

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AKAMAI TECHNOLOGIES, INC.,

Petitioner,

v.

EQUIL IP HOLDINGS LLC,

Patent Owner.

Case IPR2023-00332

U.S. Patent No. 9,158,745

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

TABLE OF ABBREVIATIONS	iii
EXHIBIT LIST	iv
LIST OF CHALLENGED CLAIMS	vii
I. INTRODUCTION	1
II. MANDATORY NOTICES	3
A. Real Parties-in-Interest	3
B. Related Matters.....	3
C. Counsel and Service Information.....	3
III. PAYMENT OF FEES	4
IV. REQUIREMENTS FOR <i>INTER PARTES</i> REVIEW.....	5
A. Grounds for Standing	5
B. Identification of Challenge.....	5
1. Specific Art on Which the Challenge is Based.....	5
2. Statutory Grounds on Which the Challenge is Based.....	6
V. '745 PATENT AND PROSECUTION HISTORY	6
VI. §325(D) AND §314(A) DISCRETION DO NOT APPLY.....	8
A. §325(d).....	8
B. §314(a)	9
VII. LEVEL OF ORDINARY SKILL	10
VIII. CLAIM CONSTRUCTION	10
A. Preambles	11
B. “content generation operations”	11
C. “transformation operations”	11
IX. GROUNDS OF UNPATENTABILITY.....	12
A. Ground 1: Tso in view of Huang (Claims 1-5)	12
1. Tso.....	12

2.	Huang and Motivation to Modify Tso with Huang’s Teachings	16
3.	Claim Chart.....	21
B.	Ground 2: Tso in view of Huang and Lawler (Claims 6-7)	40
1.	Lawler	40
2.	Motivations to Modify Tso in view of Huang with Lawler’s Teachings.....	41
3.	Claim Chart.....	44
C.	Grounds 3/4: Samaniego (Claims 1-7).....	49
1.	Samaniego.....	49
2.	PO Intentionally Changed Inventorship	50
3.	Claim Chart.....	52
D.	Ground 5: Samaniego in View of Tso (Claims 2-4).	60
E.	Ground 6: Samaniego in View of Lawler (Claims 6-7).....	63
X.	SECONDARY CONSIDERATIONS	68
XI.	CONCLUSION.....	68

TABLE OF ABBREVIATIONS

'009 FH	File History of U.S. Application No. 09/929,904
'046 FH	File History of U.S. Application No. 12/173,747
'110 FH	File History of U.S. Application No. 12/238,842
'242	U.S. Patent No. 8,495,242
'575	U.S. Patent No. 6,792,575
'575 FH	File History of U.S. Application No. 09/425,326
'745	U.S. Patent No. 9,158,745
'745 FH	File History of U.S. Application No. 13/752,110
'916	U.S. Patent Application No. 11/269,916
'916 FH	File History of U.S. Application No. 11/269,916
Claims	Claims 1-7 of U.S. Patent No. 9,158,745
Davis	U.S. Patent No. 5,969,716
Delp	U.S. Patent No. 5,420,967
Huang	U.S. Patent No. 6,438,576
IPR	<i>Inter Partes</i> Review
Lawler	U.S. Patent No. 5,905,522
MTM	Motivation to Modify
Petitioner	Petitioner Akamai Technologies, Inc
PO	Patent Owner Equil IP Holdings LLC
POSITA	Person of Ordinary Skill in the Art
PTAB	Patent Trial and Appeal Board
RHI	Receiver Hint Information
Samaniego	U.S. Patent Application No. 2002/0078093
Starnes	U.S. Patent No. 6,144,996
Tso	U.S. Patent No. 6,421,733
USPTO	United States Patent and Trademark Office

EXHIBIT LIST

Ex. 1001	U.S. Patent No. 9,158,745 (“’745”)
Ex. 1002	File History of U.S. Application No. 13/752,110 (“’745 FH)
Ex. 1003	Declaration of Vijay K. Madiseti in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 9,158,745
Ex. 1004	U.S. Patent No. 6,421,733 (“Tso”)
Ex. 1005	U.S. Patent No. 6,438,576 (“Huang”)
Ex. 1006	U.S. Patent No. 5,905,522 (“Lawler”)
Ex. 1007	U.S. Patent Application Publication No. US 2002/0078093 (“Samaniego”)
Ex. 1008	U.S. Patent No. 5,420,967 (“Delp”)
Ex. 1009	U.S. Patent No. 5,969,716 (“Davis”)
Ex. 1010	File History of U.S. Application No. 11/269,916 (“’916 FH)
Ex. 1011	(Reserved)
Ex. 1012	U.S. Patent No. 6,792,575 (“’575”)
Ex. 1013	Redline Comparison of specifications from Samaniego and the ’745 patent
Ex. 1014	Loralee Stevens, Internet Video Startup Gets \$3.5 Million, N. Bay Bus. J. (Apr. 14, 2008), https://web.archive.org/web/20080421191837/www.northbaybusinessjournal.com/article/20080414/BUSINESSJOURNAL/19773530/1207/BUSINESSJOURNAL02
Ex. 1015	LinkedIn Profile for Chris Samaniego, https://www.linkedin.com/in/chris-samaniego-18b6a01

Ex. 1016	<i>Equil IP Holdings LLC v. Akamai Technologies, Inc.</i> , 1:22-cv-00677 (D. Del. 2022) First Amended complaint (Aug. 3, 2022)
Ex. 1017	(Reserved)
Ex. 1018	(Reserved)
Ex. 1019	(Reserved)
Ex. 1020	Docket Report for Berger-Great South Ventures LLC v. Equilibrium Techs.
Ex. 1021	Redline Comparison of specifications from Samaniego and Publication No. 2006-0265476 of Application No. 11/269,916
Ex. 1022	(Reserved)
Ex. 1023	(Reserved)
Ex. 1024	(Reserved)
Ex. 1025	(Reserved)
Ex. 1026	File History of U.S. Application No. 12/173,747 (“’046 FH”)
Ex. 1027	File History of U.S. Application No. 12/238,842 (“’110 FH”)
Ex. 1028	Redline Comparison of specifications from Samaniego and Publication No. 2009-0070485 of Application No. 12/173,747
Ex. 1029	Redline Comparison of specifications from Samaniego and Publication No. 2009-0089422 of Application No. 12/238,842
Ex. 1030	(Reserved)
Ex. 1031	(Reserved)
Ex. 1032	(Reserved)
Ex. 1033	Gerald Graef, <i>Graphics Formats for Linux</i> , 23 LINUX J., March 1996, https://dl.acm.org/doi/fullHtml/10.5555/325530.325533

Ex. 1034	File History of U.S. Application No. 09/425,326 (“’575 FH”)
Ex. 1035	File History of U.S. Application No. 09/929,904 (“’009 FH”)
Ex. 1036	Luca Delgrossi et al., <i>Media Scaling in a Multimedia Communication System</i> , 2 MULTIMEDIA SYS. 172 (1994), https://link.springer.com/article/10.1007/BF01210448
Ex. 1037	Armando Fox and Eric A. Brewer, <i>Reducing WWW latency and bandwidth requirements by real-time distillation</i> , COMPUT. NETWORKS & ISDN SYS., PROCS. 5TH INT’L WORLD WIDE WEB CONF. (May 1996), https://www.sciencedirect.com/science/article/abs/pii/016975529600027X
Ex. 1038	U.S. Patent No. 6,144,996 (“Starnes”)
Ex. 1039	Declaration of Jonathan Bradford

LIST OF CHALLENGED CLAIMS

[1.pre] A method in a host computer for developing transformation processing operations to optimize media content play-back to a plurality of playback devices connected with the host computer in a network, the method comprising:

[1.a] receiving a first request from a first playback device for media content;

[1.b] wherein the first request contains information, the information indicating a first original media content, first content generation operations, and first transformation operations;

[1.c] determining whether a previously-generated first intermediate media content is available for reuse, the previously-generated first intermediate media content having been created using the first original media content and the first set of content generation operations; and

[1.d] responsive to determining that a previously-generated first intermediate media content is available, creating a first optimized media content for the first playback device by performing the first set of transformation operations on the previously-generated first intermediate media content; and

[1.e] responsive to determining that a previously-generated first intermediate media content is not available, creating a first optimized media content for the first playback device by creating a first intermediate content using the first original media content and the first set of content generation operations,

and performing the first set of transformation operations on the first intermediate media content; and

[1.f] sending the first optimized media content to the first playback device

[2.pre] The method of claim 1, further comprising:

[2.a] receiving a second request from a second playback device for media content; wherein the second request contains information, the information indicating a second original media content, second set of content generation operations, and second set of transformation operations;

[2.b] wherein the second request is received substantially concurrently with receiving the first request;

[2.c] wherein the first original media content and the second original media content are the same media content;

[2.d] determining whether a second intermediate media content is available having been previously created using the second original media content and the second set of content generation operations; and

[2.e] responsive to determining that the second intermediate media content is available, creating a second optimized media content for the first playback device by performing the second set of transformation operations on the second intermediate media content; and

[2.f] responsive to determining that the second intermediate media content is

not available, creating a second optimized media content for the first playback device by creating the second intermediate content using the second original media content and the second set of content generation operations, and performing the second set of transformation operations on the second intermediate media content; and

[2.g] sending the second optimized media content to the second playback device.

[3] The method of claim 2, wherein the first set of content generation operations and the second set of content generation operations are the same, and the first set of transformation operations and the second set of transformation operations are different.

[4] The method of claim 2, wherein the first set of content generation operations and the second set of content generation operations are different, and the first set of transformation operations and the second set of transformation operations are different.

[5.pre] The method of claim 1, further comprising:

[5.a] determining whether a previously-generated first optimized media content is available for reuse, wherein the previously-generated first optimized media content was created using the first original media content, the first content generation operations, and the first transformation operations;

[5.b] responsive to determining that the previously-generated first optimized media content is available, sending the previously-generated first optimized media content to the first playback device.

[6.pre] The method of claim 1, further comprising:

[6.a] determining whether the host computer has sufficient processing resources to create the first optimized media content for the first playback device;

[6.b] responsive to determining that the host computer does not have sufficient processing resources to create the first optimized media content, determining an alternate first set of content generation operations or an alternate first set of transformation operations;

[6.c] creating an alternate first optimized media content using the alternate first set of content generation operations or the alternate first set of transformation operations.

[7.pre] The method of claim 6, wherein the first optimized media content is at a first level of quality and the alternate first optimized media content is at a second level of quality, wherein the first level of quality is higher than the second level of quality;

[7.a] wherein a level of quality of a media content is measured based on a compression format, a bit rate, and an image resolution of a media content sent to a playback device.

Pursuant to §§311-319 and §42.1,¹ Akamai Technologies, Inc. (“Petitioner”) petitions for *inter partes* review (“IPR”) of claims 1-7 (“Claims”) of U.S. Patent No. 9,158,745 (Ex. 1001, “’745”), assigned to Equil IP Holdings LLC (“PO”). Petitioner requests review of the Claims, and judgment finding them unpatentable under §§102 and 103.

I. INTRODUCTION

The ’745’s purported invention is a method of processing web media content via a cache for playback in a network. The Claims purport to provide a method for automated graphics delivery in which graphics are modified from an original file based on specified parameters. Specifically, the Claims recite receiving a content request including information specifying “content generation” and “transformation” operations. The system determines whether an “intermediate” version created using the “content generation” operations is already cached and, if so, generates and applies the “transformation” operations to that version or, if not, applies both operations to an original version of the content. Madisetti, ¶35.

¹ Section cites reference 35 U.S.C. (pre-AIA), 37 C.F.R., or internal sections as context indicates. Emphasis/annotations added unless noted. Figure annotations generally quote the Claims. All citations herein are exemplary and not meant to be limiting.

But these features were well-known in the art. For example, **Tso** (Ex. 1004) discloses a request packet carrying “selection criteria” specifying content properties—*e.g.*, data “type” and “encoding/compression”—and transformation operations—*e.g.*, “the degree of alteration” of content. Tso, 6:64-7:62. Tso’s server uses a “GetScaledObject()” call to find a cached “particular version” (*e.g.*, a processed version) of the content. Tso, 6:9-13. If the requested version exists, it is returned to the requestor. Tso, 4:1-5. Otherwise, the original content item is retrieved and processed. Tso, 6:9-50. **Huang** (Ex. 1005) further discloses retrieving partially-rendered intermediate content from cache, and completing rendering of that content. Huang, 7:23-8:11. A POSITA would have been motivated to modify Tso with Huang’s teachings, yielding a system that retrieves from cache and completes rendering of partially-rendered content. Madisetti, ¶¶36-37.

Separately, PO’s own art also discloses the Claims. The ’745 purports to claim, but is not entitled to, priority to Application 09/929,904, published as Samaniego (Ex. 1007) on 6/20/2002. Samaniego’s and the ’745’s specifications are substantively identical, but Samaniego shares no common inventors with the ’745, and is therefore §102(b) art. Madisetti, ¶38. At earliest, the ’745 can claim priority to Application 11/269,916 (*see* Ex. 1010), which shares common inventorship and was filed 11/7/2005. PO is estopped from disputing Samaniego discloses the Claims because the Examiner considered and agreed with PO’s argument that Samaniego

supports the Claims.

The Claims are rendered obvious by the combinations of **Tso, Huang, Lawler**, and **Samaniego** as set forth herein, and are also anticipated by or obvious in view of **Samaniego**. Madisetti, ¶39. Petitioner requests that the Board institute trial and find the Claims unpatentable.

II. MANDATORY NOTICES

A. Real Parties-in-Interest

Petitioner Akamai Technologies, Inc. is the real party-in-interest. No other party had access to or control over the present Petition, and no other party funded or participated in preparation of the present Petition.

B. Related Matters

Petitioner is concurrently filing IPR petitions challenging claims of related U.S. Patent Nos. 8,495,242 (“’242”) (IPR2023-00330) and 6,792,575 (“’575”) (IPR2023-00329). There are no other IPRs pending for the ’745 or patents related to the ’745 beyond those being concurrently filed.

The ’745 is the subject of the following co-pending civil action: *Equil IP Holdings LLC v. Akamai Technologies, Inc.*, No. 1-22-cv-00677 (D. Del.).

C. Counsel and Service Information

Scott McKeown (Reg. No. 42,866) (Lead)
ROPES & GRAY LLP
2099 Pennsylvania Avenue NW
Washington, DC 20006

Phone: 202-508-4740
Fax: 202-383-9335
scott.mckeown@ropesgray.com
akamai-equil-ropes-ipr-service@ropesgray.com

James L. Davis, Jr. (Reg. No. 57,325) (Back-up)
ROPES & GRAY LLP
1900 University Avenue, 6th Floor
East Palo Alto, CA 94303-2284
Phone: 650-617-4000
Fax: 617-235-9492
james.l.davis@ropesgray.com

Daniel W. Richards (Reg. No. 69,652) (Back-Up)
ROPES & GRAY LLP
1900 University Avenue, 6th Floor
East Palo Alto, CA 94303-2284
Phone: 650-617-4000
Fax: 617-235-9492
daniel.richards@ropesgray.com

Customer No. 28120

Mailing address for all PTAB correspondence:
ROPES & GRAY LLP, IPRM—Floor 43
Prudential Tower, 800 Boylston Street,
Boston, MA 02199-3600

Petitioner consents to electronic service of documents to the email addresses
of the counsel identified above.

III. PAYMENT OF FEES

The USPTO is authorized to charge any fees due during this proceeding to
Deposit Account No. 18-1945, under Order No. AKTI-011-653.

IV. REQUIREMENTS FOR *INTER PARTES* REVIEW

A. Grounds for Standing

§42.104(a): Petitioner certifies that the '745 is available for IPR. Petitioner and the real parties-in-interest are not barred or estopped from requesting IPR challenging the Claims on the grounds identified herein.

B. Identification of Challenge

§§42.104(b)-(b)(1): Petitioner requests IPR of claims 1-7, and that the Board cancel the same as unpatentable.

1. Specific Art on Which the Challenge is Based

Petitioner relies on the following prior art:

Name	US Patent/ Publication	Exhibit	Filed	Published/ Issued	Prior art under at least
Tso	6,421,733	1004	9/8/1997	7/16/2002	§§102(b)/(e)
Huang	6,438,576	1005	3/29/1999	8/20/2002	
Lawler	5,905,522	1006	8/31/1995	5/18/1999	§102(b)
Samaniego	2002/0078093	1007	8/14/2001	6/20/2002	

The '745's priority date is no earlier than 11/7/2005, when Application 11/269,916 was filed. The '745 cannot claim priority to the earlier-filed purported priority applications (09/425,326 and 09/929,904) because none of the '745's named inventors is "name[d] ... in the previously filed application[s]." 35 U.S.C. §120. A

valid priority claim requires the “involved application” and the “earliest application” to “share at least one common inventor.” *Polaris Wireless, Inc. v. TruePosition Inc.*, IPR2013-00323, Pap. 62, *25, *aff’d* 622 F. App’x 915 (Fed. Cir. 2015); *see also* Exs. 1002, 1034-1035. Moreover, the Examiner found that the Claims do not “receive the priority date of” 09/427,326 because that application does not provide written description support for the Claims. Ex. 1002, 165; *see In re NTP, Inc.*, 654 F.3d 1268, 1277 (Fed. Cir. 2011) (earlier application’s “written description” must “satisfy the requirements of §112”).

2. Statutory Grounds on Which the Challenge is Based

Ground	Claims	References
1 (§103)	1-5	Tso in view of Huang
2 (§103)	6-7	Tso in view of Huang, Lawler
3/4 (§§102/103)	1-7	Samaniego
5 (§103)	2-4	Samaniego in view of Tso
6 (§103)	6-7	Samaniego in view of Lawler

V. ’745 PATENT AND PROSECUTION HISTORY

Claims 1-7 recite methods for processing media using cached intermediate content items. ’745, 1:22-26, 17:39-41, 23:9-39. Figure 21 shows an example embodiment (Madisetti, ¶40).

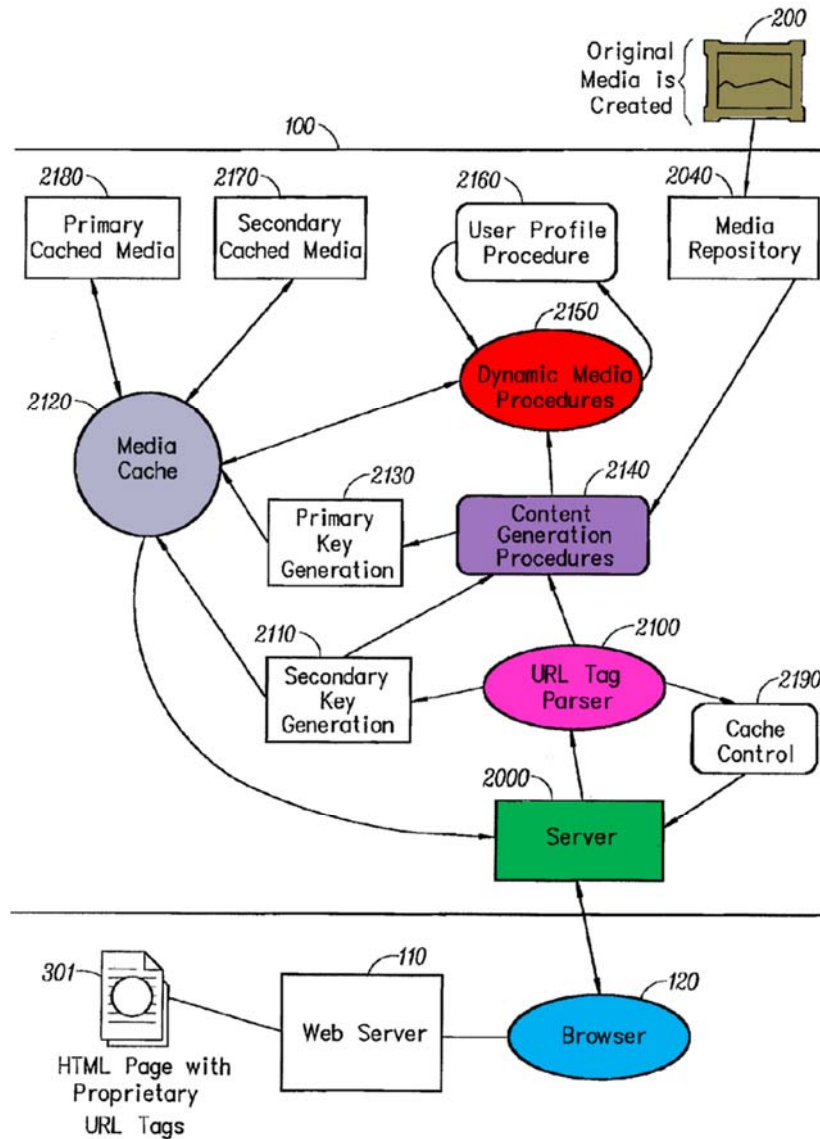


FIG. 21

A **server** receives a request from a user's **browser**. '745, 19:7-15. The server extracts from the request information specifying the requested **original media content**, and **content generation** and **transformation (e.g., "dynamic content processing")** operations to apply to that content. '745, 19:15-23. A **parser** determines whether a fully processed "primary" version of the requested media is

cached and if so, returns the cached content to the requester. '745, 19:24-27. Otherwise, the **parser** searches for previously-generated intermediate media in **cache**; if such media is found, **transformation operations** are applied to that media, and the resulting generated media is returned to the **browser**. 19:27-36, 20:1-2. If intermediate media was not cached, the requested media is generated using the **content generation** and **transformation operations** and then returned to the **browser**. *Id.* 20:1-2; Madisetti, ¶41.

During prosecution, the Claims were allowed, without rejection, in a first action. Ex. 1002, '745 FH, 156-57, 267-78; Madisetti, ¶¶42, 44. In a Notice of Allowance, the Examiner stated that “support for the claimed subject matter has been found in ... 09929904”—which published as **Samaniego**. Ex. 1002, 165; Ex. 1007; Madisetti, ¶43.

Notably, the Applicant filed a Request to Correct Inventorship during prosecution changing named inventors. Ex. 1002, 220-25. None of the originally-named or newly-identified inventors is named on **Samaniego** or the patent that issued therefrom. *See* Exs. 1001-1002, 1007, 1012, 1035.

VI. §325(D) AND §314(A) DISCRETION DO NOT APPLY

A. §325(d)

Under *Advanced Bionics v. Med-El Elektromedizinische Gerate*, IPR2019-01469, Pap. 6, *8-9, the Board should not exercise §325(d) discretion.

This Petition’s grounds are not the same or substantially the same as the prosecution art or arguments. None of the cited references was considered as art during prosecution. The Examiner did not issue any rejections before allowance. The Examiner did find that Samaniego fully supports the Claims, but did not consider that Samaniego is prior art. *See* Ex. 1002, 150-213. As explained below, the Claims are anticipated and/or obvious under the identified Grounds. *See* §IX.

Even if the art or arguments were substantially the same, the Examiner erred in a manner material to the patentability of the Claims. The Examiner “did not expressly consider” any cited references as prior art, rendering it difficult, if not impossible, to explain “how the Examiner might have considered the arguments presented in the Petition.” *Bowtech, Inc. v. MCP IP, LLC*, IPR2019-00379, Pap. 14, *20 (declining to exercise §325(d) discretion). Even if any art considered by the USPTO was substantially similar to these references (none is), the Examiner would have erred in not rejecting the Claims. It was also error not to consider Samaniego as prior art during prosecution in light of the inventorship difference.

B. §314(a)

The co-pending litigation also does not warrant the exercise of discretion under §314(a) under *Fintiv*.

1: Following institution, Petitioner intends to seek a stay of the Delaware case pending the outcome of this IPR and concurrently-filed IPR2023-00329 and

IPR2023-00330.

2: No trial date is set.

3: The case is in the pleading stage and the court has not ruled on the initial motion to dismiss.

4: After the final written decision, which is expected before trial even without a stay, the same grounds, arguments, and evidence could not be presented in the litigation.

5: The litigation and PTAB parties are the same.

6: The merits of this Petition are compelling as shown herein.

VII. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) as of the ’745’s claimed priority date would have had a bachelor’s degree in computer systems, computer science, or the equivalent thereof, and at least two years of experience with networked media delivery or related technologies. Madisetti, ¶¶45-46, 48-49. More education can supplement practical experience, and vice-versa. *Id.*

To the extent PO contends the ’745 is entitled to a priority date earlier than November 7, 2005 (it is not—see §IX.B.1), the Claims are still unpatentable. Madisetti, ¶¶47, 50-51.

VIII. CLAIM CONSTRUCTION

Claim terms subject to IPR are construed using *Phillips v. AWH Corp.*, 415

F.3d 1303, 1313 (Fed. Cir. 2005). §42.100(b). Only terms necessary to resolve the controversy need construction. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). Because the prior art asserted herein discloses embodiments within the Claims’ indisputable scope, the Board need not construe the Claims’ outer bounds. All claim terms should be construed according to their plain and ordinary meaning. Madisetti, ¶¶52-53.

A. Preambles

Regardless of whether the preambles are limiting, the prior art discloses them. See §IX; Madisetti, ¶54.

B. “content generation operations”

The specification provides examples of “content creation commands”: *e.g.*, converting media to a specified type/bit-depth, scaling to a specified size, and saving to a specified file. ’745, 21:16-22:39. A POSITA would have understood “content generation operations” to encompass the disclosed “content creation commands,” including the exemplary commands above, which are disclosed in the prior art (see §§IX.A, IX.C). Madisetti, ¶¶55-56.

C. “transformation operations”

The ’745 does not clearly define “transformation operations” and uses a form of “transform” only once outside of its claims, when discussing prior art. See ’745, 3:56. However, the specification discloses “proprietary tags” for implementing

“media processing script commands”: *e.g.*, “SetResolution” and “Colorize” operations. ’745, 10:27-15:20. A POSITA would have understood “transformation operations” to at least encompass the media processing script commands, including the exemplary commands above, which are disclosed in the prior art (see §§IX.A, IX.C). Madisetti, ¶¶57-59.

IX. GROUNDS OF UNPATENTABILITY

This Petition is supported by the Declaration of Dr. Vijay Madisetti, which describes the prior art’s scope and content at the time of the ’745. Madisetti, ¶¶1-34, 60-64. The art renders the Claims unpatentable under each of the Grounds.

A. Ground 1: Tso in view of Huang (Claims 1-5)

1. Tso

Tso discloses a networked “system for dynamically transcoding data.” Tso, 1:13-14; Madisetti, ¶65. Figure 3 shows an embodiment of the system:

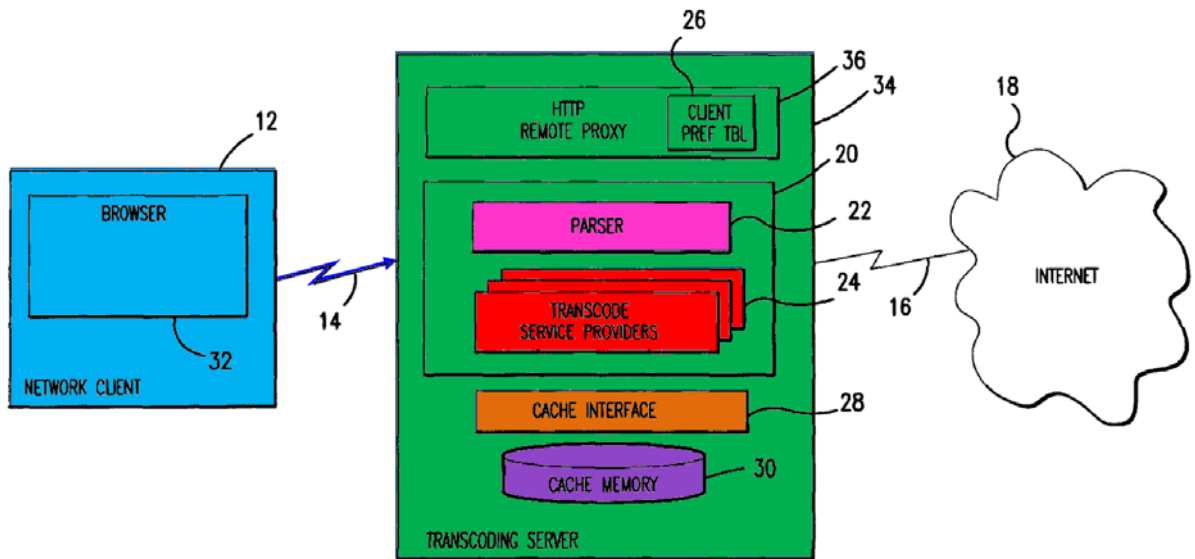


FIG.3

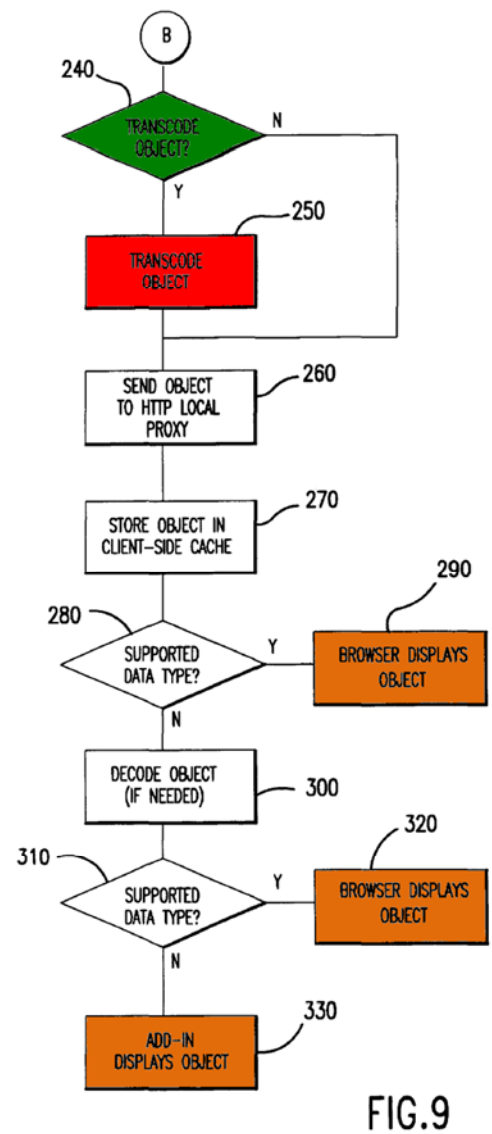
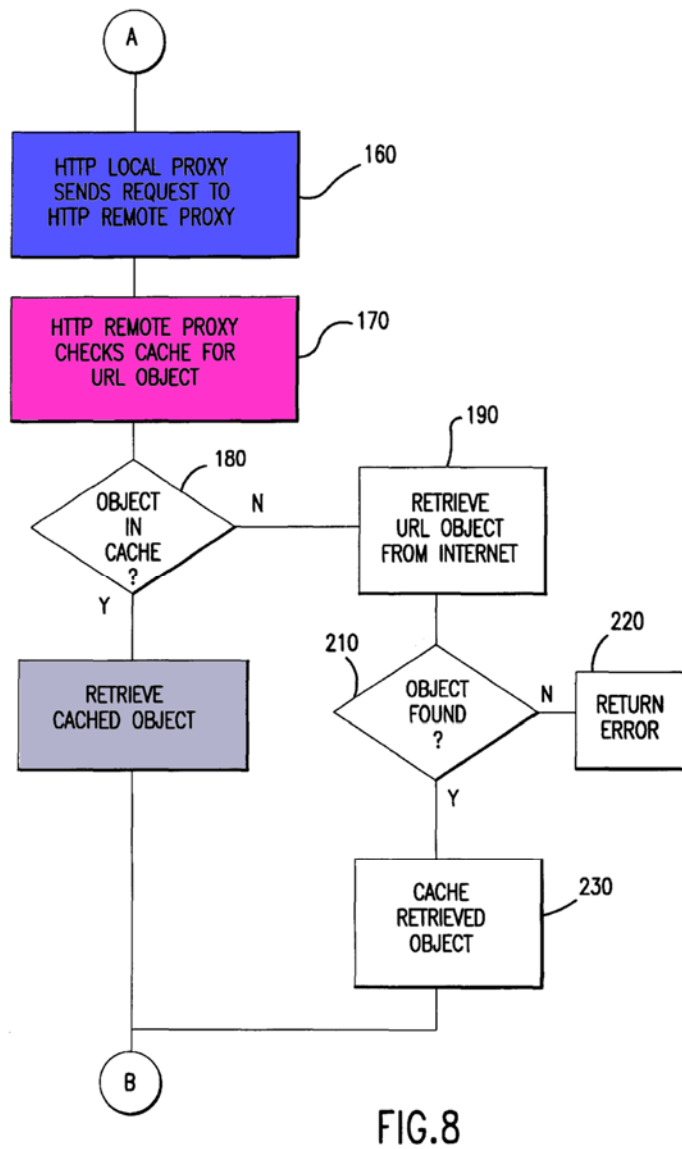
Tso, Fig. 3; Madisetti, ¶66. A **network client** includes a browser and sends a **request for “image, video, or HTML” content** to a **transcoding server**. Tso, 4:6-10.

The request packet stores “selection criteria.” Tso, 6:64-8:9. “Selection criteria” specify, *e.g.*, the “particular content” being requested (Tso, 7:4-7); “content characteristics, such as its data type, size, and dimension” (Tso, 7:20-33—an example of the claimed content generation operations (*see* §V))—and “the degree of alteration desired” for content and other “rules or programs to run for customizing content” including “resolution” (Tso, 7:20-29, 7:60-8:9—examples of the claimed transformation operations (*see* §V)). Madisetti, ¶¶66-67. Criteria also specify a “dynamically update[d]” “current processing load.” Tso, 6:64-7:14. The system applies these criteria and “combinations thereof” to select “transcode service providers” used to provide a “virtually limitless range” of “transcoding services.”

Tso, 3:48-65, 7:15-17, 8:10-14; Madisetti, ¶67.

The **transcoding server** receives the request and hands it to a transcoder, which includes a “**parser**,” “**transcode service providers**,” and a “**cache interface**” connected to “**cache memory**.” Tso, 3:8-17, 3:66-4:5, 4:14-19; Madisetti, ¶68.

Figs. 8-9 show relevant parts of Tso’s transcoding process:



Tso, Figs. 8-9; Madisetti, ¶69. First, the client **sends a content request**. Tso, 3:11-

16, 14:21-25. The server issues a “GetScaledObject()” call to **determine if the requested version of the content is cached**. Tso, 14:25-30. If so, the **content is retrieved** and **returned to the client**. Tso 14:30-32. Otherwise, **the parser uses “selection criteria” to determine whether additional transcoding is required**. Tso, 14:47-50, 7:15-8:9. **Any required transcoding is then performed** using transcode service providers. Tso, 14:50-55, 10:37-49. Different transcode service providers perform different transcoding operations, and additional providers are “dynamically added” to provide new transcoding “features” or “algorithms.” *Id.*; Tso, 4:23-37; Madisetti, ¶70.

Both “original” and “several different” “transcoded” versions of content are cached. Tso, 4:1-5, 6:13-23. Once the requested version of the content is generated and cached, the content is **returned to the client’s browser**. Tso, 6:42-50, 14:52-57; Madisetti, ¶71.

Tso also discloses using multiple-thread processing on servers to handle concurrent requests for content: the content is retrieved only once, and “appropriate versions” are delivered to multiple clients “concurrently.” Tso, 5:43-50, 6:51-63, 7:60-8:9; Madisetti, ¶72.

Tso also discloses using these teachings to “collaboratively” transcode content using multiple servers and clients. Tso, 14:47-15:6, 16:10-14. During collaborative transcoding, servers partially transcode content before sending it to clients or other

servers to complete transcoding; servers thus send, receive, and further transcode partially-transcoded content. *Id.*; Madisetti, ¶73.

2. Huang and Motivation to Modify Tso with Huang's Teachings

Huang teaches networked “object rendering” in which “rendering” includes, *e.g.*, transcoding that “renders” content “into different forms or resolutions” or from color to “black-and-white.” Huang, 1:17-22, 1:57-58, 6:59-63; Madisetti, ¶74.² Proxy servers cache partially-transcoded content, and retrieve such cached partially-transcoded content to complete rendering in response to client requests. Huang, 7:34-36, 8:4-11, 9:4-14; Madisetti, ¶74. Fig. 4 shows an embodiment of this process:

² As used herein, “rendering” refers to transcoding. These “resolution” and “bit depth” rendering operations describe similar operations to Tso’s “data type, type of encoding/compression,” “resolution,” and “number of colors” transcoding operations. *See* Tso, 7:15-8:9; Madisetti, ¶74.

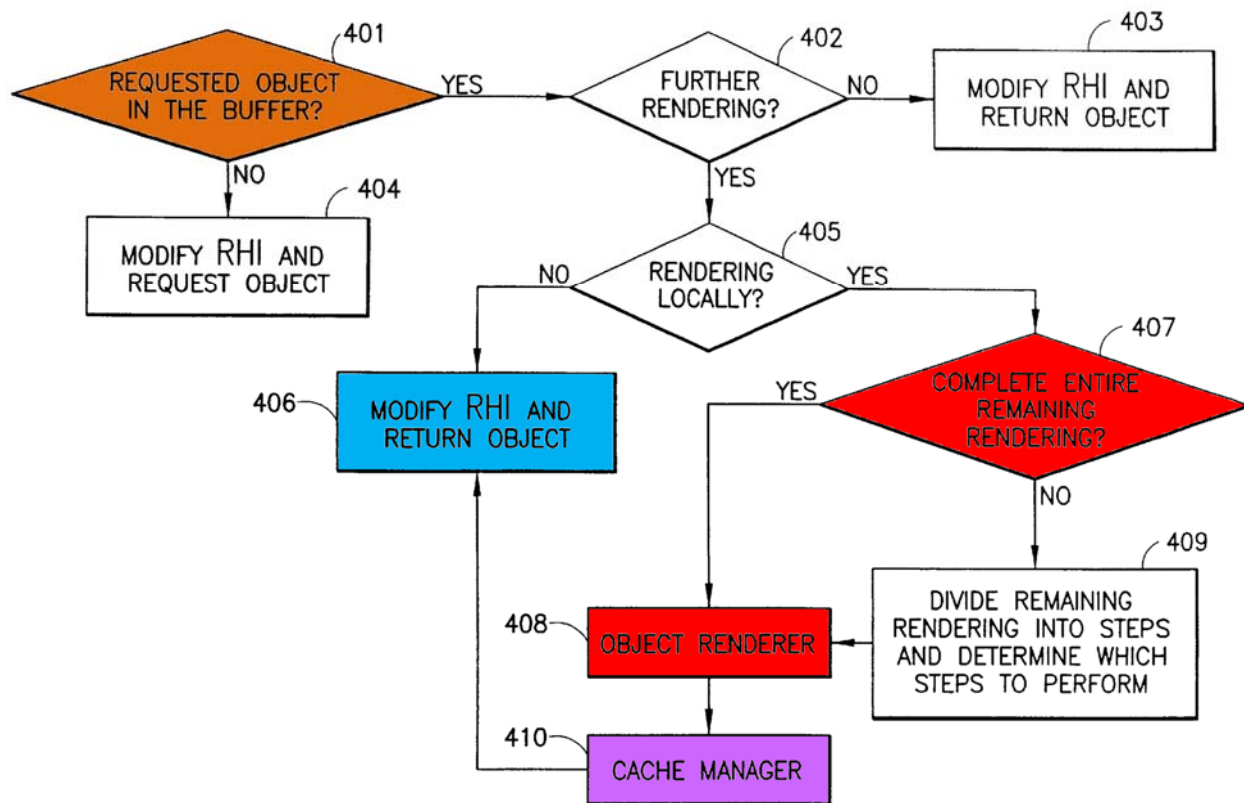


FIG.4

Huang, Fig. 4; Madisetti, ¶¶75-76. In response to a request, the proxy server **searches for a sufficiently “detailed” cached version** of the content for the request. Huang, 7:23-36. If one is found, the server **performs any additional needed rendering**, and then **caches** and **returns** the rendered content. Huang, 7:53-8:11; Madisetti, ¶76.

Content is stored along with Receiver Hint Information (“RHI”) and “object-specific descriptor” “meta-information.” Huang, 5:42-52. A server uses the RHI to determine whether “complete or partial rendering” is required, “partition” the rendering process into “two or more steps,” and, when performing rendering, modify

the RHI to “reflect the processing that [the server] performed.” Huang, 6:9-22, 6:56-68. When receiving a new request for previously-cached content, RHI is used to determine if “further rendering is necessary” and whether to “complete the entire rendering process” locally. Huang, 7:43-64. Rendering steps can occur in different orders: *e.g.*, “steps with the most bandwidth reduction” or steps to “reduce load unbalancing” can be performed first. Huang, 4:33-37; Madisetti, ¶77.

Tso and Huang are analogous prior art in the ’745’s same field and are reasonably pertinent to the ’745’s alleged problems of automating media delivery and improving overhead and speed for Web page generation. ’745, 1:23-26, 4:52-58, 5:1-10, 6:6-10; Tso, 1:10-31, 1:59-2:5, 4:1-5; Huang, 1:20-26, 2:1-10, 2:37-44; Madisetti, ¶78.

While **Tso** discloses servers sending, receiving, and further transcoding partially-transcoded content (*see* §IX.A.1), **Huang** teaches tracking, retrieving from local cache, and further rendering partially transcoded content, and organizing rendering into steps. *E.g.*, Tso, 5:39-41, 14:64-15:6; Huang, 6:63-67; Madisetti, ¶79. A POSITA would have been motivated to modify **Tso** with **Huang’s** partial rendering teachings for at least the following independent reasons. Madisetti, ¶79.

First, **Tso** and **Huang** disclose complementary mechanisms for improving processing efficiency. **Tso’s** teachings of storing “several different versions” of content to avoid “re-transcod[ing]” content, sending/receiving partially-transcoded

content, a client requesting cached content that requires “additional transcoding,” and using partially-transcoded content for, *e.g.*, “staged” or “collaborative” transcoding, provide **Tso** with partially-transcoded cached content. *See* Tso, 4:1-5, 4:39-61, 5:35-42, 6:13-17, 14:57-15:6, 15:66-16:14; *see also* §IX.A.1. For example, partially or fully-transcoded content may be further transcoded “to greyscale for users lacking a color display.” Tso, 17:9-15. **Tso** further teaches that such transcoding of already-cached content improves speed and efficiency by eliminating “the need to re-retrieve” or “re-transcode” “content.” Tso, 4:1-5. **Huang** explicitly teaches that retrieving partially-rendered content from local cache also advantageously allows a server to fully render content without unnecessary re-rendering and re-retrieval. Huang, 6:63-67, 7:42-8:11. **Huang** thus provides detailed teachings explaining how to further improve speed and efficiency by utilizing local previously-cached content, including “partially rendered” versions, to avoid “repeating” “rendering operations.” Huang, 6:63-67, 2:9-13. Thus, whether retrieving a fully-transcoded file from Tso’s cache as “partially rendered” content for further transcoding as suggested by Huang, or retrieving a partially transcoded version for further transcoding (*e.g.*, Tso, 14:57-15:6), both paths would improve Tso’s speed and efficiency. Madiseti, ¶80. A POSITA would have been motivated to improve **Tso** with **Huang’s** teachings to allow a server with its own locally-cached and partially-rendered/transcoded copy of content to also complete

rendering/transcoding on its own, which advantageously avoids unnecessary re-rendering/transcoding and re-retrieval of content. *Id.*; Huang, 7:23-50, 2:8-13; Tso, 4:1-5.

Second, Huang provides additional detailed teachings of tracking transcoding status of content, as referenced by **Tso**. **Tso** broadly teaches using “GetProperties()” and “SetProperties()” calls to track transcoding status, but leaves it to a POSITA to implement this function. *E.g.*, Tso, 5:36-42, 14:30-36. **Huang** provides additional details of how to track transcoding steps applied to content by managing available versions of content on a server. Huang, 5:60-6:23, 8:12-16. Applying these RHI teachings to **Tso** advantageously allows splitting transcoding/rendering into separate “steps,” caching “partially” transcoded/rendered versions, and tracking the transcoding/rendering status of each cached version in a manner that allows the server to configure rendering steps to, *e.g.*, maximize “bandwidth reduction” or reduce “load unbalancing” while avoiding “repeating” transforming/rendering steps on a server. Huang, 4:24-39, 6:9-23, 6:63-67. For example, it would have been obvious to adjust the “size” of content first to obtain “the most bandwidth reduction first.” Huang, 4:24-39. Madisetti, ¶¶81-82. *E.g.*, Ex. 1036 (Delgrossi), 173; Ex. 1037 (Fox), 1446; Ex. 1008 (Delp), 3:61-4:2.

In light of the above teachings, a POSITA would have had a reasonable expectation of success in modifying **Tso** with **Huang’s** teachings of reusing

partially-processed content. **Tso** already teaches caching retrieving specific versions of content, and both sending and receiving partially-transcoded content. Modifying **Tso** with **Huang’s** teachings would require using **Tso’s** already-taught functions, *e.g.*, “GetScaledObject()” and “Get/SetProperties(),” to additionally retrieve cached partially-transcoded content and track its properties as taught by Huang. Tso, 5:36-42; 12:53-13:11; Madisetti, ¶83. Indeed, such use of partially-transcoded content was well-known. *E.g.*, Ex. 1008, Abstract; Ex. 1009, 4:2-4. Modifying **Tso** with **Huang’s** teachings would have been a straightforward and predictable application of those teachings. Madisetti, ¶83.

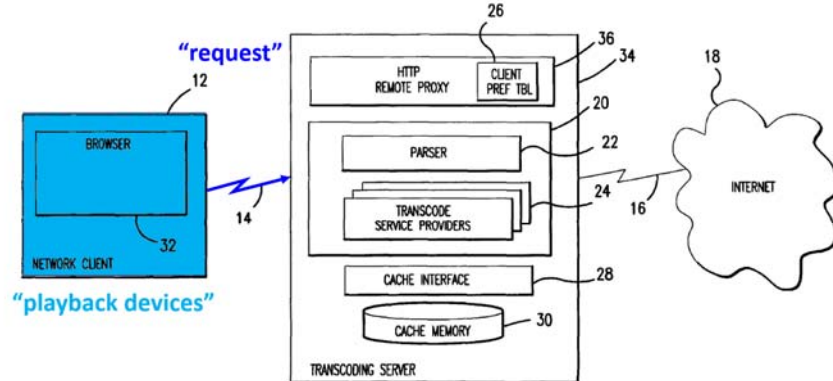
3. Claim Chart

’745	Tso in view of Huang
[1.pre] ³ A method in a host computer for developing transformation processing operations to optimize media content playback to a plurality of playback devices	Tso discloses a method in a host computer (<i>e.g.</i>, “server”) for developing transformation processing operations (<i>e.g.</i>, using “selection criterion” to “provide a virtually limitless range of dynamic transcoding services”) to optimize media content playback (<i>e.g.</i>, “compress and/or scale” media including “image, video, or HTML” to optimize or “reduce[] user-visible latencies”) to a plurality of playback devices (<i>e.g.</i>, “multiple users (i.e., clients),” “network client computer[s]”) connected with the host computer in a network (<i>e.g.</i>, “an ISP’s network,” “Internet 18”). <u><i>E.g., Tso:</i></u>

³ Petitioner charts preambles to the extent they are limiting without contending they are limiting for purposes of the Petition.

'745	Tso in view of Huang
connected with the host computer in a network, the method comprising:	Tso discloses that “a network server computer” develops a process for “transcoding” operations by using combinations of “selection criteria” to provide a “virtually limitless range of dynamic transcoding services” for “image, video, or HTML” content. Tso, 2:44-47, 3:45-60, 8:10-13, Fig. 3. This content is sent to “multiple users” across, <i>e.g.</i>, the Internet and is optimized to reduce “user-visible latencies.” Tso, 3:17-30, 16:63-17:15; Madisetti, ¶¶84-86. Fig 3 FIG.3 2:44-47 (“Embodiments of the present invention provide the ability to dynamically transcode information transmitted between, for example, a network server computer and a network client computer.”) 3:17-30 (“Transcoder 20 may be implemented, for example, as a software module installed in a network proxy, in a client device, in a network server device, or in a content server device. . . Remote transcoding server 34 may be coupled, for example, to an ISP's network, a corporate network, or anywhere on Internet 18, and may provide multiple users (i.e., clients) with a means to obtain content on Internet 18.”) 3:45-60 (“[T]ranscoder 20 is coupled to HTTP remote proxy 36. Parser 22 manages the transcoding of data to be transmitted from transcoding server 34 to network client 12. To this end, parser 22 controls transcode service

'745	Tso in view of Huang
	<i>providers 24 to selectively transcode content based on a predetermined selection criterion. For example, one or more transcode service providers 24 may provide the capability to compress and/or scale different types of data content, such as image, video, or HTML....")</i> • 8:10-13 (“<i>Applying the above-listed selection criteria, or combinations thereof, embodiments of the present invention may be used to provide a virtually limitless range of dynamic transcoding services.</i>”) • 16:63-17:15 (“The unique features of such embodiments <i>enhance the ability of these computers to access data</i> on the network in a timely fashion <i>with reduced user-visible latencies</i>”) • See also 2:8-12, 2:50-3:6, 6:55-63, 7:15-8:9, 15:7-28, 16:63-17:15, 22:13-18, Fig. 5.
[1.a] receiving a first request from a first playback device for media content;	Tso discloses receiving a first request (<i>e.g.</i>, “HTTP request”) from a first playback device (<i>e.g.</i>, “user,” “network client”; see [1.pre]) for media content (<i>e.g.</i>, “image” or “video” “data content”). <u>E.g., Tso:</u> See [1.pre]. Tso also discloses a “transcoding server” receiving a “HTTP request” for image or video data from a “network client.” Madisetti, ¶¶87-89.

'745	Tso in view of Huang
	Fig. 3  FIG.3 9:49-65 (“Network client 12, via browser 32, transmits an HTTP request for the hypertext object to transcoding server 34 over client/server communications link 14.”) - See also 3:7-16, 3:39-60, Fig. 5.
[1.b] wherein the first request contains information, the information indicating a first original media content, first content generation operations, and first transformation operations;	Tso discloses wherein the first request contains information (e.g., request “packet” contains “selection criteri[a]”), the information indicating a first original media content (e.g., indicating “particular content,” “requested hypertext object,” which can be saved as the “original version”), first content generation operations (e.g., indicating “content characteristics” specifying “data type, type of encoding/compression” or “size”), and first transformation operations (e.g., indicating “network client” properties, “content characteristics” specifying a “resolution,” or “content provider preferences” specifying how to “customize content”). <u>E.g., Tso:</u> See [1.pre]-[1.a]. Tso also discloses that request packets “selection criteri[a]” specifying the “particular content” requested and transcoding parameters used by a server’s “parser” to “selectively invoke” one or more “transcode service providers.” Tso, 6:64-7:14. The “selection criteria” specify content generation operations: e.g., “content characteristics” including “data type, type of

'745	Tso in view of Huang
	encoding/compression,” and “size.” Tso, 7:15-8:9.⁴ Madisetti, ¶¶90-92. The “selection criteria” also specify transformation operations: <i>e.g.</i>, “client” and additional “content characteristics” such as “number of colors,” “encoding/compression,” or “resolution,” and “content provider preferences” including “the degree of alteration desired for its content.” Tso, 7:20-8:4.⁵ Madisetti, ¶93. • 6:64-7:7 (“As noted above, <i>parser 22 may selectively invoke one of transcode service providers 24 based upon satisfaction of a predetermined selection criterion. Such selection criterion may comprise</i>, for example, <i>information contained in a header portion of a data packet [T]he predetermined selection criterion may comprise ... particular content</i>, key words, structures (for example, heading levels), and so on.”) • 7:15-8:9 (“The following discussion provides still more <i>examples of the types of information which may be used to dictate which of transcode service providers 24 are invoked</i>. It should be noted, however, that these examples are provided by way of illustration only, and are not

⁴ A POSITA would have understood these to disclose content generation operations consistent with, *e.g.*, the “save” and “scale” content generation operations in ’745 Table E, which save media to a “specified file” and scale media to a “specified size.” ’745, 21:20-22:38; Madisetti, ¶93.

⁵ A POSITA would have understood these to disclose transformation operations consistent with the ’745 “media processing script command” transformation operations including, *e.g.*, “Set Resolution” and “Colorize.” ’745, 10:50-15:20; Madisetti, ¶94.

'745	Tso in view of Huang
	intended to limit in any way the scope of the invention claimed herein. The predetermined <i>selection criterion</i> may comprise: (1) <i>network client 12, such as a display dimension, resolution, number of colors, ...</i> (3) <i>content characteristics, such as its data type, type of encoding/compression, size, and dimension; ...</i> (8) <i>content provider preferences, including the degree of alteration desired for its content, ... rules or programs to run for customizing content</i> (for example, news or advertising, custom language translation software) <i>based on user or client characteristics....</i>") • 14:37-46 ("If parser 22 determines that the requested <i>hypertext object does not exist</i> in the server-side cache memory 30, HTTP remote proxy 36 issues an HTTP request to retrieve the hypertext object from Internet 18 (Step 190). ... [I]f the object is found, HTTP remote proxy 36 passes the handle for the incoming data stream to parser 22, which in turn initiates <i>caching of an original version of the retrieved hypertext object</i> (Step 230).") • 17:9-15 ("Embodiments of the present invention may also be advantageously used for ... <i>converting color images to greyscale images for users lacking a color display....</i>") • See also 9:26-41, 9:49-65, 10:16-49.
[1.c] determining whether a previously-generated first intermediate media content is available for reuse, the previously-generated first	Tso discloses determining whether a previously-generated media content is available for reuse (<i>e.g.</i>, use "GetScaledObject()" to "request[] a particular version of that object, such as a high-quality rendition" from "cache" "determine whether a ... version of the requested hypertext object already exists" in "cache" and, if "not found," "create ... a placeholder"), and media content created using the first original media and the first set of content generation operations ("transcoded version[] of content" created using original media (see [1.b]) and content generation operations (see [1.b])).

'745	Tso in view of Huang
intermediate media content having been created using the first original media content and the first set of content generation operations; and	<u>E.g., Tso:</u> See [1.pre], [1.b]. Tso also discloses using a “GetScaledObject()” call to retrieve from cache a “particular version of [a media] object” that has been transcoded—<i>e.g.</i>, a version transcoded with the claimed content generation operations—to avoid the “need to...re-transcode the content.” Tso, 4:1-5, 6:9-28. If the system determines that the item is “not found” in cache, then the system creates “a placeholder” for the item to be cached. Tso, 6:9-28. Madisetti, ¶¶95-96, 98. Tso further discloses that each server creates and receives partially-transcoded content, and caches transcoded versions of content. Tso, 4:1-5, 5:36-41, 14:47-15:6, 15:66-16:14; §IX.A.1. As a result, a POSITA would have understood that a server responsible for performing only a partial transcoding caches that partially-transcoded version of content. <i>Id.</i>; Madisetti, ¶99. Tso is agnostic to the order in which transcoding steps are performed, and teaches adding new “features” using new “transcode service providers” without affecting existing providers. Tso, 4:23-29. Because providers can perform transcoding in any order, Tso not only renders obvious, but affirmatively discloses, first converting content to a particular “encoding/compression” or “size” (i.e., according to the claimed content generation operations) before performing additional transcoding operations. Madisetti, ¶100; see §IX.A.1.

'745	Tso in view of Huang
	• Figs. 8 and 9 “determine whether a previously-generated first intermediate content is available” FIG. 8 FIG. 9 • 4:1-5 (“<i>Server-side cache memory 30 may be used to store both original and transcoded versions of content for later transmission to network client 12 without the need to re-retrieve the content from Internet 18 or to re-transcode the content.</i>”) • 4:23-29 (“The WOSA architecture ... enables <i>additional transcode service providers 24 to be dynamically added to the system</i> to provide new features and/or better transcoding algorithms, while at the same time <i>not requiring changing or retesting other software components in the system.</i>”) • 5:36-41 (“The GetProperties() and SetProperties() calls retrieve and store <i>information about cached objects</i>, including information maintained by transcode service provider 24 <i>used to determine transcoding properties and transcoding status of a cached object</i>. Transcode service provider 24 may use such information, for example, <i>to determine current compression progress for scaled data access and staged refinements.</i>”) • 6:9-28 (“The <i>GetScaledObject()</i> call ... is also used to request an object from server-side cache memory 30; however, it <i>adds support for requesting a particular</i>

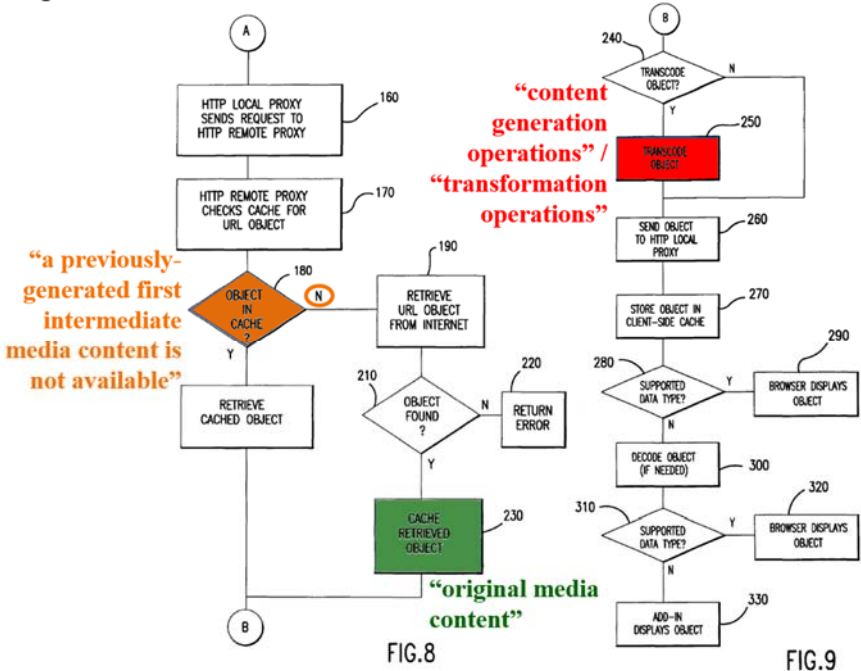
'745	Tso in view of Huang
	<i>version of that object, such as a high-quality rendition. ... [W]hen network client 12 requests a hypertext object, HTTP remote proxy 36 uses either the GetObject() or GetScaledObject() call (depending on if network client 12 is capable of receiving scaled/transcoded datatypes) to retrieve the hypertext object from parser 22. If the hypertext object is not found, parser 22 uses the CreateEntry() call to create an entry (in effect, a placeholder) in server-side cache memory 30 for the new object.”)</i> • 14:47-15:6 (“Upon receipt, HTTP local proxy 48 initiates caching of the transcoded hypertext object (Step 270). In addition, <i>client-side parser 50 determines whether any further processing is required</i> before the hypertext object is rendered <i>If additional transcoding is required, client-side parser 50 passes the handle to an appropriate transcode service provider 52</i> (Step 300).”) • 15:66-16:14 (“[A]n embodiment of the present invention may use <i>a special proxy-to-proxy protocol that extends the existing request/ response structures to indicate whether and what sort of transcoding has already been performed on the content</i>. Such a specialized protocol ... <i>enables multiple proxies to work collaboratively</i>, yet still transparently to users, client software, existing "standard" proxies and content servers.”) • See also 4:38-47, 4:62-5:7, 5:36-42, 13:39-44, 14:21-15:6. Huang discloses determining whether a previously-generated first intermediate media content is available for reuse (<i>e.g.</i>, determining that “a copy of the requested object can be found in the local cache” for which “further rendering is necessary”), the previously-generated first intermediate media content having been created using the first original media content and the first set of operations (<i>e.g.</i>, the copy found in local cache is “partially rendered” using “only one” of “two or more steps” tracked with “RHI” information).

'745	Tso in view of Huang
	<u>E.g., Huang:</u> In addition to Tso's teachings of servers sending, receiving, and caching partially transcoded content, Huang further teaches a server determines if a partially-rendered—<i>i.e.</i>, intermediate—version of content is cached and, if so, “further rendering” the “partially rendered” content on which, <i>e.g.</i>, “only one” of “two or more steps” was performed. Huang, 4:24-39, 6:9-23, 6:63-67, 7:42-8:11; §IX.A.2. Content rendering status is tracked with “RHI” information. Huang, 6:9-23, 6:56-58, 7:42-8:11; Madisetti, ¶¶97, 101. The server selects an order to perform rendering steps based on, <i>e.g.</i>, load balancing or bandwidth reduction. Huang, 4:24-39. As discussed in §IX.A.2, a POSITA would have been motivated to modify Tso with Huang's teachings to advantageously avoid re-retrieval and re-transcoding as previously performed on partially-transcoded content, and perform transcoding steps in an order that improves the efficiency of the system, advantageously streamlining content generation. Tso, 14:47-15:6, 15:66-16:14; Huang, 6:63-67, 7:42-8:11; Madisetti, ¶102. Applying Huang's teachings, a POSITA would have found it obvious for a server to use Tso's GetScaledObject() and GetProperties() calls to retrieve and check the transcoding status of a cached copy of content, including determining previously-performed transcoding/rendering steps, before further transcoding content using that partially-transcoded version instead of re-retrieving and re-transcoding original content. <i>Id.</i> When transcoding, a POSITA also would have found it obvious to apply Huang's teaching of performing “bandwidth reduction” transcoding steps by first performing Tso's “size”-based transcoding (a content generation operation) prior to performing other types of transcoding—improving “bandwidth” consumption of transcoded content. Huang, 4:24-39. Content transcoded for size is then stored as “partially-rendered” content in cache used to “avoid repeating some object rendering operations.” Huang, 6:63-67. As a result, subsequent searches for partially-transcoded content would locate partially-

'745	Tso in view of Huang
	transcoded cached versions of content on which content generation operations, but not transformation operations, had been performed, rendering obvious this element. <i>Id.</i>; Madisetti, ¶103. • 4:24-39 (“A selection method is provided for <i>each intermediate node to decide dynamically and independently of other nodes what portion of the required rendering to perform locally using the RHI information</i>. The selection method can include steps of (a) <i>dividing a remaining rendering operation into steps</i>; (b) <i>selecting one or more rendering steps to be performed locally</i> in order to optimize a given objective function, using the RHI information as an input parameter; and (c) <i>performing the one or more rendering steps selected for the current node</i>. The objective function can be to <i>perform the rendering steps with the most bandwidth reduction first, and/or to perform the rendering steps so as to reduce load unbalancing</i> among the remaining nodes on the path to the requesting client device node.”) • 6:9-23 (“When a requested object passes through the proxy network, <i>any proxy server 110, 111, 112 can perform a complete or partial rendering based on the associated RHI</i>. For example, <i>if the entire rendering process can be partitioned into two or more steps, a given one of the proxy servers (e.g., 110) may decide to perform only one of the rendering steps</i>, and to then forward the partially rendered object to another proxy server (e.g., 111). <i>The proxy server 110 also modifies the RHI to reflect the processing that it performed</i>, and forwards the modified RHI as well to the proxy server 111.”) • 6:56-58 (“An object renderer ... performs object rendering according to the <i>RHI associated with a particular object...</i>”)

'745	Tso in view of Huang
	• 6:63-67 (“Proxy server logic 203 may also include a <i>cache manager 207</i> which <i>maintains a local copy of the partially rendered or completely rendered objects in order to avoid repeating some object rendering operations with the same proxy server.</i>”) • 7:42-8:11 (“<i>If a copy of the requested object can be found in the local cache, at step 402, the proxy server checks the cached object against the RHI to see if any further rendering is necessary. ... By checking the RHI, the proxy server 110, 111, 112 can determine if any further rendering is necessary.</i> If no further rendering is necessary, the proxy server modifies the RHI to indicate its local condition for providing rendering services and returns the object at step 403. <i>If further rendering is found to be necessary</i>, based on the RHI or the requesting device, then <i>the proxy server</i> executes at step 405 a selection function to <i>determine whether</i> or not it wishes <i>to perform the rendering locally.</i> ... If the proxy server instead decides to perform the rendering locally, it checks the RHI at step 407 to <i>determine if it wishes to complete the entire rendering process</i>, or just some part of the required rendering process. ... <i>After</i> the rendering process is complete (<i>either a complete or partial rendering</i>), the cache manager 207 is called at step 410 to determine whether or not to <i>cache a copy of the rendered object locally.</i> <i>The proxy server 110, 111, 112 then modifies the RHI at step 406</i> to reflect its local condition and <i>returns the rendered (completely rendered or partially rendered) object along with the modified RHI.</i>”) • See also 3:32-47, 8:50-59.
[1.d] responsive to determining that a previously-generated first intermediate	Tso discloses performing the first set of transformation operations (see [1.b]). See [1.pre]-[1.c]. Huang discloses responsive to determining that a previously-generated first intermediate media content is available (e.g., “if a copy of the requested object can be found in the local

'745	Tso in view of Huang
media content is available, creating a first optimized media content for the first playback device by performing the first set of transformation operations on the previously-generated first intermediate media content; and	cache” for which “further rendering is necessary”), creating a first optimized media content for the first playback device (e.g., creating a “completely rendered ... object”) by performing the first set of operations on the previously-generated first intermediate media content (e.g., “proxy server” “decides to perform the rendering locally” and “complete the entire rendering process”). <u>E.g., Huang:</u> See [1.b]-[1.c]. As discussed in [1.c], Tso in view of Huang renders obvious the recited determination; Huang further discloses that if a “partially rendered” object is found in cache, the node creates an optimized media content by “complet[ing] the entire rendering process” based on “RHI” information stored with the content. Huang, 7:42-8:11, 6:9-23; see [1.c]. Applying this teaching to Tso’s teaching of performing transformation operations (see [1.b]) renders obvious this element. See §IX.A.2 (motivation to modify (“MTM”)); Madisetti, ¶¶104-07. • 7:42-8:11, 6:9-23 (see [1.c]) • See also Abstract, 3:17-48, 4:2-15, 5:44-55, 6:63-67, 10:46-56.
[1.e] responsive to determining that a previously-generated first intermediate media content is not available, creating a first optimized media content for the first	Tso discloses responsive to determining that a previously-generated first media content is not available (e.g., responsive to “determin[ing]” that “a copy of the required version” does not “resid[e] in server-side cache memory”), creating a first optimized media content for the first playback device (e.g., “retriev[ing] the hyper text object” and “transcode[] the data stream appropriately” for “network client 12”), by using the first original media content and the first set of content generation operations, and performing the first set of transformation operations (see [1.b]-[1.d]). <u>E.g., Tso:</u> See [1.pre], [1.b]-[1.d].

'745	Tso in view of Huang
playback device by creating a first intermediate content using the first original media content and the first set of content generation operations, and performing the first set of transformation operations on the first intermediate media content; and	Tso also discloses that if the system determines that the required version is not cached, then the system retrieves from the Internet and transcodes the original content for the client using the content generation and transformation operations, and caches the resulting transcoded content (see [1.b]-[1.d]). Madisetti, ¶¶108-09, 111. Figs. 8-9  FIG. 8 (Left): - Starts at node A. - Step 160: HTTP LOCAL PROXY SENDS REQUEST TO HTTP REMOTE PROXY. - Step 170: HTTP REMOTE PROXY CHECKS CACHE FOR URL OBJECT. - Decision 180: OBJECT IN CACHE? - If Y (Yes): RETRIEVE CACHED OBJECT. - If N (No): RETRIEVE URL OBJECT FROM INTERNET. - Decision 210: OBJECT FOUND? - If N (No): RETURN ERROR. - If Y (Yes): CACHE RETRIEVED OBJECT (labeled "original media content"). - Ends at node B. FIG. 9 (Right): - Starts at node B. - Decision 240: TRANSCODE OBJECT? - If N (No): TRANSCODE OBJECT (labeled "content generation operations" / "transformation operations"). - Step 250: TRANSCODE OBJECT. - Step 260: SEND OBJECT TO HTTP LOCAL PROXY. - Step 270: STORE OBJECT IN CLIENT-SIDE CACHE. - Decision 280: SUPPORTED DATA TYPE? - If Y (Yes): BROWSER DISPLAYS OBJECT. - If N (No): DECODE OBJECT (IF NEEDED). - Decision 310: SUPPORTED DATA TYPE? - If Y (Yes): BROWSER DISPLAYS OBJECT. - If N (No): ADD-IN DISPLAYS OBJECT. 10:24-49 ("Upon being invoked, <i>parser 22</i> first calls cache interface 28 with the requested hypertext object to determine whether a copy of the required version already resides in server-side cache memory 30. For purposes of illustration, assume no entry exists in server-side cache memory 30 for the requested hypertext object. <i>HTTP remote proxy 36</i> then invokes a call to retrieve the hypertext object from Internet 18 over server/network communications link 16. ... <i>Transcode service provider 24</i> then transcodes the data stream appropriately, and <i>HTTP remote proxy 26</i> transmits the transcoded data stream to network client 12.")

'745	Tso in view of Huang
	• See also 3:43-60, 4:62-66, 5:36-42, 6:9-50, 6:64-66, 14:21-63, 14:37-63. Huang further discloses the first intermediate media content (see [1.c]) and creating a first optimized media content for the first playback device (e.g., creating a “completely rendered ... object”) by creating a first intermediate content using the first original media content (e.g., creating and maintaining a “partially rendered ... object” in “cache” “based on objects retrieved from the content servers”) and the first set of operations (e.g., operations required for “partially rendered ... object”), and performing additional operations on the first intermediate media content (performing the “further rendering” required to generate a “completely rendered” object). <u>E.g., Huang:</u> In addition to Tso’s teaching of transcoding content that has not been cached using multiple transcode service providers selected using selection criterion, as discussed for [1.c], Huang further teaches generating optimized content based on objects retrieved from a content server by “stag[ing]” rendering and storing “partially rendered” content items. Huang, 6:63-67, 5:42-44, 7:36-41; Madisetti, ¶¶110-12. As discussed in §IX.A.2 and for [1.c] (MTM), a POSITA would have found it obvious to modify Tso with this teaching to advantageously cache the “partial” transcoding output from each of the transcode service providers. For example, in addition to transcoding the final version of content, modifying Tso with these teachings, the partially-transcoded output from Tso’s content generation operation transcode service providers is cached as intermediate media content, as taught by Huang, in order to “avoid repeating” transcoding steps. <i>Id.</i>; Madisetti, ¶113. • 5:42-44 (“Object renderings are performed by the proxies 110, 111, 112 based on objects retrieved from the content servers 120, 121.”) • 6:63-67, 7:42-8:11 (see [1.c])

'745	Tso in view of Huang
	• 7:36-41 (“<i>If the requested object cannot be found</i> in the cache, at step 404, <i>the proxy server</i> 110, 111, 112 modifies the associated RHI to indicate its ability for providing rendering services and then <i>sends the request</i> and the modified RHI to another proxy server or <i>to the content server ...</i>”) • See also Abstract, 2:32-44, 3:32-37.
[1.f] sending the first optimized media content to the first playback device.	Tso discloses sending the first optimized media content to the first playback device (e.g., “transmit[]” the “transcoded data stream” to “network client[] 12”). <u>E.g., Tso:</u> See [1.pre], [1.c]. Tso also discloses that after generating the “transcoded version” of the content, the server “transmits” a “data stream for the transcoded hypertext object to network client 12.” Madisetti, ¶¶114-16. • 6:51-63 (“<i>Multiple-thread processing improves the efficiency of the present embodiment by not waiting for a hypertext object to be received</i> in its entirety by HTTP remote proxy 36, <i>or added in its entirety to server-side cache</i> memory 30, <i>before beginning to send the object to network client 12</i>. Another benefit of multiple-thread processing is that <i>parser 22 may efficiently process requests for the same hypertext object from multiple network clients 12. The hypertext object need only be retrieved from Internet 18 once, and appropriate versions may be transmitted to such multiple network clients 12 concurrently.</i>”) • 10:24-49 (see [1.e]). • See also 6:24-50, 14:47-15:6.
[2.pre] The method of claim 1,	See claim 1. Madisetti, ¶117.

'745	Tso in view of Huang
further comprising:	
[2.a] receiving a second request from a second playback device for media content; wherein the second request contains information, the information indicating a second original media content, second set of content generation operations, and second set of transformation operations;	Tso discloses [2.a] as it discloses that “multiple network clients 12” send “requests” for media content containing the claimed information as discussed in [1.a]-[1.b] and thus discloses a second request from a second network client. Madisetti, ¶¶118-20. • 6:51-63 (<i>see</i> [1.f]) • 3:45-60 (<i>see</i> [1.a])
[2.b] wherein the second request is received substantially concurrently with receiving the first request;	Tso discloses wherein the second request is received substantially concurrently with receiving the first request (<i>e.g.</i>, “process requests” from “multiple network clients” and “concurrently” deliver “appropriate versions” of content to each of those “clients”). <u>E.g., Tso:</u> <i>See</i> [1.a]. Tso also discloses using “multiple-thread processing” to “concurrently” transmit appropriate versions of content to

'745	Tso in view of Huang
	multiple clients in response to requests from those clients; two such requests that are processed and served concurrently are received substantially concurrently. Madisetti, ¶¶121-23. • 6:51-63 (<i>see</i> [1.f]) • <i>See also</i> 4:62-5:7.
[2.c] wherein the first original media content and the second original media content are the same media content;	Tso discloses wherein the first original media content and the second original media content are the same media content (<i>e.g.</i>, two network clients request “the same hypertext object”). <u>E.g., Tso:</u> <i>See</i> [1.c], [2.b]. Tso also discloses that the system concurrently serves content in response to two client requests for the same hypertext object. Madisetti, ¶¶124-26. • 6:51-63 (<i>see</i> [1.f])
[2.d]-[2.g]	<i>See</i> [1.c]-[1.f], [2.a]-[2.c]. Tso in view of Huang renders obvious these elements, which recite performing the same steps with respect to the second request as recited in [1.c]-[1.f] with respect to the first request. Madisetti, ¶¶127-34.
[3]/[4] The method of claim 2, wherein the first set of content generation operations and the second set of content generation operations are [the same (claim 3)/	See claim 2. Tso discloses that the first set of content generation operations and the second set of content generation operations are [the same/different] (<i>see</i> [1.b], [2.a], the same or different content generation operations), and the first set of transformation operations and the second set of transformation operations are different (<i>see</i> [1.b], [2.a], different transformation operations). <u>E.g., Tso:</u> <i>See</i> [1.b]-[1.e]; claim 2. Claims 3-4 are the same, except for reciting “the same” or “different” content generation operations. Madisetti, ¶135.

'745	Tso in view of Huang
different (claim 4)], and the first set of transformation operations and the second set of transformation operations are different.	As explained for [1.b] and [1.d]-[1.e], Tso also discloses specifying different “content generation” and “transformation” operations using “selection criteria.” Tso, 6:64-7:14, 7:15-9. Because combinations of such criteria can be used to generate “virtually limitless” transformations, a POSITA would have understood Tso to disclose a second request for content that contains either the same or different content generation operations and different transformation operations from the first set of operations—<i>e.g.</i>, operations to generate a different data type, resolution, or number of colors for the content. Madisetti, ¶¶136-37. • 6:50-63 (<i>see</i> [1.f]) • 8:10-13 (<i>see</i> [1.pre])
[5.a] The method of claim 1, further comprising: determining whether a previously-generated first optimized media content is available for reuse, wherein the previously-generated first optimized media content was created using the first original media content, the first content generation	See claim 1. Tso discloses determining whether a previously-generated first optimized media content is available for reuse (<i>e.g.</i>, “determine whether a copy of the required version already resides in server-side cache memory 30”; <i>see</i> [1.c/1.e]), wherein the previously-generated first optimized media content was created using the first original media content, the first content generation operations, and the first transformation operations (<i>e.g.</i>, previously-generated “appropriate version[]” generated using the same “network client” “selection criterion” and “content characteristics”; <i>see</i> [1.b]-[1.e]). <u>E.g., Tso:</u> <i>See</i> [1.b]-[1.e]; claim 1. Tso also discloses that a server’s parser searches for the required version of the content—i.e., a version generated with the same content generation and transformation operations—in server-side cache memory. Madisetti, ¶¶138-40. • 10:24-49 (<i>see</i> [1.e]) • See also 3:43-60, 4:62-66, 5:36-42, 6:9-50, 6:64-66, 14:21-63, 14:37-63.

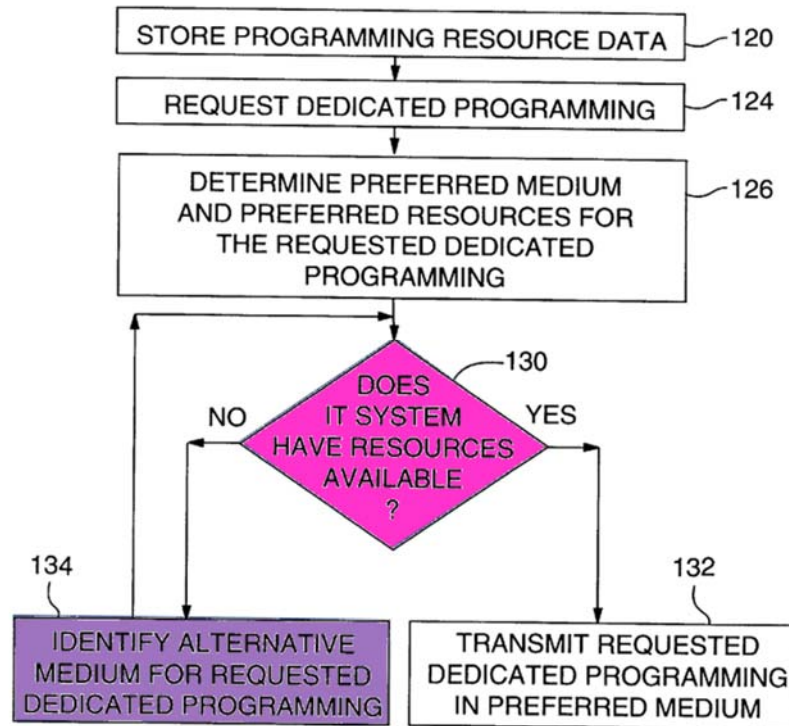
'745	Tso in view of Huang
operations, and the first transformation operations;	
[5.b] responsive to determining that the previously- generated first optimized media content is available, sending the previously- generated first optimized media content to the first playback device.	Tso discloses responsive to determining that the previously-generated first optimized media content is available (<i>e.g.</i>, if “the required version already resides in” cache), sending the previously-generated first optimized media content to the first playback device (<i>e.g.</i>, sending “transcoded versions ... without the need to ... re-transcode the content”). <u>E.g., Tso:</u> See [1.d], [1.f], [5.a]. Tso also discloses that when the “required version” is already cached, the cached version is transmitted to the client. Madisetti, ¶¶141-43. • 4:1-5 (<i>see</i> [1.c]) • 10:24-36 (<i>see</i> [1.e])

B. Ground 2: Tso in view of Huang and Lawler (Claims 6-7)

1. Lawler

Lawler teaches, in an interactive television or televideo (IT) system providing image and video content, using an alternative format of content to avoid exceeding “[t]he capacity of CPU 66.” Ex. 1006 (Lawler), 4:59-5:6, 5:57-62, 10:3-8; Madisetti, ¶¶144-45. Fig. 4 summarizes an embodiment:

FIG. 4



When a system “**lacks the resources required**” for a “**preferred format**,” the system uses “**an alternative format that requires fewer IT system resources**.” Lawler, 6:14-16, 2:53-55. Alternative formats include, *e.g.*, lower, “VHS” quality instead of higher, “S-VHS” quality streaming video; 8-bit instead of 24-bit “graphic code”; or “lower resolution or quality” versions. Lawler, 6:2-35; Madisetti, ¶¶146-47.

2. Motivations to Modify Tso in view of Huang with Lawler’s Teachings

Like **Tso** and **Huang**, **Lawler** is analogous art in the same field as the ’745, and reasonably pertinent to automating media delivery and reducing overhead and

time to generate Web pages. Lawler, 1:20-26 (“allocating resources of an [IT] system for delivering programming to viewers”), 2:31-33; 3:1-4, 4:59-5:6; *see also* §§IX.A.1-2; Madisetti, ¶148.

Both **Tso** and **Huang** teach considering processing resources when determining how to transcode content. *E.g.*, Tso, 7:15-31; Huang, 6:63-67, 2:9-13. A POSITA would have been motivated to modify **Tso’s** (as modified by **Huang’s**) teachings with Lawler’s teachings of “alternative” formats that consume fewer resources to avoid “interrupting acceptable delivery of the programming” for at least the following reasons. **Lawler**, 2:47-61, 5:40-6:18, 9:25-35; Madisetti, ¶149.

First, **Tso** already teaches using “system load” as a “selection criterion” when deciding how to transcode content. Tso, 7:15-31. **Lawler’s** detailed “CPU” capacity teachings provide additional details for how to implement **Tso’s** “system load” teachings by using this criterion to decide whether to use an “alternative format.” Lawler, 5:57-6:18. A POSITA would have found it obvious based on these complementary teachings to select an “alternative format” when CPU capacity use is too high. Madisetti, ¶150.

Second, **Tso** and **Lawler** recognize that, as known in the art, efficient and reliable service is advantageous. Tso, 6:55-58; Lawler, 11:42-50, 2:1-9; *see also* Starnes, Abstract, 2:66-3:5, 4:55-5:4, 6:14-18; Madisetti, ¶151. A POSITA would have been motivated to apply **Lawler’s** teachings to advantageously avoid

“interrupting acceptable delivery of the programming” by using alternative formats when needed. Lawler, 5:57-6:50; Madisetti, ¶151.

Third, Lawler teaches that using alternative, lower-quality content requires “fewer [*e.g.*, by “one-half”] IT system resources.” Lawler, 6:7-50, 2:53-61. A POSITA would have been motivated to modify **Tso** with **Lawler’s** teachings of such alternative formats (*e.g.*, reduced quality based on resolution or bitrate) to advantageously reduce “system load.” *Id.*; Tso, 7:15-31; *see also* Starnes, 2:24-41; Madisetti, ¶152.

Fourth, Lawler teaches that alternative content may vary by using, *e.g.*, lower-bit graphics or lower resolution content. Lawler, 6:7-12, 6:37-50. A POSITA would have found it obvious to modify **Tso** with these teachings because these are additional detailed teachings of “content characteristics” and “content provider preferences” for a content provider to change the quality of content. Tso, 7:15-8:9; Madisetti, ¶153.

A POSITA would have had a reasonable expectation of success in applying **Lawler’s** teachings. **Lawler** provides additional detail for **Tso’s** teachings of using “current processing load” to select transcoding processes (Tso, 7:7-14), and **Tso** recognizes that higher quality transcoding requires additional resources. Tso, 16:50-54. Modifying **Tso** with **Lawler’s** teachings thus is nothing more than a straightforward application of using **Tso’s** teachings of “current processing load” as

a selection criterion when determining how to transcode content to improve the system's efficiency when servicing user requests. Madisetti, ¶154.

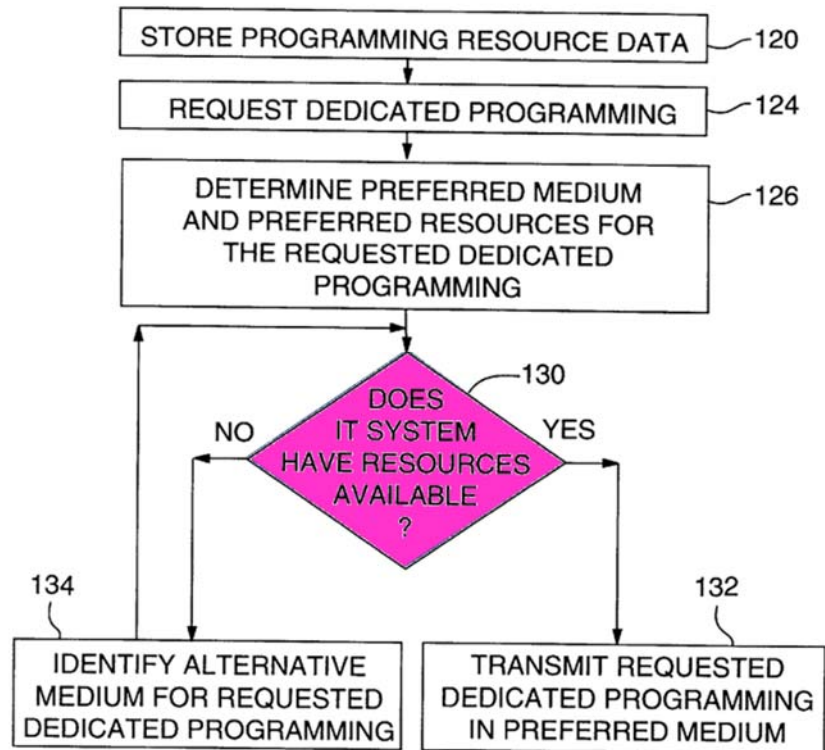
3. Claim Chart

'745	Tso in view of Huang and Lawler
[6.a] The method of claim 1, further comprising: determining whether the host computer has sufficient processing resources to create the first optimized media content for the first playback device;	See claim 1. Tso discloses determining whether the host computer has sufficient processing resources (<i>e.g.</i>, “dynamically updat[ing]” the “current processing load” to use as a “predetermined selection criterion”) and creating the first optimized media content for the first playback device (<i>see</i> [1.d]-[1.e]). <u>E.g., Tso:</u> <i>See</i> [1.d]-[1.e]; claim 1. Tso also discloses that the parser uses a “dynamically updated” “current processing load” as a selection criterion. Madisetti, ¶¶155-56, 158. • 7:7-31 (“Still further, <i>the predetermined selection criterion may comprise</i> a condition of the device on which transcoding server 34 is installed (for example, <i>a current processing load</i>), a condition of a device to which transcoding server 34 is coupled, or a condition of a communications link. Transcoding server 34 may provide the ability to <i>dynamically update such predetermined selection criteria</i>. ...”) Lawler discloses determining whether the host computer has sufficient processing resources for the first optimized media content for the first playback device (<i>e.g.</i>, making “an inquiry as to whether” “CPU 66” has “sufficient capacity” to meet the “ProcessorRequirement”). <u>E.g., Lawler:</u> Lawler discloses that a server checks whether the CPU has sufficient capacity to provide content in the preferred format. Lawler, Fig. 4, 9:24-27. Modifying Tso with Lawler's

teachings (see §IX.B.2), it would have been obvious to determine whether the system has sufficient “CPU” capacity (*i.e.*, processing load) for the preferred (*i.e.*, optimized) media content for playback. Madisetti, ¶¶157-59.

• **Fig. 4:**

FIG. 4



- **9:25-35** (“Decision block 130 represents an *inquiry as to whether* IT system 10 has available the resources to meet the resource requirements of the preferred medium. ... [C]entral control node ... inquires of **CPU 66** whether it and memory system 68 at station controller **have sufficient capacity to meet the** RAMRequirement and **ProcessorRequirement.**”)

- **See also Abstract, 10:3-8, 2:53-3:3, 2:1-4, 5:40-53.**

[6.b]
responsive to
determining
that the host
computer does

Tso not only renders obvious, but discloses creating the first optimized content (see [6.a]) and a first set of alternate content generation operations or first set of alternate transformation operations (see [1.b]-[1.d], [3]/[4]).

not have sufficient processing resources to create the first optimized media content, determining an alternate first set of content generation operations or an alternate first set of transformation operations;	As discussed for [6.a] and [1.b]-[1.d], Tso teaches that the system uses selection criteria including “current processing load,” “file size,” and “resolution” to determine how to transcode content among “virtually limitless” transcoding options. Tso, 6:64-7:7, 7:15-8:9. Using “current processing load,” content generation operations (<i>e.g.</i>, “file size”), and transformation operations (<i>e.g.</i>, “resolution” and “number of colors”) as selection criteria discloses using, <i>e.g.</i>, a smaller file size and/or lower resolution based on “processing load” as alternative content generation and/or transformation operations, respectively. Madisetti, ¶¶160, 163. Lawler discloses responsive to determining that the host computer does not have sufficient processing resources (<i>see</i> [6.a]), determining an alternate media content (<i>e.g.</i>, “identif[ying]” “an alternative format that requires fewer IT system resources”). <u>E.g., Lawler:</u> In addition to Tso’s and Lawler’s teachings of determining that a server does not have sufficient processing resources (<i>see</i> [6.a]), and Tso’s teaching of different content generation and transformation operations, Lawler further discloses using an alternative format that requires fewer server resources, rather than a preferred format, when CPU capacity is consumed so that the system can continue to reliably serve content to users. Lawler, 2:53-61; Madisetti, ¶¶162-64. A POSITA would have been motivated to modify Tso with this teaching, yielding a system in which the system uses an alternative format when the system’s CPU capacity selection criterion indicates that the server does not processing resources for the preferred format. <i>See</i> Lawler, 2:53-3:3; Tso, 7:7-31; Madisetti, ¶164; §IX.B.2. • 2:53-61 (“The IT system also <i>identifies alternative resources for delivering the programming in an alternative format that requires fewer IT system resources</i>. For example, <i>an alternative format to the preferred high-resolution digital video signal format could be a conventional-resolution digital video signal that corresponds to a standard VHS quality video signal</i>.”)
--	--

	<i>Such an alternative format could require only one-half the level of some resources</i> required by the preferred format.”) • 9:25-35 (see [6.a]) • See also 6:13-18.
[6.c] creating an alternate first optimized media content using the alternate first set of content generation operations or the alternate first set of transformation operations.	Tso discloses creating an alternate first optimized media content using the alternate first set of content generation operations (see [6.b], [1.b]-[1.d]) or the alternate first set of transformation operations (see [6.b], [1.b]-[1.c]). <u>E.g., Tso:</u> As explained for [1.b]-[1.d], Tso discloses generating content using content generation and transformation operations specified by the selection criteria. This includes the alternate operations discussed in [6.b]. Madisetti, ¶¶165-67.
[7.a] The method of claim 6, wherein the first optimized media content is at a first level of quality and the alternate first optimized media content is at a second level of quality, wherein the first level of quality is higher than	See claim 6. Lawler further discloses the first optimized media content is at a first level of quality (e.g., “preferred format” is “higher resolution or quality”), and an alternate first optimized media content is at a second level of quality, wherein the first level of quality is higher than the second level of quality (e.g., “[t]he alternative format” is “lower resolution or quality”). <u>E.g., Lawler:</u> See claim 6. Lawler also discloses that the first “preferred” format is “higher” quality, and the “alternative” format is “lower” quality. Madisetti, ¶¶168-70; <i>see also</i> §§IX.B.2-IX.B.3, [6.a]-[6.b] (discussing MTM). • 6:7-12 (“The <i>preferred format</i> of dedicated programming services typically <i>provides higher resolution or quality</i> at the expense of greater IT system resources requirements. <i>The alternative format provides the dedicated</i>

the second level of quality	<i>programming services at a lower resolution or quality that requires fewer IT system resources.”)</i> • <i>See also 2:53-61.</i>
[7.b] wherein a level of quality of a media content is measured based on a compression format, a bit rate, and an image resolution of a media content sent to a playback device.	Tso discloses wherein a level of quality of a media content is measured based on a compression format (<i>e.g.</i>, use “more resource-intensive compression algorithms” for “better quality content”) of a media content sent back to a playback device (<i>e.g.</i>, “content” sent back to system “users”). <u>E.g., Tso:</u> Tso discloses selecting more “resource-intensive compression algorithms” to generate “better” quality media content. Madisetti, ¶¶171-72, 174. • 16:50-54 (“Still further, transcoding server 34 may <i>use more resource-intensive compression algorithms</i>, for example, <i>to provide better quality content</i> for the same latency at the expense of slowing down access for non VIP users.”) • <i>See also 7:31-33, 8:25-28, 11:19-24, 12:26-32, 12:61-13:5, 16:2-5.</i> Lawler discloses wherein a level of quality of a media content is measured based on a bit rate (<i>e.g.</i>, quality is measured based on “24-bit” vs. “8-bit”), and an image resolution of a media content sent to a playback device (<i>e.g.</i>, “high-resolution” vs. “conventional-resolution” of “programming services” sent back to a user’s device). <u>E.g., Lawler:</u> Lawler discloses that “preferred” formats are “higher” quality, and that lower quality formats use lower (<i>e.g.</i> 8-bit) bitrate and resolution. Modifying Tso with Lawler’s teachings of using alternative lower quality content (<i>see</i> §IX.B.2), a POSITA would have found it obvious to measure quality using Lawler’s teachings of bitrate and resolution, in addition to Tso’s teaching of compression. Madisetti, ¶¶173-76. • 2:53-61 (<i>see</i> [6.b])

	• 6:7-12 (<i>see</i> [7.a]) • 6:37-50 (“For example, a preferred format of S-VHS quality digital video for video on-demand previews could have a first alternative format of VHS quality digital video, and subsequent <i>alternative modes in the form of 24-bit graphics and 8-bit graphics.</i>”) • <i>See also</i> 7:44-58, 4:59-65, Tables 1, 3A-3F.
--	---

C. Grounds 3/4: Samaniego (Claims 1-7)

1. Samaniego

Samaniego, published 6/20/2002, is §102(b) prior art as of that date (*see* §IV.B.1). PO cannot dispute Samaniego’s disclosure of the Claims because the ’745’s and Samaniego’s specifications are substantively identical. *See* Ex. 1013; *compare* ’745 with Samaniego; Madisetti, ¶¶177-78. If the ’745’s specification supports the Claims, then those same disclosures in Samaniego anticipate the Claims.

PO is estopped from disputing that Samaniego discloses the Claims. The Examiner considered and expressly found that “support for the claimed subject matter has been found in” Samaniego’s application. Ex. 1002, 165; Madisetti, ¶178; *see also* 35 U.S.C. §120; *Transco Prod. Inc. v. Performance Contracting, Inc.*, 38 F.3d 551, 557 n.6 (Fed. Cir. 1994) (parent must provide a “proper disclosure” for priority) (quoting MPEP §201.11). Applicant’s “silence during the original prosecution creates an estoppel” barring PO from now challenging this finding. *Ex*

Parte Smith, 2010 WL 3269939, at *6 (B.P.A.I. Aug. 17, 2010); *see also Zenith Labs., Inc. v. Bristol-Myers Squibb Co.*, 19 F.3d 1418, 1425 n.8 (Fed. Cir. 1994) (conduct that “played a material role in the issuance of the patent usually suffices [for estoppel to apply].”); *KangaROOS U.S.A., Inc. v. Caldor, Inc.*, 778 F.2d 1571, 1576 (Fed. Cir. 1985) (“tak[ing] advantage of an error by the PTO” violates “duty of square dealing and full disclosure”).

PO is therefore estopped. Nonetheless, to the extent required, Grounds 4-6 explain why Samaniego alone and in view of Tso or Lawler also renders obvious the Claims.

2. PO Intentionally Changed Inventorship

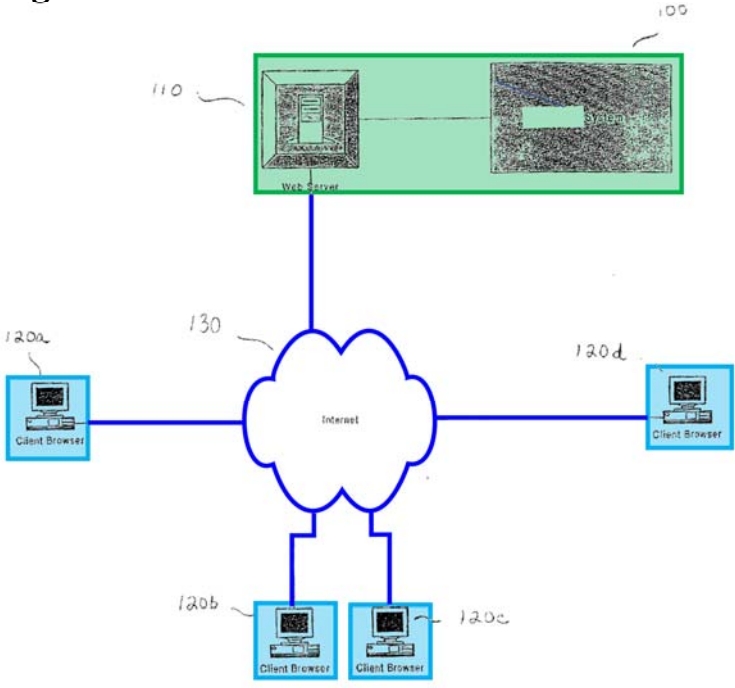
Original assignee Equilibrium and its current CEO Sean Barger intentionally identified different inventors in Samaniego and the '745. Before Samaniego's filing in 2001, Mr. Barger left Equilibrium, and did not return until 2004, when his company (AMPS d/b/a Equilibrium) acquired Equilibrium's assets. *See* Ex. 1014. Before Mr. Barger returned, Mr. Samaniego left Equilibrium in 2003 to join imaging software company Scene7, which became the exclusive distributor for Equilibrium's MediaRich, which Equilibrium alleges practices the '745. *See* Exs. 1014-1016. In September 2004, Mr. Barger sued Mr. Samaniego, Scene7, and Equilibrium's prior ownership alleging misconduct when transferring assets to Scene7. *See* Ex. 1020.

While this litigation was ongoing, Mr. Barger and Equilibrium filed

Application No. 11/269,916 (“’916”)—an alleged “continuation-in-part” but, as discussed in §IX.C.1, with a substantively identical specification to Samaniego. *See generally* Ex. 1010; Ex. 1021. The ’916 and Samaniego initially named the same inventors (*see* Ex. 1010, 68-70; Ex. 1034, 8), but the USPTO in 2005 raised that those named inventors did not sign the application. Ex. 1010, 76-77. In response, Mr. Barger and Equilibrium amended the application, adding Mr. Barger as an inventor, and attested that the applicant was unable to reach any of the original named inventors—an attestation that was blatantly false given Mr. Barger’s ongoing and active litigation against Mr. Samaniego. *See* Ex. 1010, 78-98; Ex. 1020, 1. Mr. Barger and Equilibrium then filed Application Nos. 12/173,747 and 12/238,842, both again with substantively identical specifications to Samaniego (*see* Exs. 1026-1029), both of which replaced all of Samaniego’s named inventors with new named inventors. *See* Ex. 1026, 76-87; Ex. 1027, 77-86; Ex. 1007. The ’745 issued from a continuation of 12/238,842. Equilibrium thus intentionally changed inventorship in the ’745—despite maintaining the same specification.

Samaniego therefore qualifies as invalidating prior art under pre-AIA §102(b) that the USPTO already found discloses the claims.

3. Claim Chart

'745	<i>E.g., Samaniego</i>
[1.pre] A method in a host computer for developing transformation processing operations to optimize media content playback to a plurality of playback devices connected with the host computer in a network, the method comprising:⁶	- Fig. 1:  Fig. 1 [0002] (“[T]he invention relates to an Internet server-based software system that provides delivery of automated graphics and other media to Web sites for access by an end user or consumer.”) [0094] (“The system 100 is attached to a Web server 110, which is connected to multiple client browsers 120(a-d) via the Internet 130.”) [0148] (“FIG. 21 is a schematic diagram showing the process flow of a proprietary enabled page”)

⁶ Color is used consistently through Grounds 3-6 to show disclosures from the cited art. Madisetti, ¶179.

'745	<i>E.g., Samaniego</i>
	delivered to a Web browser A content generation procedure 2140 is created with instructions on how the media is to be transformed to create the desired Web page content.”) • See also [0049], [0065], [0091], [0096]-[0097], [0137], [0140], [0148], Figs. 3, 7, 9, 21; Madisetti, ¶¶180-81.
[1.a] receiving a first request from a first playback device for media content;	• [0148] (“[T]he browser contacts the system 100 requesting media specified in the URL.”) • See also [0108]-[0110], [0138], [0143], Figs. 7, 9; Madisetti, ¶¶182-83.
[1.b] wherein the first request contains information, the information indicating a first original media content, first content generation operations, and first transformation operations;	See [1.a] for previously-recited elements. • [0148] (“...[T]he browser contacts the system 100 requesting media specified in the URL. The system parses the URL 2100 to determine the content generation procedure 2140 to execute, ... any dynamic content processing 2150 to be performed by dynamic media procedures, ...”) • See also [0064], [0066], [0090]-[0092], [0097], [0102], [0108], [0111], [0134], [0135], [0143], Figs. 7, 9, 10; Madisetti, ¶¶184-85.
[1.c] determining whether a previously-generated first intermediate media content is available for reuse, the previously-generated first intermediate media content having been created using the first original media	See [1.b] for previously-recited elements. Samaniego’s system performs a “secondary lookup” for a cached version of content created using the first set of content generation operations. If that version does not exist, the system creates and caches this intermediate version for future requests. Madisetti, ¶¶186-87. • [0148]-[0149] (“Upon receipt of the page 301, the browser contacts the system 100 requesting media specified in the URL. The system parses the URL 2100 to determine the content

'745	<i>E.g., Samaniego</i>
content and the first set of content generation operations; and	generation procedure 2140 to execute, ... In the case of media generation requiring dynamic content processing, a unique secondary lookup key corresponding to intermediate media is generated 2130. If intermediate media 2170 corresponding to this key is found, ... [a] unique key is generated for the derivative 2130 and passed to the media caching system 2120. If the media caching system finds no such intermediate image, such intermediate image is generated according to instructions specified by the content generation procedure...."). • <i>See also</i> [0064], [0065], [0104], [0108], [0111], [0140], [0143]-[0146], [0152].
[1.d] responsive to determining that a previously-generated first intermediate media content is available, creating a first optimized media content for the first playback device by performing the first set of transformation operations on the previously-generated first intermediate media content; and	<i>See</i> [1.a]-[1.c] for previously-recited elements. • [0149] (“If intermediate media 2170 corresponding to this key is found, such media is passed directly to the dynamic media content system 2150 having dynamic media procedures, wherein appropriate action is taken to generate the required derivative from the intermediate media data.”) • <i>See also</i> [0065], [0103], [0105], [0108], [0111], [0112]-[0114], [0125], [0136], [0138], [0143], [0145], [0147], [0148], [0150]-[0152], Figs. 6, 7, 9, 20; Madisetti, ¶¶188-89.
[1.e] responsive to determining that a previously-generated first intermediate media content is not available, creating a first optimized media	<i>See</i> [1.a]-[1.c] for previously-recited elements. • [0097] (“Web images 420 are Web versions of the original images. Web images 420 are ready for retrieval by the Web server 110 to be delivered to a Web browser 120. ... HTML

'745	<i>E.g., Samaniego</i>
content for the first playback device by creating a first intermediate content using the first original media content and the first set of content generation operations, and performing the first set of transformation operations on the first intermediate media content; and	pages 460 have references to Web images 420...") • [0149] ("If the media caching system finds no such intermediate image, such intermediate image is generated according to instructions specified by the content generation procedure, cached by the media cache system 2120 as a secondary cached media 2170, and passed to the dynamic media system 2150. Again, appropriate action is taken to generate the required derivative from the intermediate image data.") • See also [0064]-[0065], [0105], [0108], [0111]-[0114], [0125], [0136], [0138], [0140], [0143], [0145], [0147]-[0148], [0150]-[0151], Figs. 6, 7, 9, 20; Madisetti, ¶¶190-91.
[1.f] sending the first optimized media content to the first playback device.	See [1.a], [1.d]-[1.e] for previously-recited elements. • [0151] ("The resulting image is handed to the media cache system 2120 for caching and returned to the browser 120."); • See also [0108], [0111], [0138], [0147], [0149], Figs. 7, 10, 18; Madisetti, ¶¶192-93.
[2.pre] The method of claim 1, further comprising:	See claim 1. Madisetti, ¶¶194-95.
[2.a] receiving a second request from a second playback device for media content; wherein the second request contains information, the information indicating a second original media	Samaniego discloses [2.a] as it discloses that "multiple client browsers" send "requests" for media content containing the claimed information as discussed in [1.a]-[1.b] and thus discloses a second request from a second client browser. Madisetti, ¶¶196-97. • [0137] ("Multiple client browsers 120a-120d communicate with both the Web server 110 and the image server 100 via the Internet 130.")

'745	<i>E.g., Samaniego</i>
content, second set of content generation operations, and second set of transformation operations;	• [0135] (“[G]enerated media is cached such that further requests for the same media require little overhead.”) <i>See also</i> [0065], [0092], [0135], [0108] [0110]-[0111], [0137]-[0138], [0140], [0143], Figs. 7, 9, 17-18, 20.
[2.b] wherein the second request is received substantially concurrently with receiving the first request;	PO is estopped from disputing this limitation is disclosed (<i>see</i> §IX.C.1-2). As shown in Fig. 1 (<i>see</i> [1.pre]), multiple clients communicate with the Web server over the Internet, which not only renders obvious, but discloses, a server receiving substantially concurrent requests from multiple clients due to the number of clients in the Internet. Madisetti, ¶198; <i>see also</i> [2.a]. <i>See also</i> [0138] (<i>see</i> [1.pre], [2.a]), [0054], [0065], [0090]-[0094], [0126], [0136]-[0138], [0140], [0143], [0146], [0149], Figs. 1, 17-19.
[2.c] wherein the first original media content and the second original media content are the same media content;	• [0135] (“[T]he generated media is cached so that further requests for the same media require little overhead.”) • <i>See also</i> [0066], [0091]-[0092], [0099], [0136]-[0138], [0143], Fig. 6; Madisetti, ¶¶199-200.
[2.d]-[2.g]	<i>See</i> [1.c]-[1.f]. These elements recite performing the same steps with respect to the second request as recited in [1.c]-[1.f] with respect to the first request. Madisetti, ¶¶201-04.
[3]/[4] The method of claim 2, wherein the first set of content generation operations and the second set of content generation operations are [the same/ different], and the first set of	<i>See claim 2.</i> <i>See</i> [1.b]-[1.c]. Samaniego discloses handling requests for content using “script commands” and “content creation commands.” This not only renders obvious, but discloses, the same media content being subjected to the same or different script commands and/or content creation commands based on different requests. Madisetti, ¶¶205-06.

'745	<i>E.g., Samaniego</i>
transformation operations and the second set of transformation operations are different.	• [0114] (“Table B herein below provides the claimed script commands according to a preferred embodiment of the invention. Additional commands may be added as needed.”) • [0136] (“Because the invention takes the original media, content generation procedures, and proprietary URL tags as inputs for generating the Web media, it is possible to modify any of these inputs and have the system automatically update the media on the associated Web pages.”) • [0156] (“Table E below is a list of example supported content creation commands according to a preferred embodiment of the invention. Additional commands may be added as necessary.”) • <i>See also</i> Tables B, E, [0139], [0046]-[0050].
[5] The method of claim 1, further comprising: determining whether a previously-generated first optimized media content is available for reuse, wherein the previously-generated first optimized media content was created using the first original media content, the first content generation operations, and the	See claim 1. See [1.c]. The system performs a “primary” lookup for media generated using content generation and transformation (e.g., “dynamic content processing”) operations. If such content is available, the system “immediately” returns the content to the browser. Madisetti, ¶¶207-08. • [0148]-[0149] (“Upon receipt of the page 301, the browser contacts the system 100 requesting media specified in the URL. The system parses the URL 2100 to determine the content generation procedure 2140 to execute, ... any dynamic content processing 2150 to be performed by dynamic media procedures.... The parser generates a unique primary lookup key 2110 for the specified resulting media. If

'745	<i>E.g., Samaniego</i>
first transformation operations; responsive to determining that the previously-generated first optimized media content is available, sending the previously-generated first optimized media content to the first playback device.	the key corresponds to an existing generated media 2180, such media is returned immediately to the browser 120....”) • <i>See also</i> [0064]-[0065], [0104], [0108], [0111], [0138], [0152], Figs. 7, 10, 22.
[6.a] The method of claim 1, further comprising: determining whether the host computer has sufficient processing resources to create the first optimized media content for the first playback device;	<i>See claim 1.</i> PO is estopped from disputing that this limitation is disclosed (see §IX.C.1). Samaniego discloses that the system uses “web server traffic” and “client connection speed,” which are variables impacting host processing resources, as “input[s]” to “optimize” requested images. Samaniego, [0135], Table C. This not only renders obvious, but discloses, that the system “optimizes” content based on these variables by determining whether there are sufficient processing resources to generate requested content. Madisetti, ¶¶209-10. • [0135] (“The system takes as input the client connection, server traffic, ... to generate optimized media for the client.”) • Table C (“Optimizes Web server traffic by adjusting the bandwidth of graphics; Optimizes images for client connection speed; Allows clients to specify the quality of images on a Web site”) • <i>See also</i> [0044], [0051], [0065], [0091], [0102], [0108], [0111], [0134]-[0135], [0140], [0143],

'745	<i>E.g., Samaniego</i>
	[0148]-[0149], [0152], Figs. 6, 7, 10, 20, 22, Tables E-F.
[6.b] responsive to determining that the host computer does not have sufficient processing resources to create the first optimized media content, determining an alternate first set of content generation operations or an alternate first set of transformation operations;	PO is estopped from disputing that this limitation is disclosed (see §IX.C.1). See [6.a]. Samaniego also discloses determining whether the host computer has sufficient processing resources to generate optimized media (see [6.a]), as well as content generation operations and transformation operations (see [1.b]-[1.d]). When there are insufficient processing resources, Samaniego not only renders obvious, but discloses, that the content is “optimized” by, <i>e.g.</i>, adjusting the bandwidth of graphics. Samaniego, Table C; Madisetti, ¶¶211-12.
[6.c] creating an alternate first optimized media content using the alternate first set of content generation operations or the alternate first set of transformation operations.	PO is estopped from disputing that this limitation is disclosed (see §IX.C.1). Samaniego not only renders obvious, but discloses, performing the content generation and transformation operations that were determined to create optimized media content. See [6.a]-[6.b], [1.b]-[1.d]); Madisetti, ¶¶213-14.
[7] The method of claim 6, wherein the first optimized media content is at a first level of quality and the alternate first optimized media content is at a second level of quality, wherein the first	See claim 6. PO is estopped from disputing that this limitation is disclosed (see §IX.C.1). See [6.a]-[6.c], [1.f]. Samaniego discloses that content sent to the client is “optimized” for “web server traffic” and “client connection speed” by, <i>e.g.</i>, adjusting graphics bandwidth. Samaniego, Table C. The “save” command permits selecting a lossy “JPG” or lossless “PNG” compression format. Samaniego, Tables B-C. “Convert” permits changing a bit rate or “bit-

'745	<i>E.g., Samaniego</i>
level of quality is higher than the second level of quality; wherein a level of quality of a media content is measured based on a compression format, a bit rate, and an image resolution of a media content sent to a playback device.	depth". <i>Id.</i> And SetResolution specifies an image resolution. <i>Id.</i> These disclosures not only render obvious, but disclose, generating these lower quality (alternate) graphics using these parameters in order to use fewer processing resources. Madisetti, ¶¶215-16. • Table B ("Convert(rtype @ <bit-depth> {dither @ <value 0..10>}) This command converts the image to the specified type/bit-depth. ... Save Save([type @ <image-type>]) This command saves a media to the specified file. Set Resolution SetResolution([dpi @ <value>] [xdpi @ <value>] [ydpi @ <value>]) This command changes the DPI of the image in memory.") • Table C ("Reads and writes various file formats: BMP, GIF, JPG, PNG, TIF, PICT, TGA, PSD, FPX; Optimizes Web server traffic by adjusting the bandwidth of graphics; Optimizes images for client connection speed; Allows clients to specify the quality of images on a Web site") • <i>See also</i> [0005], [0032], Tables E-F (similar to Tables B-C).

D. Ground 5: Samaniego in View of Tso (Claims 2-4).

To the extent it is argued that **Samaniego** does not disclose [2.b], **Tso** teaches

“concurrently” handling two client requests for the same content. See §IX.A.3 (claim 2). A POSITA would have been motivated to apply these teachings to **Samaniego**. With its substantially common specification, **Samaniego** is in the same field as the ’745. *Compare Samaniego with ’745*; Madisetti, ¶217.

While **Samaniego** teaches that it is advantageous to avoid “regeneration of images,” **Tso** teaches using multiple-thread processing to avoid generating two copies of the same image concurrently. Samaniego, [0054]; Tso, 6:45-50; Madisetti, ¶¶218-19. **Samaniego** also teaches optimizing web server traffic by adjusting graphics bandwidth, while **Tso** teaches adjusting the transcoding of content based on processing load, and reducing the content quality for non-VIP clients. Samaniego, [0115], Table C; Tso, 6:39-54; Madisetti, ¶219. A POSITA would have been motivated to modify **Samaniego** with **Tso’s** teachings to achieve the beneficial and predictable results of handling concurrent requests and ensuring priority customers are efficiently provided with high quality content. Madisetti, ¶219.

First, **Tso’s** multiple-thread processing complements **Samaniego’s** teaching of handling multiple requests without regenerating the same content by avoiding processing each request separately. Tso, 6:51-63; Samaniego, [0110]; Madisetti, ¶220. A POSITA would have been motivated to modify **Samaniego** with **Tso’s** teachings to reduce this excess processing. Madisetti, ¶220.

Second, **Tso** teaches that multiple-thread processing improves system

efficiency because the system does not “wait[] for a hypertext object” before reusing original items. Tso, 6:51-63; Madisetti, ¶221. **Samaniego** teaches that such improvements to the “speed” to “add images, video, and sound to sites” are advantageous. Samaniego, [0045]; Madisetti, ¶221.

Third, Tso’s teaching of managing processing load complements **Samaniego’s** teaching of adjusting the quality of images based on client “connection speed” and “server traffic.” Samaniego, [0115], Table C. **Tso** discloses using the “condition of the device” to adjust content processing—which expands on **Samaniego’s** teaching by identifying similar parameters to “server traffic,” including “current processing load” at a transcoding server. Tso, 6:64-7:14. A POSITA would have found it obvious to further consider “current processing load” at a server when determining how to adjust the bandwidth of images to beneficially maximize the use of server processing resources. Madisetti, ¶222.

In light of these teachings, a POSITA would have had a reasonable expectation of success in modifying **Samaniego** with **Tso’s** teachings of multiple thread processing. Indeed, **Tso** teaches that its disclosed multiple thread access to cached objects uses a “standard” interface and does not “require any particular configuration” for the cache, while **Samaniego** teaches using a cache to avoid regenerating content for subsequent requests. Tso, 4:48-5:7, 5:2-7; Samaniego, [0135], [0138]. A POSITA therefore would have found it straightforward and

obvious to modify **Samaniego** with **Tso's** teachings, yielding a system in which **Samaniego's** teachings are used to concurrently serve content to multiple users using multiple thread processing. Madisetti, ¶223. A POSITA likewise would have had a reasonable expectation of success because **Tso** merely discloses using factors to consider when adjusting the bandwidth of images that are similar to those already disclosed in **Samaniego**. Madisetti, ¶223.

E. Ground 6: Samaniego in View of Lawler (Claims 6-7).

To the extent it is argued that **Samaniego** does not disclose using alternate content generation or transformation operations when the host computer does not have sufficient processing resources or measuring quality using bit rate or resolution, a POSITA would have been motivated to apply **Lawler's** teachings of selecting alternative, lower quality media formats to **Samaniego**. As discussed above in §IX.B.1, **Lawler**, like **Samaniego** is in the same field as the '745. Madisetti, ¶224.

Samaniego discloses using server load—which a POSITA would have understood includes processing capacity—to determine the content generation and transformation operations. **Samaniego**, [0091], [0135]. **Lawler** provides supplemental teachings of selecting alternative, lower quality content, including lower in quality based on bit rate or resolution, depending on CPU processing capacity to avoid interrupting delivery of content to clients. *E.g.*, **Lawler**, 2:47-61, 5:57-6:18, 9:25-35; Madisetti, ¶225. A POSITA thus would have been motivated to

modify **Samaniego** with **Lawler's** teachings for at least the following independent reasons. Madisetti, ¶225.

First, both **Samaniego** and **Lawler** describe systems for transmitting content to multiple clients. *E.g.*, Samaniego, [0094]; Lawler, 5:40-56, 6:13-16. **Lawler discloses** that such systems suffer from capacity limitations when serving multiple clients in such systems because “deliver[ing] programming to multiple subscribers or viewers” can result in “[c]apacity limitations” when “IT system 10 lacks the resources required to provide the service in the requested format.” **Lawler**, 5:40-56; 6:13-16. A POSITA would have been motivated to modify **Samaniego** with **Lawler's** teachings of using “alternative” formats to advantageously avoid interrupting content delay to clients when serving multiple clients concurrently. Lawler, 6:14-16, 2:53-55; Madisetti, ¶226.

Second, **Samaniego** discloses that it would be advantageous to service requests with “little overhead.” Samaniego, [0135]. A POSITA would have recognized that **Lawler's** teaching of using alternative requirements advantageously reduces overhead as described in **Samaniego** by using “less resources” for alternative content when additional processing capacity is unavailable. Lawler, 7:18-43; *see also id.* 5:40-56; 6:13-16; Madisetti, ¶227.

Third, **Lawler** provides complementary teachings of measuring quality. **Samaniego** discloses that its system allows clients to “specify the quality” of

content, and that it would be advantageous to allow a system to specify an “optimum quality” for content, and describes a system that allows “image quality and compression to be set.” Samaniego, [0052], [0032], [0045], Tables C, F. A POSITA would have found it obvious to further modify **Samaniego’s** teachings of quality based on compression with **Lawler’s** teachings of measuring quality using bit rate and resolution, yielding a system that is able to measure quality using bit rate, resolution, and compression. Lawler, 6:7-12, 2:53-61, 6:37-50; Madisetti, ¶228.

In light of the above teachings, a POSITA would have had a reasonable expectation of success in applying **Lawler’s** teachings. Indeed, **Samaniego** already teaches that “system features” include optimizing web server traffic by “adjusting the bandwidth of graphics.” Samaniego, Tables C, F. Modifying **Samaniego** with **Lawler’s** teachings would have been a straightforward application of those teachings in a manner that advantageously improves the efficiency of the system in servicing many user requests. Madisetti, ¶229.

With respect to [6.a], as disclosed in §IX.C.3([6.a]), **Samaniego** discloses that the system uses “server traffic” as an input when generating optimized media. Samaniego, [0051], [0065], [0091], [0135], Tables C, F. **Lawler** further discloses determining whether the host computer has sufficient processing resources for the first optimized media content for the first playback device (*see* §IX.B.3) by, *e.g.*, determining whether the system “has available the resources,” including “capacity”

in “CPU 66,” to satisfy a “ProcessorRequirement” for generating content. Lawler, Fig. 4, 9:25-35. A POSITA would have been motivated to modify **Samaniego’s** teaching of using “server traffic” as an input with this teaching from **Lawler**, yielding a system in which **Samaniego’s** system uses a determination of whether the server has sufficient processing resources as an input when generating optimized media for content—rendering this element obvious. Madisetti, ¶230.

With respect to [6.b], as discussed in §IX.C.3 ([1.b]-[1.d], [6.b]), **Samaniego** discloses that the server determines content generation and transformation operations based on system inputs. Samaniego, [0066], [0094], [0097], [0148]-[0149]. **Lawler** further discloses responsive to determining that the host computer does not have sufficient processing resources, determining an alternate media content (see §IX.B.3) by, *e.g.*, “identif[ying] alternative resources” for delivering media content “in an alternative format that requires fewer IT system resources.” Lawler, 2:53-61. A POSITA would have been motivated to modify **Samaniego** with this teaching, yielding a system in which, when in [6.a] the system uses processing resources as an input and determines that it does not have enough resources, the system responds by using alternative content generation or transformation operations for which it does have available processing resources—disclosing this element. Madisetti, ¶231.

With respect to [6.c], as discussed in §IX.C.3 ([6.c], [1.b]-[1.d]), **Samaniego**

discloses creating optimized media content using content generation operations and transformation operations. Samaniego, [[0066], [0148]-[0149]. A POSITA would have been motivated to modify **Samaniego** with **Lawler’s** teachings as discussed above for [6.a]-[6.b] to generate alternative content generation and transformation operations; the resulting system generates media using the alternative, rather than preferred, content generation or transformation operations—disclosing this element. Madisetti, ¶232.

With respect to [7.a], as discussed above for claim 6, **Samaniego** in view of **Lawler** renders obvious generating the first optimized media. **Lawler** further discloses the first optimized media content is at a first level of quality, and an alternate first optimized media content is at a second level of quality, wherein the first level of quality is higher than the second level of quality (*see* §IX.B.3) by using, *e.g.*, an alternative media format that “requires fewer IT system resources.” Lawler, 2:53-61; Madisetti, ¶233. A POSITA would have been motivated to modify **Samaniego** with this teaching to generate media of an alternative format that uses fewer processing resources when the determination in [6.a] indicates there are insufficient processing resources; this alternative format is of a lower quality because “higher resolution or quality” is generated “at the expense of greater IT system resources requirements.” Lawler, 2:53-61, 5:40-6:12; Madisetti, ¶233.

With respect to [7.b], as disclosed above in §IX.C.3 ([7.b]), **Samaniego**

discloses wherein a level of quality of a media content is measured based on a compression format (Samaniego, [0032]). **Lawler** further discloses wherein a level of quality of a media content is measured based on a bit rate, and an image resolution of a media content sent to a playback device *see* §IX.B.3). Madisetti, ¶234. A POSITA would have been motivated to modify **Samaniego** by applying **Lawler's** teachings to measure quality using bit rate and image resolution by using these properties in addition to compression format to measure quality because each of these features separately affects the processing required to generate the alternative format of content—and therefore the quality of content that can be generated. Lawler, 2:53-61, 4:59-65, 5:40-6:12; Madisetti, ¶234.

X. SECONDARY CONSIDERATIONS

There is no evidence in the '745's prosecution history or elsewhere supporting any secondary considerations arguments, or evidence of nexus to any Challenged Claim. *See generally* '745 FH; Madisetti, ¶235. Indeed, as demonstrated by the prior art referenced herein, any purported solutions to problems or unexpected results in the '745 were already well known. *Id.* To the extent PO asserts the existence of any secondary considerations in its responses, Petitioner reserves the right to address any such evidence.

XI. CONCLUSION

Substantial, new, and noncumulative technical teachings have been presented

for the '745's Claims, which are anticipated and obvious for the reasons set forth above. Madisetti, ¶¶236-41. There is a reasonable likelihood that Petitioner will prevail as to the Claims. *Inter partes* review of claims 1-7 is accordingly requested.

Respectfully submitted,

Dated: January 10, 2023

By: /Scott McKeown/
Name: Scott McKeown
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE

Pursuant to §42.24(d), the undersigned certifies that the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 9,158,745 contains, as measured by the word-processing system used to prepare this paper, 13,979. This word count does not include the items excluded by §42.24 as not counting towards the word limit.

Respectfully submitted,

Dated: January 10, 2023

By: /Daniel W. Richards/
Name: Daniel W. Richards
Counsel for Petitioner

CERTIFICATE OF SERVICE

I hereby certify that on January 10, 2023, I caused a true and correct copy of the foregoing Petition for *Inter Partes* Review of U.S. Patent No. 9,158,745 and supporting exhibits to be served via express mail on the Patent Owner at the following correspondence address of record as listed on PAIR:

22862 - GLENN PATENT GROUP
c/o Perkins Coie LLP
P.O. Box 1247
Seattle, WA 98111-1247
UNITED STATES

A courtesy copy was also sent via electronic mail to the Patent Owner's litigation counsel at the following addresses:

Jason R. Bartlett (jbartlett@mkwllp.com)
Jason A. Crotty (jcrotty@mkwllp.com)
Marc J. Pernick (mpernick@mkwllp.com)
MAURIEL KAPOUYTIAN WOODS LLP
450 Sansome Street, Suite 1005
San Francisco, CA 94111
Tel: (415) 738-6228

Steven Callahan (scallahan@ccrglaw.com)
Christopher T. Bovenkamp (cbovenkamp@ccrglaw.com)
CHARHON CALLAHAN ROBSON &
GARZA, PLLC
3333 Lee Parkway, Suite 460
Dallas, TX 75219
Tel: (214) 521-6400

David E. Moore (#3983)

Bindu A. Palapura (#5370)
Brandon R. Harper (#6418)
Carson R. Bartlett (#6750)
Hercules Plaza, 6th Floor
1313 N. Market Street
Wilmington, DE 19801
Tel: (302) 984-6000
dmoore@potteranderson.com
bpalapura@potteranderson.com
bharper@potteranderson.com
cbartlett@potteranderson.com

Dated: January 10, 2023

By: /Jonathan Bradford/
Name: Jonathan Bradford
ROPES & GRAY LLP