 7, 8, 11, and 12 because (1) Carmel discloses that a client computer may download data slices from the network server using TCP and (2) TCP is a “reliable transmission protocol.” Pet. 52–54 (citing Ex. 1002 ¶¶ 142–146; Ex. 1004, 6:28–31, 6:50–54, 7:4–9).

Patent Owner makes no arguments specific to claims 3, 4, 7, 8, 11, and 12 and instead “disputes the Petition with regard to” these claims for the reasons Patent Owner disputes the obviousness of claims 1, 5, and 9. *See, e.g.*, Resp. 21–51; Sur-reply 8–25.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony and for the reasons discussed above for claims 1, 5, and 9, we agree with Petitioner that Carmel teaches the subject matter in claims 3, 4, 7, 8, 11, and 12. *See* Pet. 52–54; Ex. 1002 ¶¶ 142–146; *supra* §§ III.D.2(a)–(l), III.D.3.

(c) Conclusion About Obviousness for the Dependent Claims

For the reasons discussed above, Carmel in light of an ordinarily skilled artisan’s knowledge teaches the subject matter in claims 2–4, 6–8,

and 10–12. *See supra* §§ III.D.2(a)–(l), III.D.3, III.D.4(a)–(b). Hence, Petitioner has shown by a preponderance of the evidence that claims 2–4, 6–8, and 10–12 are unpatentable under § 103(a) as obvious over Carmel.

E. Alleged Obviousness over Carmel and Shteyn: Claims 1–12

Petitioner contends that claims 1–12 are unpatentable under § 103(a) as obvious over Carmel and Shteyn. *See* Pet. 6, 54–68. Patent Owner disputes Petitioner’s contentions, e.g., that Shteyn teaches limitation 1.j and the corresponding limitations in claims 5 and 9. *See, e.g.*, Resp. 21–22, 54–61; Sur-reply 25–26.

Above, we provided an overview of Carmel. *See supra* § III.D.1. Below, we provide an overview of Shteyn. Then, we consider the obviousness issues raised by the parties. For the reasons explained below, we agree with Petitioner that claims 1–12 are unpatentable under § 103(a) as obvious over Carmel and Shteyn.

1. OVERVIEW OF SHTEYN (EXHIBIT 1008)

Shteyn is a U.S. patent titled “Partitioning of MP3 Content File for Emulating Streaming,” filed on November 4, 1999, and issued on May 5, 2009. Ex. 1008, codes (12), (22), (45), (54). Shteyn states that the invention “relates to content and/or control communications between multiple computer systems, or to such communications between computer systems and consumer devices.” *Id.* at 1:6–9; *see id.* at code (57). More specifically, the invention “relates to communication[s] constrained by bandwidth or limited by data processing resources available to the receiving system or device, especially if the communications are received by the user in real time.” *Id.* at 1:9–13.

Shteyn describes a problem with current “technologies for delivering digital content, such as audio, video, etc.,” namely, “inherent latency” in the “complete download approach.” Ex. 1008, 1:18–19, 1:51–52. More specifically, “there is a delay between the beginning of the download and the start of the play-out.” *Id.* at 1:52–53. “The larger the file and or smaller the communication bandwidth, the longer it takes to transfer the content from the server to the client.” *Id.* at 1:53–55.

Shteyn addresses this problem by splitting a content file into multiple “parts” or “segments” at a server. Ex. 1008, 1:62–65, code (57). Shteyn refers to “parts” and “segments” interchangeably. *See, e.g., id.* at 1:65–2:38, 2:53–3:16, 3:31–36, 3:44–61, 4:20–31, code (57), Fig. 2.

“Each part or segment requires a relatively short download time.” Ex. 1008, 1:65–66. The “communications bandwidth” may determine the size of each part or segment. *Id.* at 2:1–3. The “play-out latency is determined by the download time” of the first part or segment because it “is played out upon downloading.” *Id.* at 1:67–2:1, code (57).

Shteyn discloses that a client downloads control information about a selected content file that “enables the retrieving and playing out” of the selected content file. Ex. 1008, 2:3–4, 2:57–59. According to Shteyn, the control information:

- (1) “comprises, for example, information relating to the size and memory location of the whole file as well as of it [sic] parts at the server”;
- (2) “contains references to the file location as well as references to the locations of the parts”;
- (3) “describes the locations, e.g., URL’s [sic], and size of the various file segments”;

- (4) preferably “is coded in XML [Extensible Markup Language]”; and
- (5) provides “UI [user interface] functionalities at the client.”

Id. at 2:4–7, 2:20–21, 2:60–63; *see id.* at 3:44–45, 3:53–55, 4:26–28, Fig. 2.

A “control information file” may contain the control information. *See* Ex. 1008, 5:45–67, 6:23–35, 6:41–62. Preferably, “the control information file is an XML file.” *Id.* at 6:16–17, 6:36–37, 6:63–64.

A content file’s parts or segments may “have different data formats” chosen “according to user-related information,” such as “quality of the equipment” and “bandwidth sharing/fluctuation conditions.” Ex. 1008, 2:25, 2:35–38; *see id.* at 4:20–23. A content file’s parts or segments may “physically reside on different servers.” *Id.* at 2:28–29.

Shteyn's Figure 1 (reproduced below) depicts steps in a method for emulating the streaming of files while using a download approach:

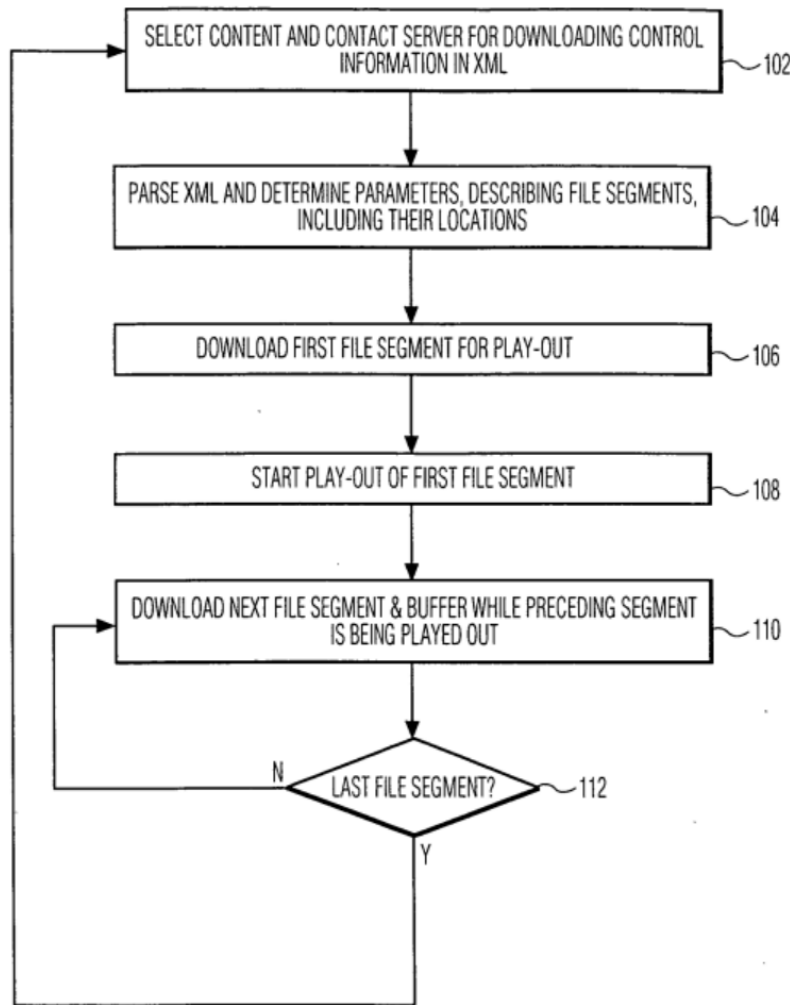


FIG. 1

Figure 1 illustrates “flow diagram 100 with various steps involved in the playing-out of a segmented file at the client,” i.e., steps 102 to 112. Ex. 1008, 2:45–46, 2:54–56, Fig. 1.

In step 102, the client contacts the server, “selects the particular content file,” and “downloads the control information that enables the retrieving and playing out of the segmented file.” Ex. 1008, 2:57–59, Fig. 1.

The control information “describes the locations, e.g., URL’s [sic], and size of the various file segments.” *Id.* at 2:60–61.

In step 104, the client parses the XML code, as “is well known in the art.” Ex. 1008, 2:64–65, Fig. 1. This enables the client to receive “information about the content information and the URLs of the first and subsequent file segments.” *Id.* at 2:67–3:2.

In step 106, the client downloads the first file segment for play-out. Ex. 1008, 3:3–4, Fig. 1. In step 108, the client starts play-out of the first file segment. *Id.* at 3:8, Fig. 1; *see id.* at 2:12–14.

In step 110, the client downloads the next file segment and stores it “in a buffer while the previous file segment, here the first file segment, is being played out.” Ex. 1008, 3:14–16, Fig. 1. “One option is to have the downloaded files buffered in a sequence or linked list of buffers.” *Id.* at 3:16–18. “This functionality is typically provided by the operating system of the client.” *Id.* at 3:18–19.

“Upon completion of playing out the first segment, the second segment is passed on from the buffer to the decoding/playing module.” Ex. 1008, 3:31–33. “This can be implemented by means of, e.g., a linked list.” *Id.* at 3:33–34.

In step 112, the client determines whether the last file segment has been played out. Ex. 1008, Fig. 1.

Shteyn explains that the control information may identify alternative versions of a content file’s parts or segments that have different formats and different locations. *See* Ex. 1008, 3:44–61, 4:20–31, Fig. 2. Shteyn’s Figure 2 (reproduced below) depicts control information identifying

alternative versions of a content file's first part or segment followed by multiple periods denoting subsequent parts or segments:

```
<XML>
  <title>
    The best ever music
  </title>
  <artist>
    V.R. Famous
  </artist>
  <parts>                                     (Preferred format)
    <part1>
      <length> 1024 </length>
      <format> MP3 </format>
      <location> ftp://137.27.52.87 </location>
      <min_bandwidth> 10,000 </min_bandwidth>
    </part1>

                                     (Alternative format)
    </part1_alt>
      <length> 512 </length>
      <format> OTHER </format>
      <location> http:// yevgeniy.net/ .... </location>
      <min_bandwidth> 8,000 </min_bandwidth>
    </part1_alt>

    .....
  </parts>
</XML>
```

FIG. 2

Figure 2 illustrates example control information coded in XML including information about “<title>,” “<artist>,” and “<parts>.” *Id.* at 2:47, 3:44–47, Fig. 2.

As Figure 2 shows, the example control information about “<parts>” includes the following:

- (1) for a segment labeled “part1” (and shown as “<part1>”) having a preferred format, “the length of the part, e.g., in bytes, the format, the minimum bandwidth required for a connection, and the location on the Internet”; and

- (2) for a segment labeled “part1_alt” (and shown as “<part1_alt>”) having a different format, the “different length, different format, different minimum bandwidth requirement,” and “different location” on the Internet.

Id. at 3:47–53, Fig. 2.

With the ability to download alternative versions of a content file’s parts or segments, a client may “select a next segment in a different format for the same content to adapt to changing circumstances, e.g., lower bandwidth due to network congestion.” Ex. 1008, 4:20–23; *see id.* at 2:21–22, 3:55–61.

Additionally, the client “controls the downloading and playing out” of the content segments by “pull[ing] the content segments from the locations indicated in the XML control information.” Ex. 1008, 4:26–31. When pulling the content segments, the client uses “the URLs of the first and subsequent file segments.” *Id.* at 2:67–3:2; *see id.* at 2:60–63.

2. DIFFERENCES BETWEEN THE CLAIMED INVENTION AND THE PRIOR ART

Petitioner contends that “[t]o the extent that Patent Owner argues that Carmel alone fails to disclose” limitations 1.j and 1.l and the similar limitations in claims 5 and 9, “Carmel in view of Shteyn disclose these limitations.” Pet. 54; *see id.* at 56–61, 63–66.

(a) Limitation 1.j and the Similar Limitations in Claims 5 and 9

Claim 1 recites “the one or more media data element[s] sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems.” Ex. 1001, 17:4–7 (limitation 1.j). Claims 5 and 9 recite similar limitations. *Id.* at 17:61–64, 18:54–57.

(i) Petitioner's Contentions

Petitioner contends that Shteyn teaches limitation 1.j and the similar limitations in claims 5 and 9 for two independent reasons. *See* Pet. 56–61. First, Petitioner asserts that Shteyn discloses that:

- (1) a client media player downloads an XML control file consisting of segment identifiers, e.g., “URL or filename,” and other control information about a content file available at a server;
- (2) the “control information describes the locations, e.g., URL’s [sic], and size of the various file segments, and provides, e.g., UI [user interface] functionalities at the client”;
- (3) the client “uses the control file to request in sequence each file by serial number,” e.g., by “URL or filename,” over the Internet; and
- (4) “the client pulls the content segments from the locations indicated in the XML control information for buffering and subsequent play-out.”

Id. at 56–59 (quoting Ex. 1008, 2:60–62, 4:28–31).

According to Petitioner, “the client media player selects segments by identifier without regard to the server and in fact, the client media player even buffers or records each segment by identifier,” including “the last segment received, in sequence until the end of the program.” Pet. 60 (citing Ex. 1002 ¶ 169).

Additionally, Petitioner asserts that Shteyn (1) “uses a URL as an identifier, including the filename and/or format, to select the proper file from the server” and (2) “requires a filename” to select the proper file from the server. Reply 25 (citing Ex. 1008, 3:46–61, Fig. 2). Petitioner also asserts that “select[ing] a next segment in a different format for the same content to

adapt to changing circumstances,” as Shteyn discloses, requires “some identifier” of the file to select the different format. *Id.* at 25–26 (quoting Ex. 1008, 4:20–22) (citing Ex. 1002 ¶¶ 149, 166–167; Ex. 1008, 3:55–61; Inst. Dec. 56).

Second, Petitioner asserts that Shteyn discloses that “the client media player maintains a linked list or data structure of buffers or segments” where “each element (here: segment) has content data and a pointer to a next element (here: next segment).” Pet. 60 (quoting Ex. 1008, 3:34–36). Petitioner further asserts that “the currently playing element in the linked list will maintain a record of the next or last segment received by identifier to be played.” *Id.* at 61 (citing Ex. 1002 ¶¶ 170–171).

(ii) Patent Owner’s Contentions

Patent Owner disputes that Shteyn teaches limitation 1.j and the similar limitations in claims 5 and 9. *See* Resp. 21–22, 51–52, 54–60; Sur-reply 25–26. Specifically, Patent Owner asserts that “Shteyn does not disclose a URL for each part, but instead only discloses a ‘locations’ where respective ‘parts’ (plural) may be found.” Resp. 51–52 (citing Ex. 2006 ¶¶ 65, 145, 149–152). Patent Owner also asserts that a client in Shteyn requests a “next” part after obtaining a “first” part, and the server responds by sending the “next” part. *Id.* at 52, 57–59 (citing Ex. 1008, 3:14–16, 4:20–22, 5:45–67, 6:41–62, code (57), Fig. 1; Ex. 2006 ¶¶ 66, 156–157, 161). According to Patent Owner, “[t]here is actually no teaching in Shteyn that reflects how the server determines the next element to send.” *Id.* at 59.

Regarding Shteyn’s disclosure that the client receives “information about the content information and the URLs of the first and subsequent file segments,” Patent Owner asserts that Petitioner “makes the leap that a

separate URL is provided for ‘each’ of the subsequent file segments, where Shteyn’s actual disclosure reflects no such thing.” Resp. 54–55 (quoting Ex. 1008, 3:1–2) (citing Pet. 57). According to Patent Owner, Shteyn reflects that location information “is for URLs (at different encoding levels)” with “each of the URLs being a URL from which the client may obtain the first and subsequent file segments for a given stream encoding.” *Id.* at 55; *see* Sur-reply 25.

Regarding Shteyn’s Figure 2, Patent Owner asserts that “part1” and “part1_alt” are “internal schema for the XML file” rather than “individual identifiers” used to request “a specific file segment on the server.” Resp. 57, 59 (citing Ex. 2006 ¶¶ 162–163). Patent Owner asserts that “part1” and “part1_alt” show only locations from which multiple parts “may be obtained.” *Id.* at 57; *see id.* at 56 (citing Ex. 1008, 2:3–7, 2:60–63, 2:67–3:2). Patent Owner also asserts that successive parts, such as “part2” and “part2_alt,” should “follow the same representation (data layout)” as “part1” and “part1_alt.” *Id.* at 57; *see id.* at 59.

Regarding Shteyn’s disclosure that the client uses a linked list to organize received segments, Patent Owner asserts that the linked list (1) “only comes into play after the segments have been requested and arrived” and (2) does not determine “what the server is asked to send.” Resp. 55 (citing Inst. Dec. 57–58).

(iii) Analysis

We agree with Petitioner that Shteyn teaches limitation 1.j and the similar limitations in claims 5 and 9. *See* Pet. 56–60; Ex. 1002 ¶¶ 149, 160–169, 175. Specifically, Shteyn discloses splitting a content file into multiple parts or segments at a server. Ex. 1008, 1:62–65, code (57); *see*

Ex. 1002 ¶¶ 149, 162; Ex. 2006 ¶ 63. “Each part or segment requires a relatively short download time.” Ex. 1008, 1:65–66.

Shteyn also discloses that a client downloads control information about a selected content file that “enables the retrieving and playing out” of the selected content file. Ex. 1008, 2:3–4, 2:57–59; *see* Ex. 1002 ¶¶ 162–163. Shteyn explains that the control information “contains references to the file location as well as references to the locations of the parts” and “describes the locations, e.g., URL’s [sic], and size of the various file segments.” Ex. 1008, 2:20–21, 2:60–61; *see id.* at 2:4–7; Ex. 1002 ¶¶ 162–163. Thus, the client receives “information about the content information and the URLs of the first and subsequent file segments.” Ex. 1008, 2:67–3:2; *see* Ex. 1002 ¶ 163. Hence, Shteyn teaches that distinct URLs may identify each part or segment in a content file, such as distinct URLs that identify “the first and subsequent file segments.” *See* Ex. 1002 ¶¶ 163–169, 175; Ex. 1008, 2:67–3:2.

Shteyn confirms that a URL may identify each part or segment in a content file by explaining that (1) the control information may identify alternative versions of the parts or segments that have different formats and different locations and (2) the parts or segments may “physically reside on different servers.” Ex. 1008, 2:28–29, 3:44–61, 4:20–31, Fig. 2; *see* Ex. 1002 ¶¶ 164–165. For instance, “part1” may reside on server #1, “part1_alt” may reside on server #2, “part2” may reside on server #3, and “part2_alt” may reside on server #4. *See* Ex. 1008, 2:28–29, 3:44–61, Fig. 2.

When a content file’s parts or segments “physically reside on different servers,” the client needs an individual identifier for each part or segment to

request a particular part or segment from the server it resides on. *See* Tr. 56:13–22. Different URLs may identify different locations for different parts or segments residing on different servers. *See* Ex. 1021, 2:20–22;²² Ex. 1029, 69:16–19; Ex. 2005, 130:3–9; Ex. 2007 § 2 (explaining that “URLs are used to ‘locate’ resources, by providing an abstract identification of the resource location”).

Consistent with this, the following excerpts from Shteyn’s Figure 2 for “part1” and “part1_alt” depict different URLs (locations) for “part1” and “part1_alt”:

<i>(Preferred format)</i>	<i>(Alternative format)</i>
<pre><part1> <length> 1024 </length> <format> MP3 </format> <location> ftp://137.27.52.87 </location> <min_bandwidth> 10,000 </min_bandwidth> </part1></pre>	<pre></part1_alt> <length> 512 </length> <format> OTHER </format> <location> http:// yevgeniynet/ </location> <min_bandwidth> 8,000 </min_bandwidth> </part_alt1></pre>

Above on the left, the excerpt from Figure 2 for “part1” shows the location of that segment as “ftp://137.27.52.87.” Ex. 1008, Fig. 2; *see* Ex. 1002 ¶ 165; Ex. 2006 ¶ 64. Above on the right, the excerpt from Figure 2 for “part1_alt” shows the location of that segment as “http://yevgeniynet/.” Ex. 1008, Fig. 2; *see* Ex. 1002 ¶ 165; Ex. 2006 ¶ 64.

Because Shteyn teaches that distinct URLs may identify each part or segment in a content file, e.g., when the content file’s parts or segments “physically reside on different servers,” we disagree with Patent Owner that Shteyn reflects that location information “is for URLs (at different encoding levels)” with “each of the URLs being a URL from which the client may

²² For Exhibit 1021 (Kliger), we cite to the page numbers of the publication rather than the page numbers that Petitioner applied to the exhibit.

obtain the first and subsequent file segments for a given stream encoding.”
See Ex. 1002 ¶¶ 163–169, 175; Ex. 1008, 2:28–29, 2:60–61, 2:67–3:2, 3:44–61, 4:20–31, Fig. 2; Resp. 55. For the same reason, we disagree with Patent Owner that “part1” and “part1_alt” show only locations from which multiple parts “may be obtained.” *See* Resp. 57.

Additionally, Shteyn discloses that the client “controls the downloading and playing out” of the content segments by “pull[ing] the content segments from the locations indicated in the XML control information.” Ex. 1008, 4:26–31; *see* Ex. 1002 ¶¶ 164–166. When pulling the content segments, the client uses “the URLs of the first and subsequent file segments.” Ex. 1008, 2:67–3:2; *see id.* at 2:60–63; Ex. 1002 ¶¶ 163–166, 175.

Consequently, Shteyn teaches that “the one or more media data element[s] sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems” according to limitation 1.j and the similar limitations in claims 5 and 9. *See* Ex. 1002 ¶¶ 149, 160–169, 175; Ex. 1008, 1:62–66, 2:3–7, 2:28–29, 2:57–63, 2:67–3:2, 3:44–61, 4:20–31, code (57), Fig. 2.

(b) Limitation 1.1 and the Similar Limitations in Claims 5 and 9

Claim 1 recites “all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein.” Ex. 1001, 17:11–15 (limitation 1.1). Claims 5 and 9 recite similar limitations. *Id.* at 18:1–5, 18:61–65.

Petitioner contends that Shteyn teaches limitation 1.1 and the similar limitations in claims 5 and 9 because Shteyn discloses:

- (1) “storing on a server computer a control information file of a format to be parsed by a client device”; and
- (2) “storing on one or more server computers multiple alternative files corresponding to a given segment of a media presentation accessible by the client device to, based on parsing of the control information file, determine which file of the multiple alternative files to retrieve based on system constraints to form a media presentation from the multiple alternative files.”

Pet. 63–64 (emphasis omitted) (quoting Ex. 1008, 6:26–34). According to Petitioner, “the segments once stored do not change format, nor order.” *Id.* at 64.

Petitioner also contends that Shteyn “necessarily” includes the claimed “data structure.” Pet. 64. Petitioner explains that “for Shteyn to divide the segments in sequence” and “then transmit based on requests from the clients for such numbered slices,” Shteyn “would store the slices in a data structure such as a linked list, array, or other organized data structure.” *Id.* (citing Ex. 1002 ¶ 181).

Patent Owner makes no arguments specific to limitation 1.1 and the similar limitations in claims 5 and 9. *See, e.g.*, Resp. 21–22, 51–61; Sur-reply 25–26.

For the reasons stated by Petitioner and supported by Dr. Houh’s testimony, we agree with Petitioner that Shteyn teaches limitation 1.1 and the similar limitations in claims 5 and 9. *See* Pet. 63–64; Ex. 1002 ¶¶ 179–181.

3. ALLEGED REASONS TO COMBINE THE TEACHINGS OF THE REFERENCES

Petitioner identifies reasons that would have prompted an ordinarily skilled artisan to combine Shteyn’s teachings with Carmel’s teachings in the way Petitioner proposes. *See* Pet. 61–64, 66–68; Reply 26–27. Further,

Petitioner asserts that an ordinarily skilled artisan “would have had a reasonable expectation of success” in combining Shteyn’s teachings with Carmel’s teachings. Pet. 62, 64. Dr. Houh’s testimony supports Petitioner’s positions. *See* Ex. 1002 ¶¶ 173–177, 182–189.

For instance, Dr. Houh testifies that “the combination is merely a matter of applying a known technique to a known device for improvement to yield predictable results.” Ex. 1002 ¶ 173; *see id.* ¶ 187. Dr. Houh also testifies that:

- (1) “Carmel discloses a client-pull streaming media system in which the client uses an index file listing the slices by ID to request slices by identifier or slice ID”;
- (2) “Shteyn similarly teaches a client-pull streaming media system in which the client uses a control file listing the segments each with an identifier to request segments one by one from the server based on, for instance, URL or filename”;
- (3) “Carmel also teaches that the client can download different quality level files from the index file depending on the network conditions”; and
- (4) “Shteyn also teaches that the client can download different bandwidth files from the control file depending on the network conditions.”

Id. ¶¶ 174–175.

Dr. Houh testifies that an ordinarily skilled artisan “would have been motivated to apply Shteyn’s teachings for a client-pull streaming media system to Carmel’s client-pull streaming media system to further enhance Carmel’s functionality.” Ex. 1002 ¶ 176. Dr. Houh explains that “Shteyn’s teaching of selecting a file based on system constraints” would “provide the benefit of ensuring continuous playback even when system constraints

caused network congestion that caused the available bandwidth to drop below the rate required by the currently [sic] quality level.” *Id.* ¶ 187.

Dr. Houh also testifies that an ordinarily skilled artisan “would have been motivated to combine Shteyn’s buffering or linked list of segments” with Carmel’s client to “further enhance Carmel’s client tracking of downloaded slices.” Ex. 1002 ¶ 176. Dr. Houh explains that “[s]uch additional buffering and linking of slices or sequences of files in Carmel” would “further enhance the buffer efficiency for uninterrupted playback.” *Id.*; *see id.* ¶ 187.

Patent Owner disputes that an ordinarily skilled artisan would have been motivated to combine Shteyn’s teachings with Carmel’s teachings. *See* Resp. 60–61; Sur-reply 25–26. Patent Owner disagrees that “Carmel discloses a client-pull streaming media system in which the client uses an index file listing the slices by ID to request slices by identifier or slice ID.” Resp. 60. Patent Owner asserts that Carmel teaches “no such thing” because Carmel discloses “seek then push” architecture. *Id.* Patent Owner also disagrees that “Shteyn similarly teaches a client-pull streaming media system in which the client uses a control file listing the segments each with an identifier to request segments one by one from the server based on, for instance, URL or filename.” *Id.* Patent Owner asserts that Shteyn teaches “no such thing” because Shteyn discloses “pull by next” architecture. *Id.* According to Patent Owner, the “alleged motivation to combine two nonexistent disclosures is thus flawed from the outset.” *Id.*

As for combining “Shteyn’s buffering or linked list of segments” with Carmel’s client, Patent Owner contends that Dr. Houh fails to explain why doing so “would help Carmel’s playback.” Resp. 60–61; *see* Sur-reply

25–26. According to Patent Owner, “Carmel’s client already has sequence numbers for each received slice” and “has no need to alter the way in which it makes requests.” Resp. 61 (citing Ex. 2006 ¶¶ 168–170); *see* Sur-reply 25–26 (citing Ex. 1004, 2:15–17). Patent Owner also contends that Dr. Houh fails to explain why combining “Shteyn’s buffering or linked list of segments” with Carmel’s client “would be even relevant” to limitation 1.j and the similar limitations in claims 5 and 9. Resp. 61.

Petitioner responds by asserting that Patent Owner mischaracterizes Carmel and Shteyn because “both references are streaming and client request systems.” Reply 26.

As for combining “Shteyn’s buffering or linked list of segments” with Carmel’s client, Petitioner contends that Patent Owner misunderstands the proposed combination. Reply 26. Petitioner explains that “Shteyn is not replacing Carmel’s system but enhancing it by tracking the slices selected and received” to “improve buffer efficiency and result in better overall playback.” *Id.* at 26–27.

We agree with Petitioner that an ordinarily skilled artisan would have had a reason, e.g., as articulated by Dr. Houh, to combine Shteyn’s teachings with Carmel’s teachings in the way Petitioner proposes. *See* Ex. 1002 ¶¶ 173–176, 187; Pet. 61–62, 66–67. As Dr. Houh testifies, “Shteyn’s teaching of selecting a file based on system constraints” would “provide the benefit of ensuring continuous playback even when system constraints caused network congestion that caused the available bandwidth to drop below the rate required by the currently [sic] quality level.” Ex. 1002 ¶ 187; *see id.* ¶ 176; Ex. 1008, 2:20–22, 2:35–38, 4:20–23.

As we understand Petitioner’s proposed combination, incorporating “Shteyn’s teaching of selecting a file based on system constraints” into a client in Carmel’s system would enable the client to “select a next segment in a different format for the same content to adapt to changing circumstances” based on a client-side identifier, e.g., a URL or filename for a data slice/segment. *See* Ex. 1002 ¶¶ 149, 162–163, 166–167, 169, 173–176, 187; Ex. 1008, 2:20–22, 2:35–38, 4:20–23; Pet. 61–62, 66–67; Inst. Dec. 62. The client’s ability to “select a next segment in a different format” for downloading based on a client-side identifier would have complemented the ability to employ different compression levels for downloading and helped to achieve uninterrupted playback. *See* Ex. 1002 ¶¶ 176, 187; Ex. 1004, 3:6–11, 4:43–47, 7:45–49, 9:6–8, 11:45–48; Ex. 1008, 2:20–22, 2:35–38, 4:20–23.

Additionally, the client’s ability to “select a next segment in a different format” for downloading based on a client-side identifier relates directly to limitation 1.j and the similar limitations in claims 5 and 9, i.e., that “the one or more media data element[s] sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems.” *See* Ex. 1001, 17:4–7, 17:61–64, 18:54–57.

Moreover, “[t]here is a motivation to combine when a known 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 ‘prior art elements according to their established functions.’” *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1380–81 (Fed. Cir. 2023) (citations omitted) (quoting *KSR*, 550 U.S. at 417; *Intel*

Corp. v. Qualcomm Inc., 21 F.4th 784, 799–800 (Fed. Cir. 2021)). Here, Petitioner proposes using “Shteyn’s buffering or linked list of segments” according to their established functions, i.e., “each element (here: segment) has content data and a pointer to a next element (here: next segment).” *See* Ex. 1002 ¶¶ 91–92, 150, 171–172, 176; Ex. 1008, 3:16–19, 3:31–36; Pet. 60–62. Using “Shteyn’s buffering or linked list of segments” in Carmel’s client would “further enhance the buffer efficiency for uninterrupted playback.” Ex. 1002 ¶ 176.

As for Patent Owner’s assertion that Carmel discloses “seek then push” architecture, we disagree for the reasons discussed above for limitation 1.k in the challenge based on Carmel alone. *See* Resp. 60; *supra* § III.D.2(k)(v).c. Specifically, for the 3A-6A “separate files” embodiment, an ordinarily skilled artisan would have understood that “every slice file is requested individually using an HTTP GET request” and that the server sends only the requested slice file in response to a GET request for that slice file. Ex. 1002 ¶¶ 128, 130; Ex. 1029, 66:17–67:1; Ex. 1030 ¶¶ 32–33, 37; Ex. 2005, 61:18–62:12, 65:10–66:6, 69:10–18, 78:7–10, 79:9–12, 82:6–83:12.

As for Patent Owner’s assertion that Shteyn discloses “pull by next” architecture, we disagree for the reasons discussed above for limitation 1.j in the challenge based on Carmel and Shteyn. *See* Resp. 60; *supra* § III.E.2(a)(iii).

As for combining “Shteyn’s buffering or linked list of segments” with Carmel’s client, the fact that “Carmel’s client already has sequence numbers for each received slice,” as Patent Owner asserts, does not undermine the proposed combination. *See* Resp. 61. “Shteyn’s buffering or linked list of

segments” would “further enhance the buffer efficiency for uninterrupted playback” without altering how the client uses time stamps (sequence numbers) to determine an ordering for the downloaded slices/segments. *See* Ex. 1002 ¶ 176; Ex. 1004, 2:15–17, 10:50–54; Ex. 1008, 3:14–19, 3:31–36. In particular, with the downloaded and ordered slices/segments “buffered in a sequence or linked list of buffers,” the next slice/segment “is passed on from the buffer to the decoding/playing module” upon “completion of playing out” the previous slice/segment using a pointer to the next slice/segment. Ex. 1008, 3:14–19, 3:31–36; *see* Ex. 1002 ¶ 171.

4. CONCLUSION ABOUT OBVIOUSNESS

For the reasons discussed above, the combined disclosures in Carmel and Shteyn teach the subject matter in independent claims 1, 5, and 9 as well as the subject matter in dependent claims 2–4, 6–8, and 10–12. *See supra* §§ III.D.2(a)–(i), III.D.2(k), III.D.3, III.D.4(a)–(b), III.E.2(a)–(b). Moreover, an ordinarily skilled artisan would have been motivated to combine Shteyn’s teachings with Carmel’s teachings in the way Petitioner proposes and would have had a reasonable expectation of success. *See supra* § III.E.3. Hence, Petitioner has shown by a preponderance of the evidence that claims 1–12 are unpatentable under § 103(a) as obvious over Carmel and Shteyn.

IV. CONCLUSION

Based on the evidence presented with the Petition, the evidence introduced during the trial, and the parties’ respective arguments, Petitioner has shown by a preponderance of the evidence that claims 1–12 are

unpatentable under § 103(a) as obvious over either (1) Carmel or (2) Carmel and Shteyn.²³

In summary:

Claims	35 U.S.C. §	Reference(s)/ Basis	Claims Shown Unpatentable	Claims Not Shown Unpatentable
1–12	103(a)	Carmel	1–12	
1–12	103(a)	Carmel, Shteyn	1–12	
Overall Outcome			1–12	

V. ORDER

Accordingly, it is

ORDERED that claims 1–12 in the '824 patent are determined to be unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision, the parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

²³ Should Patent Owner wish to pursue amendment of the challenged claims in a reexamination proceeding after the issuance of this Final Written Decision, we draw Patent Owner's attention to the April 2019 Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding. *See* 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. *See* 37 C.F.R. §§ 42.8(a)(3), (b)(2).

IPR2022-01228
Patent 9,742,824 B2

For PETITIONER:

Larissa S. Bifano
Thomas Fuller
DLA PIPER LLP (US)
larissa.bifano@dlapiper.com
thomas.fuller@dlapiper.com
DLA-Disney-IPR@us.dlapiper.com

For PATENT OWNER:

Ronald Abramson
M. Michael Lewis
Ari J. Jaffess
Gina K. Kim
LISTON ABRAMSON LLP
ron.abramson@listonabramson.com
michael.lewis@listonabramson.com
ari.jaffess@listonabramson.com
gina.kim@listonabramson.com